

BRUNO LUIZ PEREIRA

**APLICAÇÃO DE SISTEMAS NEURO-FUZZY E
EVOLUÇÃO DIFERENCIAL NA MODELAGEM E
CONTROLE DE VEÍCULO DE DUAS RODAS**



**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA**

2017

BRUNO LUIZ PEREIRA

**APLICAÇÃO DE SISTEMAS NEURO-FUZZY E EVOLUÇÃO
DIFERENCIAL NA MODELAGEM E CONTROLE DE VEÍCULO DE
DUAS RODAS**

Dissertação de Mestrado apresentada ao
programa de pós-graduação da Faculdade de
Engenharia Mecânica da Universidade Federal
de Uberlândia como parte dos requisitos para
obtenção do título de **MESTRE EM**
ENGENHARIA MECÂNICA.

Área de Concentração: Mecânica dos Sólidos e
Vibrações.

Orientador: Prof. Dr. Leonardo Sanches.

Coorientador: Prof. Dr. José Jean-Paul
Zanlucchi de Souza Tavares.

UBERLÂNDIA - MG

2017

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

P436a
2017

Pereira, Bruno Luiz, 1993-
Aplicação de sistemas neuro-fuzzy e evolução diferencial na
modelagem e controle de veículo de duas rodas / Bruno Luiz Pereira. -
2017.
172 f. : il.

Orientador: Leonardo Sanches.

Coorientador: José Jean-Paul Zanlucchi de Souza Tavares.

Dissertação (mestrado) - Universidade Federal de Uberlândia,
Programa de Pós-Graduação em Engenharia Mecânica.

Disponível em: <http://dx.doi.org/10.14393/ufu.di.2017.46>

Inclui bibliografia.

1. Engenharia mecânica - Teses. I. Sanches, Leonardo. II. Tavares, José Jean-Paul Zanlucchi de Souza. III. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Engenharia Mecânica. IV. Título.



ALUNO: Bruno Luiz Pereira

NÚMERO DE MATRÍCULA: 11522EMC014

ÁREA DE CONCENTRAÇÃO: Mecânica dos Sólidos e Vibrações

LINHA DE PESQUISA: Dinâmica de Sistemas Mecânicos

PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA: NÍVEL MESTRADO

TÍTULO DA DISSERTAÇÃO:

“Aplicação de Sistemas Neuro-Fuzzy e Evolução Diferencial na Modelagem e Controle de Veículo de Duas Rodas”

ORIENTADOR: Prof. Dr. Leonardo Sanches

CO-ORIENTADOR: Prof. Dr. José Jean-Paul Zanlucchi de Souza Tavares

A Dissertação foi **APROVADA** em reunião pública, realizada no Anfiteatro G do Bloco 50, Campus Santa Mônica, em 25 de agosto de 2017, às 14:00 horas, com a seguinte Banca Examinadora:

NOME

Prof. Dr. Leonardo Sanches

Prof. Dr. José Jean-Paul Z. de Souza Tavares

Prof. Dr. Roberto Mendes Finzi Neto

Prof. Dr. Laécio Carvalho de Barros

ASSINATURA

UFU

UFU

UFU

UNICAMP

Uberlândia, 25 de agosto de 2017

Dedico este trabalho a minha família, sobretudo aos meus pais, por sempre estarem ao meu lado em todos os momentos.

Agradecimentos

Aos meus pais, Cristiane e Wagner, pelo carinho, amor, compreensão, incentivo, patrocínio, por acreditarem na minha capacidade, serem exemplos de seres humanos, e por estarem presentes nos vários momentos de alegria vividos, e também nos momentos de dificuldade.

Aos demais familiares, sobretudo às minhas avós Zélia e Maria Adeile, pelo amor demonstrado a mim todos os dias e por manifestarem, de forma sincera, satisfação e alegria em ver minhas conquistas.

À minha namorada Luana Paula (Lulu), por ter me oferecido seu ombro em um momento em que precisava de alguém em quem me apoiar. Agradeço-lhe por me fazer olhar a vida de uma forma diferente, amadurecer, valorizar as pessoas que prezam por mim e por me ensinar que não há nada mais importante na vida do que amar, principalmente amar-se.

Aos amigos da UFU, da dança e aos amigos de infância, sobretudo o Dr. Carlos Henrique (Carlão), que me acompanham e me apoiam sobretudo nos momentos difíceis.

Aos companheiros do MAPL (*Manufacturing Automated Planning Lab*), agradeço a amizade e a disposição em ajudar sempre que eu precisava. Agradecimentos especiais aos amigos Alexandre (Xandão) e Rodrigo (Didigol) por estarem ao meu lado durante todo o tempo em que sou membro do laboratório, oferecendo-me suporte técnico e principalmente amizade; e ao amigo Hiroshi pela confiança ao me permitir lecionar as aulas práticas de controle digital.

Ao professor Dr. Leonardo Sanches (Leo), por ter aceito o desafio de me orientar a menos de 1 ano do término do mestrado, pelo conhecimento e disposição em ensinar e também aprender juntamente comigo sobre essa área fascinante que é o controle inteligente. Além disso, agradeço ao Leo a confiança no meu trabalho e na minha capacidade.

Ao professor Dr. Jean Tavares, que me proporcionou várias oportunidades de crescimento pessoal e profissional, não só na pós-graduação, mas também em boa parte do meu curso de graduação em Engenharia Mecatrônica.

À professora Dra. Rosana Jafelice, por ser exemplo em minha vida de seriedade, e compromisso com a profissão a qual almejo. Agradeço-lhe também por ter auxiliado a despertar em mim o desejo de contribuir com o progresso científico e principalmente com o desenvolvimento social do país através da educação.

À Universidade Federal de Uberlândia e à Faculdade de Engenharia Mecânica pela oportunidade de cursar uma pós-graduação de excelência. Agradeço também ao CNPq, CAPES e Fapemig pelo suporte financeiro.

PEREIRA, B.L. APLICAÇÃO DE SISTEMAS NEURO-FUZZY E EVOLUÇÃO DIFERENCIAL NA MODELAGEM E CONTROLE DE VEÍCULO DE DUAS RODAS. 2017. Dissertação de Mestrado – Mestrado Acadêmico em Engenharia Mecânica, Universidade Federal de Uberlândia, Uberlândia, 2017.

RESUMO

Esse trabalho propõe a modelagem e o controle neuro-fuzzy aplicados na estabilidade estática de um veículo de duas rodas do tipo pêndulo invertido, utilizando como método de otimização a evolução diferencial. Durante a fase de modelagem, determinam-se as incertezas relacionadas aos parâmetros e também à resposta do modelo neuro-fuzzy. Verifica-se que este é capaz de se ajustar satisfatoriamente aos dados extraídos experimentalmente do veículo. Na determinação do controlador neuro-fuzzy, testam-se três estratégias de ajuste de parâmetros, sendo duas delas propostas neste texto, e os resultados são comparados entre si e aos obtidos através de controladores clássicos, e verifica-se experimentalmente e por meio de testes estatísticos que as abordagens propostas apresentam grande capacidade de adaptação às restrições impostas à planta, garantindo a estabilidade estática e a eficiência energética do sistema.

Palavras-Chave: Sistema Neuro-Fuzzy, Modelagem Neuro-Fuzzy, Controle Neuro-Fuzzy, Evolução Diferencial, Controle Inteligente, Veículo de Duas Rodas.

PEREIRA, B.L. **APPLICATION OF NEURO-FUZZY SYSTEMS AND DIFFERENTIAL EVOLUTION IN TWO-WHEELED VEHICLE MODELING AND CONTROL.** 2017. Masters Dissertation – MSc in Mechanical Engineering, Universidade Federal de Uberlândia, Uberlândia, 2017.

ABSTRACT

This work proposes the neuro-fuzzy modeling and control applied to the static stability of a two-wheeled inverted pendulum vehicle, using differential evolution as optimization technique. During the modeling phase, the uncertainties related to the parameters and also to the neuro-fuzzy model response are determined. It is possible to verify that the neuro-fuzzy system is capable of satisfactorily adjusts to the data experimentally extracted from the vehicle. In the determination of the neuro-fuzzy controller, three strategies of parameter adjustment are tested, two of them being proposed in this text, and the results are compared between them and those obtained through classical controllers, and it is verified experimentally and through tests that the proposed approaches present a great capacity to adapt to the constraints imposed on the plant, guaranteeing the static stability and the energy efficiency of the system.

Keywords: Neuro-Fuzzy System, Neuro-Fuzzy Modeling, Neuro-Fuzzy Control, Differential Evolution, Intelligent Control, Two-wheeled vehicle.

Lista de Figuras

Figura 1.1: Ilustração de um veículo de duas rodas do tipo pêndulo invertido.	16
Figura 2.1: Representação da arquitetura de uma rede neural artificial multicamadas (Silva, 2006).	23
Figura 2.2: Esquema de um neurônio i da camada I (Silva, 2006).	24
Figura 2.3: Funções de ativação usualmente utilizadas.	25
Figura 2.4: Relação temporal entre os instantes de execução do controle e o número de iterações realizadas para a atualização da rede (Silva, 2006).	26
Figura 2.5: Princípio do controle indireto adaptativo (Landau et al., 2011).....	27
Figura 2.6: Esquema básico da estimativa on-line dos parâmetros da rede neural.	28
Figura 2.7: Esquema do controle direto adaptativo (Landau et al., 2011).	29
Figura 2.8: Ilustração da arquitetura de um SBRF (Jafelice, 2003).	30
Figura 2.9: Ilustração do método de Mamdani (Jafelice, 2003).	31
Figura 2.10: Representação de um sistema neuro-fuzzy.	32
Figura 2.11: Representação da evolução diferencial, adaptada de Mór et al. (2015).....	34
Figura 2.12: Modelo obtido a partir da técnica de regressão, adaptada de Berland (2007) ..	36
Figura 2.13: Representação de um processo estocástico.....	40
Figura 2.14: Estimação das propriedades de uma população.	41
Figura 2.15: Testes de hipóteses.	41
Figura 2.16: Esquema simplificado dos testes de hipóteses.....	42
Figura 2.17: Representação de uma planta que segue o princípio do pêndulo invertido.....	43
Figura 2.18: (a) Veículo de duas rodas e (b) o circuito elétrico de seus motores (Bonafilia et al., 2014).	44
Figura 2.19: Diagramas de corpo livre (a) da estrutura do veículo e das rodas (b) (Bonafilia et al., 2014).....	44
Figura 2.20: Representação da zona morta.....	48
Figura 3.1: Veículo de duas rodas – (a) foto e (b) representação dos seus componentes....	51
Figura 3.2: (a) Dados de obtenção e (b) validação do modelo.	53
Figura 3.3: Evolução do erro médio quadrático no tempo.	53
Figura 4.1: Estrutura do modelo neuro-fuzzy.....	56
Figura 4.2: (a) Ação de controle e (b) ângulo de inclinação utilizados como dados de obtenção do modelo.	57
Figura 4.3: (a) Ação de controle e (b) ângulo de inclinação utilizados como dados de validação do modelo.	57
Figura 4.4: Funções de pertinência das variáveis de entrada (a) $uN, k + 1$, (b) $\theta N, k$ e (c) $\Delta\theta N, k$	58
Figura 4.5: (a) Dados de obtenção do modelo e (b) dados de validação.	60
Figura 4.6: Evolução do processo de identificação dos parâmetros do modelo.	60
Figura 4.7: MPU6050.....	61
Figura 4.8: Foto do esquadro de precisão.	62
Figura 4.9: Desvio de perpendicularidade do esquadro de precisão.	62
Figura 4.10: Histograma da saída do modelo do sensor.	66

Figura 4.11: Representação da FDP da saída do modelo.	68
Figura 4.12: Histograma da saída do modelo da planta.	71
Figura 4.13: Resposta do modelo neuro-fuzzy.	72
Figura 5.1: Representação do sistema em malha fechada.	75
Figura 5.2: Funções de pertinência das variáveis de entrada (a) uN, k , (b) $\theta N, k$ e (c) $\Delta\theta N, k$	76
Figura 5.3: (a) Ação de controle normalizada e (b) ângulo de inclinação do veículo para $\delta = 0 V$	78
Figura 5.4: (a) Ação de controle normalizada e (b) ângulo de inclinação do veículo para $\delta = 0,6 V$	78
Figura 5.5: (a) Ação de controle normalizada e (b) ângulo de inclinação do veículo para $\delta = 1,2 V$	79
Figura 5.6: (a) Ação de controle normalizada e (b) ângulo de inclinação do veículo para $\delta = 1,8 V$	79
Figura 5.7: Evolução de C_f no processo de otimização dos parâmetros do controlador.....	80
Figura 5.8: (a) Ação de controle e (b) ângulo de inclinação experimentais do veículo de duas rodas utilizando o controlador DEDMFN.	81
Figura 5.9: (a) Ação de controle e (b) ângulo de inclinação simulados do veículo de duas rodas.....	83
Figura 5.10: Evolução de C_f no processo de otimização dos parâmetros do controlador....	83
Figura 5.11: (a) Ação de controle e (b) ângulo de inclinação experimentais do veículo de duas rodas utilizando o controlador DENFNF.	84
Figura 5.12: Fluxograma da etapa 1 do algoritmo OnMoDE.....	85
Figura 5.13: Fluxograma da etapa 2 do algoritmo OnMoDE.....	86
Figura 5.14: (a) Ação de controle e (b) ângulo de inclinação simulados do veículo de duas rodas.....	89
Figura 5.15: Evolução do valor da função custo no tempo obtida durante a simulação.....	90
Figura 5.16: (a) Ação de controle e (b) ângulo de inclinação experimentais do veículo de duas rodas.....	90
Figura 5.17: Evolução do valor da função custo no tempo obtida experimentalmente durante a operação do veículo de duas rodas.	91
Figura 5.18: (a) Ação de controle e (b) ângulo de inclinação experimentais do veículo de duas rodas utilizando o controlador proporcional.....	92
Figura 5.19: (a) Ação de controle e (b) ângulo de inclinação experimentais do veículo de duas rodas utilizando o controlador ZNPID.	92
Figura 5.20: (a) Ação de controle e (b) ângulo de inclinação experimentais do veículo de duas rodas utilizando o controlador RLPID.....	93
Figura 7.1: Aeromodelo construído pela equipe Tucano.	100
Figura A.1: Ângulo de inclinação extraído com $T_s = 0,002 s$	111
Figura A.2: Amplitude do ângulo de inclinação em função da frequência.	112

Lista de Tabelas

Tabela 2.1: Parâmetros do processo.....	45
Tabela 3.1: Valor dos Parâmetros do Processo.....	54
Tabela 4.1: Base de Regras do Modelo Neuro-Fuzzy.	59
Tabela 4.2: Variáveis de Entrada.....	66
Tabela 4.3: Incerteza da Resposta da IMU.....	67
Tabela 4.4: Incerteza da Resposta do Modelo Neuro-Fuzzy.	71
Tabela 5.1: Base de Regras do Controlador Neuro-Fuzzy.....	77
Tabela 5.2: Desempenho do sistema em função de δ	94
Tabela 5.3: Desempenho dos controladores.	95

Lista de Símbolos

ANFIS – *Adaptive Neuro-Fuzzy Inference System.*

ANN – *Artificial Neural Network.*

CENIPA – Centro de Investigação e Prevenção de Acidentes Aeronáuticos.

DCLs – Diagramas de Corpo Livre.

DEDMNF – *Differential Evolution and Dynamical Model Based Neuro-Fuzzy Controller.*

DENFNF – *Differential Evolution and Neuro-Fuzzy Model Based Neuro-Fuzzy Controller.*

FDP – Função Densidade de Probabilidade.

FIS – *Fuzzy Inference System.*

GEEs – Gases de Efeito Estufa.

GUM – *Guide to the Expression of Uncertainty in Measurement.*

IMU – *Inertial Measurement Unit.*

MAPL – *Manufacturing Automated Planning Lab.*

MSE – *Mean Square Error.*

OnMoDE – *Online Modified Differential Evolution.*

PID – Controlador Proporcional Integral Derivativo.

RLPID – Controlador Proporcional Integral Derivativo Projetado a Partir do Método do Lugar das Raízes.

RNA – Rede Neural Artificial.

SBRF – Sistema Baseado em Regras Fuzzy.

ZNPID – Controlador Proporcional Integral Derivativo Projetado a Partir do Método de Ziegler-Nichols.

Sumário

CAPÍTULO I	15
1.1. Objetivos.....	18
1.2. Justificativa.....	19
CAPÍTULO II	21
2.1. Sistemas e Controle Não-Lineares	21
2.2. Redes Neurais	23
2.2.1. Conceitos Gerais	23
2.2.2. Treinamento da Rede Neural.....	25
2.3. Teoria dos Conjuntos Fuzzy.....	29
2.4. Sistema Neuro-Fuzzy	32
2.5. Evolução Diferencial.....	34
2.6. Regressão	36
2.7. Método de Monte Carlo.....	39
2.8. Análise Estatística dos Resultados.....	41
2.9. Modelagem de Veículo de Duas Rodas do Tipo Pêndulo Invertido.....	43
2.10. Zona Morta de Atuadores.....	48
CAPÍTULO III	50
3.1. Descrição da Bancada Experimental	50
3.2. Modelagem da Bancada Experimental	51
CAPÍTULO IV	55
4.1. Obtenção do Modelo Neuro-Fuzzy	55
4.2. Estudo das Incertezas Associadas aos Parâmetros do Modelo Neuro-Fuzzy	61
4.2.1. Estudo da Incerteza Associada à Resposta da <i>IMU</i>	61
4.2.2. Determinação dos Intervalos de Confiança dos Parâmetros do Modelo Neuro-Fuzzy	67
4.3. Estudo da Incerteza Associada à Resposta do Modelo Neuro-Fuzzy.....	69
CAPÍTULO V	73
5.1. Sintonização a Partir do Modelo Dinâmico	74
5.2. Sintonização a Partir do Modelo Neuro-Fuzzy	81
5.3. Sintonização a Partir do Método OnMoDE	84
5.4. Controladores Clássicos.....	91

5.5. Análise Estatística dos Resultados	94
CAPÍTULO VI	97
CAPÍTULO VII	99
REFERÊNCIAS BIBLIOGRÁFICAS	101
APÊNDICES	106
A) Filtro de Kalman	106
A.1 Expansão do Método para Problemas com Mais de Uma Variável de Estado e Equações do Filtro de Kalman	107
A.2 Determinação dos Parâmetros do Filtro de kalman	108
A.2.1 Método Analítico	109
A.2.2 Método Numérico	110
B) Estudo e Determinação do Tempo de Amostragem Adotado para o Controle do Veículo de Duas Rodas	111
C) Identificação do Modelo Dinâmico - MATLAB	113
D) Identificação do Modelo Neuro-Fuzzy - MATLAB	124
E) Determinação da Incerteza Associada à Resposta da MPU6050 – MATLAB	133
F) Determinação da Incerteza Associada à Resposta do Modelo Neuro-Fuzzy – MATLAB	138
G) Obtenção dos Parâmetros do Controlador DENFNF – MATLAB	143
H) Controlador DENFNF – Arduino	152
I) Método OnMoDE – Arduino	158

CAPÍTULO I

INTRODUÇÃO

As técnicas de controle inteligente baseiam-se na observação de como os seres vivos, e em particular os seres humanos, se comportam enfrentando e resolvendo diferentes problemas. Estas técnicas são geralmente versáteis e ajustadas para controlar diferentes sistemas dinâmicos, mesmo sob qualquer variação de suas propriedades. De fato, estas variações podem acontecer, como por exemplo, devido ao envelhecimento ou à falha de componentes mecânicos. Nesse sentido, os parâmetros de controle precisam ser acessados e adaptados em tempo real ou em operação off-line.

O aumento da capacidade de processamento dos microcontroladores e plataformas de prototipagem viabiliza o uso de técnicas de otimização e de controle inteligente aplicadas em sistemas dinâmicos com múltiplas variáveis, com acoplamento entre elas, com presença de fortes não linearidades e com variações rápidas dos parâmetros, circunstâncias nas quais os controladores clássicos apresentam baixa eficácia.

De forma análoga ao princípio de funcionamento de um pêndulo, um veículo de duas rodas do tipo pêndulo invertido, Fig. 1.1, é caracterizado por apresentar oscilação do seu centro de massa CM em torno de sua posição de equilíbrio, porém, ao contrário do pêndulo tradicional, o sistema em questão é marginalmente estável, pois qualquer perturbação submetida à estrutura é capaz de repelir a planta com relação à sua posição de equilíbrio. O principal objetivo dos controladores aplicados a plantas que seguem o princípio do pêndulo invertido está em minimizar o ângulo de inclinação θ bem como a translação x do veículo, garantindo assim sua estabilidade estática.

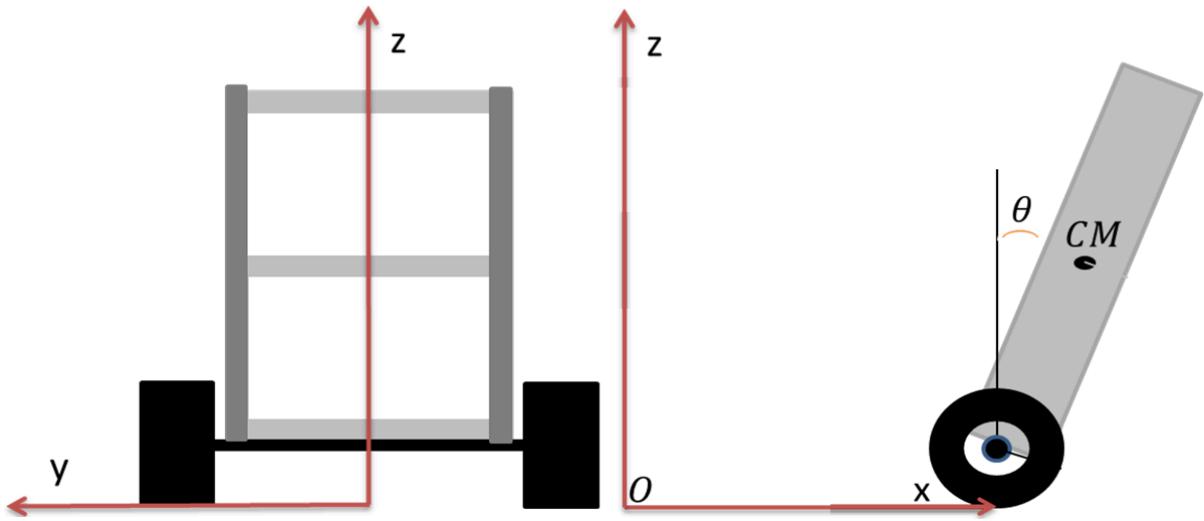


Figura 1.1: Ilustração de um veículo de duas rodas do tipo pêndulo invertido.

O estudo de metodologias de controle aplicadas a veículos que apresentam o princípio de funcionamento do pêndulo invertido é amplamente realizado (Boskovich, 1995; Salerno e Angeles, 2007; Nawawi, Ahmad e Osman, 2008; Su, Chen e Su, 2010; Juang e Lum, 2013; Bonafilia et al., 2014; Xu, Guo e Lee, 2014; Pereira et al., 2016). Devido à dinâmica não-linear deste sistema, ele é aplicado no levantamento de *benchmarks* relativos a eficiência de controladores em sistemas não-lineares.

Os atuadores de veículos de duas rodas, geralmente dotados de motores DC, também estão sujeitos a não-linearidades, dentre as quais pode-se citar a folga, a saturação e a zona morta. Estas características exercem forte influência no comportamento dinâmico destes sistemas no que se refere à sua estabilização e no aumento de sua amplitude de oscilação (Peng e Dubay, 2011).

Dentre as técnicas de controle inteligente, utiliza-se o sistema neuro-fuzzy, proposto por Jang (1992a), dado à sua característica de adaptação da rede neural aos problemas (Haykin, 2008), com a robustez e intuitividade da teoria dos conjuntos fuzzy (Zadeh, 1965). O sistema neuro-fuzzy vem sendo utilizado, dessa forma, tanto na modelagem de sistemas (Tanaka, Sano e Watanabe, 1995; Zhang e Morris, 1999; Chen e Linkens, 2001; Shen et al., 2002; Babuška e Verbruggen, 2003; Hasiloglu et al., 2004; Nayak et al., 2004; Aqil et al., 2006; Buragohain e Mahanta, 2007; Partal e Kisi, 2007), quanto no controle de sistemas dinâmicos (Jang, 1992a; Jang, 1992b; Godjevac e Steele, 1999; Grabowski et al., 2000; Denaï, Palis e Zeghbib, 2004; Tao et al., 2005; Saifizul et al., 2006; Chen, Lin e Lin, 2008; Orlowska-Kowalska e Szabat, 2008; Rusu e Petriu, 2008; Tsai, Huang e Lin, 2008; Wannige e

Sonnadara, 2009; Su, Chen e Su, 2010; Tatikonda, Battula e Kumar, 2010; Su et al., 2011; Han, Lin e Chang, 2013; Nejadfar, Yazdanpanah e Hassanzadeh, 2013; Petchinathan et al., 2014).

Na literatura são encontrados projetos de controladores neuro-fuzzy em que o modelo dinâmico da planta é conhecido, e o sistema de controle é obtido via simulações computacionais (Jang, 1992a, Jang, 1992b; Grabowski et al., 2000; Saifizul et al., 2006; Chen, Lin e Lin, 2008; Orlowska-Kowalska e Szabat, 2008; Tsai, Huang e Lin, 2008; Su, Chen e Su, 2010; Tatikonda, Battula e Kumar, 2010; Su et al., 2011; Han, Lin e Chang, 2013). Em outros trabalhos, propõe-se a estimativa dos parâmetros do controlador através de dados experimentais a partir da definição da função pseudo-inversa da planta (Jang e Sun, 1995; Petchinathan et al., 2014). Há também trabalhos que propõem a sintonização dos parâmetros do controlador neuro-fuzzy a partir do método analítico da retropropagação do erro, este definido como a diferença entre a saída do sistema e o valor desejado da saída, sendo a sintonização realizada de modo on-line (Godjevac e Steele, 1999; Rusu e Petriu, 2008; Wannige e Sonnadara, 2009; Nejadfar, Yazdanpanah e Hassanzadeh, 2013), ou via estimativa primeiramente do modelo da planta a partir de algum método de aproximação, como por exemplo, o próprio sistema neuro-fuzzy (Denaï, Palis e Zeghbib, 2004; Tao et al., 2005; Petchinathan et al., 2014).

Devido à dificuldade em se obter um modelo analítico de um veículo de duas rodas do tipo pêndulo invertido, bem como um controlador para tal, que levem em consideração simultaneamente as não-linearidades da estrutura e dos atuadores, e que permitam que seus parâmetros possam ter seus valores alterados em tempo real de acordo com as necessidades e requisitos do sistema, destaca-se a utilização da modelagem e controle neuro-fuzzy da planta.

Para a otimização dos parâmetros do modelo e do controlador neuro-fuzzy, utiliza-se a evolução diferencial (Storn e Price, 1996; Chen, Lin e Lin, 2008; Han, Lin e Chang, 2013) devido à sua relativa simplicidade de implementação, rapidez e maior facilidade em se alcançar o valor ótimo global em relação aos métodos clássicos e aos demais métodos evolutivos.

A modelagem neuro-fuzzy baseia-se nos conhecimentos qualitativos e quantitativos de que se dispõe da planta (obtidos empiricamente), logo, da mesma forma que em qualquer outro tipo de modelagem experimental, é essencial que as incertezas relacionadas aos parâmetros e também à resposta do modelo sejam determinadas.

O controlador neuro-fuzzy, aplicado na estabilidade estática do veículo, é projetado a partir de três estratégias de ajuste de parâmetros:

- Controlador neuro-fuzzy ajustado a partir da utilização do modelo dinâmico apresentado no capítulo 3;
- Controlador neuro-fuzzy ajustado a partir da utilização do modelo neuro-fuzzy descrito no capítulo 4;
- Controlador neuro-fuzzy ajustado a partir do método da evolução diferencial modificada e aplicada on-line durante a operação do sistema, denominado método OnMoDE.

As abordagens indicadas são comparadas entre si e também entre os controladores proporcional e PID com diferentes métodos de sintonização. Utilizam-se, como critérios de análise e comparação dos resultados, testes estatísticos (Montgomery e Runger, 2002) que empregam a distribuição F de Snedecor a partir dos dados da ação de controle e do ângulo de elevação extraídos da planta.

As subseções a seguir tratam respectivamente dos objetivos e da justificativa do trabalho. Em sequência, no capítulo II é apresentada a fundamentação teórica dos assuntos pertinentes ao desenvolvimento do trabalho, no capítulo III é apresentado o veículo de duas rodas e sua modelagem dinâmica, no capítulo IV exibem-se a modelagem neuro-fuzzy do sistema e o cálculo das incertezas relacionadas ao processo, no capítulo V os controladores neuro-fuzzy projetados e seus respectivos resultados são indicados, nos capítulos V e VI são expostos respectivamente as conclusões e os trabalhos futuros, e por fim, seguem as referências bibliográficas e apêndices.

1.1. Objetivos

Os objetivos gerais da pesquisa são a modelagem neuro-fuzzy, determinação das incertezas relacionadas ao processo, e o projeto e desenvolvimento de um sistema de controle neuro-fuzzy a ser aplicado na estabilização estática de um veículo de duas rodas, dando ênfase no desenvolvimento dos algoritmos e na contribuição da pesquisa na área de controle inteligente.

Os objetivos específicos do projeto são:

- Instrumentação e coleta de dados do veículo de duas rodas;
- Apresentação da modelagem neuro-fuzzy do veículo;

- Determinação das incertezas relacionadas aos parâmetros do modelo neuro-fuzzy;
- Determinação da incerteza da resposta do modelo neuro-fuzzy;
- Projeto de um controlador neuro-fuzzy aplicado na estabilização estática de um veículo de duas rodas;
- Desenvolvimento de estratégias de ajuste dos parâmetros do controlador neuro-fuzzy;
- Realização de simulações para verificar a capacidade do controlador neuro-fuzzy de estabilizar o veículo de duas rodas;
- Implementação do controlador em microcontrolador para validação experimental da técnica de controle; e
- Execução de testes estatísticos de modo a se analisar e comparar as estratégias de ajuste dos parâmetros dos sistemas neuro-fuzzy.

1.2. Justificativa

O estudo de metodologias para o controle de veículos de duas rodas se justifica cientificamente devido à inexistência de solução analítica ótima em função da característica de não-linearidade da planta, sendo dessa forma utilizado pela comunidade científica no levantamento de benchmarkings relativos à eficiência de controladores aplicados a sistemas não-lineares.

A proposição e aplicação de estratégias na área do controle inteligente, devido à sua versatilidade e capacidade de adaptação durante a operação dos sistemas dinâmicos, justifica-se tendo em vista que o modelo dinâmico de um veículo de duas rodas leva em consideração uma série de parâmetros muitas vezes difíceis de serem calculados ou obtidos. Além disso, alguns destes estão submetidos a variações durante a operação da planta, provocadas pelo desgaste de componentes, a variação da posição do centro de gravidade, a redução gradativa da tensão da bateria, dentre outros fatores que garantem o comportamento não-linear e variante com o tempo do sistema.

Na literatura, não existem métodos precisos para a obtenção do intervalo de confiança dos parâmetros obtidos experimentalmente de um sistema neuro-fuzzy, nem há estudos relativos à propagação dessas incertezas sob a resposta do sistema (Tanaka, Sano e Watanabe, 1995; Zhang e Morris, 1999; Chen e Linkens, 2001; Shen et al., 2002; Babuška e Verbruggen, 2003; Hasiloglu et al., 2004; Nayak et al., 2004; Aqil et al., 2006; Buragohain e Mahanta, 2007; Partal e Kisi, 2007). Assim, essa pesquisa também se justifica ao propor a introdução e aplicação deste assunto dentro do contexto do trabalho.

Por fim, o setor de transportes é um dos maiores responsáveis pelas emissões de gases de efeito estufa no Brasil e no mundo, e Rovere e Wills (2014) verificou que a implantação de uma meta nacional de eficiência energética equiparada à meta europeia para 2021 – em que, dentre outras mudanças, há o investimento em meios de transporte mais limpos, energeticamente eficientes e com o uso de energias renováveis – reduziria o consumo de energia dos automóveis em 24,5% e das emissões de GEEs em 23,9% em 2030. O estudo conclui que, além dos investimentos necessários em meios mais eficientes de transporte, como o ferroviário e o hidroviário, as metas de eficiência energética veicular, de forma complementar, podem auxiliar a transformar o Brasil em uma economia mais eficiente, sustentável, e de baixo carbono. Seguindo essa tendência, propõe-se o desenvolvimento de novas metodologias de controle de veículos de duas rodas, que podem ser utilizados como meios de transporte mais eficientes.

CAPÍTULO II

FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica deste trabalho consiste em fornecer ao leitor o conhecimento necessário para o entendimento dos conceitos e aplicações desenvolvidas, sendo dividida em sistemas e controle não-lineares, redes neurais, teoria dos conjuntos fuzzy, sistema neuro-fuzzy, evolução diferencial, regressão, método de Monte Carlo, análise estatística dos resultados e modelagem de veículo de duas rodas do tipo pêndulo invertido.

2.1. Sistemas e Controle Não-Lineares

O controle não-linear é definido como sendo o conjunto de procedimentos destinados à estabilização ou ao rastreamento das variáveis de saída de um sistema não-linear em relação a uma determinada referência.

Sistemas não-lineares apresentam comportamentos que os diferem dos sistemas lineares, e dentre estes comportamentos, os principais são indicados a seguir (Silva, 2006):

- Dependência da amplitude de excitação;
- Tempo de escape finito;
- Pontos de equilíbrio múltiplos;
- Não unicidade da solução;
- Elevada sensibilidade aos parâmetros e às condições iniciais (caos);

- Existência de ciclos limite de oscilação;
- Respostas harmônicas e sub-harmônicas a uma entrada senoidal.

De modo a facilitar a análise e controle de sistemas dessa natureza, correntemente é aplicado sobre os mesmos o processo de linearização. Dada a equação diferencial abaixo, admitindo que a função $f[x(t), u(t)]$ seja diferenciável, sendo $x(t)$ o estado do sistema e $u(t)$ a ação de controle aplicada ao mesmo, ambos no tempo t :

$$\dot{x}(t) = f[x(t), u(t)] \quad (2.1)$$

esta pode ser expandida em série de Taylor em torno de um ponto $P = (x_p, u_p)$. Ao se realizar a expansão em torno de um ponto de equilíbrio (x_e, u_e) tal que $x_e = 0$ e $u_e = 0$, tem-se que:

$$\dot{x} = \left(\frac{\partial f}{\partial x} \right)_{x=0, u=0} x + \left(\frac{\partial f}{\partial u} \right)_{x=0, u=0} u + f_{so}(x, u) \quad (2.2)$$

$$\dot{x} = Ax + Bu + f_{so}(x, u).$$

Desprezando os termos de ordem superior da expansão em série de Taylor f_{so} , realiza-se a linearização do sistema em torno do ponto de equilíbrio (x_e, u_e) .

Após o processo de linearização, pode ser necessário apresentar o sistema dinâmico na forma discreta, permitindo que o mesmo seja utilizado no projeto de controladores digitais. A Eq. (2.3) apresenta o sistema na forma discreta, considerando o seguidor de ordem zero, ou seja, mantendo valores constantes de ação de controle u_k durante o intervalo de tempo T_s , sendo T_s o tempo de amostragem adotado para o sistema e k a iteração corrente.

$$x_{k+1} = Gx_k + Hu_k. \quad (2.3)$$

As matrizes G e H são determinadas através das Eqs. (2.4) e (2.5):

$$G = e^{AT_s} \quad (2.4)$$

$$H = A^{-1}(e^{AT_s} - I)B \quad (2.5)$$

em que I é a matriz identidade.

A partir do sistema linearizado, é possível aplicar as teorias de controle clássico. Contudo, ao se realizar o processo de linearização, algumas características do sistema passam a ser desprezadas, sobretudo quando se afasta do ponto de equilíbrio em que foi realizada a linearização.

Existem técnicas aplicadas em controle, entretanto, como as redes neurais, a teoria dos conjuntos fuzzy e os sistemas neuro-fuzzy, que levam em consideração as não-

linearidades do sistema, e dessa forma são importantes objetos de estudo na área de controle de sistemas não-lineares.

2.2. Redes Neurais

2.2.1. Conceitos Gerais

Uma rede neural artificial, RNA – Fig. 2.1, é uma estrutura matemática constituída por um número finito de unidades individualizadas, também designadas por neurônios, Fig. 2.2, organizados em camadas (Silva, 2006). A rede neural é utilizada, dentre outras aplicações, para o reconhecimento de padrões, para a obtenção de modelos para sistemas estáticos e dinâmicos e também como estrutura matemática de controladores, apresentando a característica de ser adequada a uma série de problemas devido à sua capacidade de ajuste e adaptação aos problemas.

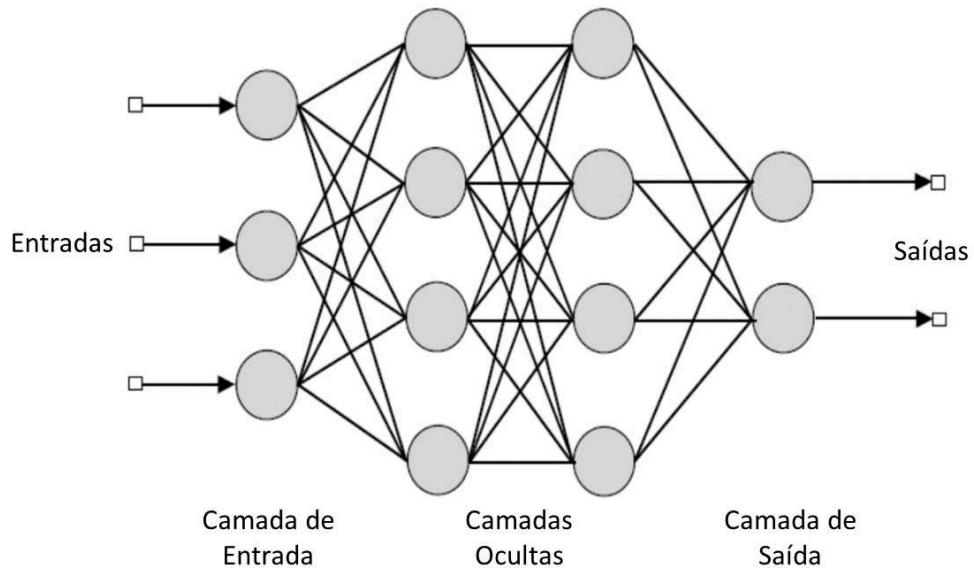


Figura 2.1: Representação da arquitetura de uma rede neural artificial multicamadas (Silva, 2006).

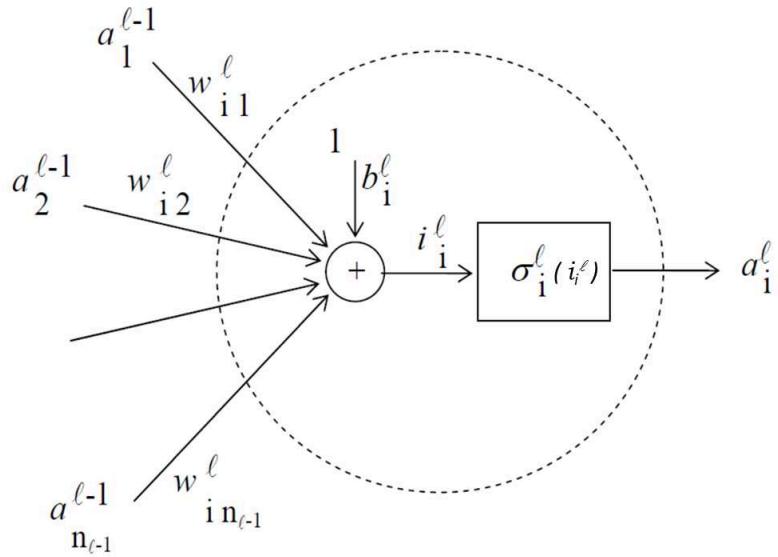


Figura 2.2: Esquema de um neurônio i da camada l (Silva, 2006).

A saída a_i^l de cada neurônio i da camada l segue a Eq. (2.6), sendo σ a função de ativação da rede neural, cuja saída é limitada geralmente ao intervalo $[-1, 1]$ e é a responsável por prover à estrutura matemática a característica desejada de não-linearidade. Os parâmetros w_{ij}^l e b_i^l são respectivamente o peso sináptico e o valor do *bias* associado ao neurônio i , ambos determinados durante o processo de treinamento da rede neural, assunto abordado na próxima subseção.

$$i_i^l = \sum_{j=1}^{n_{l-1}} w_{ij}^l a_j^{l-1} + b_i^l$$

$$a_i^l = \sigma_i^l(i_i^l) \quad (2.6)$$

A Fig. 2.3 é a representação das funções de ativação σ usualmente utilizadas, sendo elas degrau, linear, sigmoide ou tangente hiperbólica.

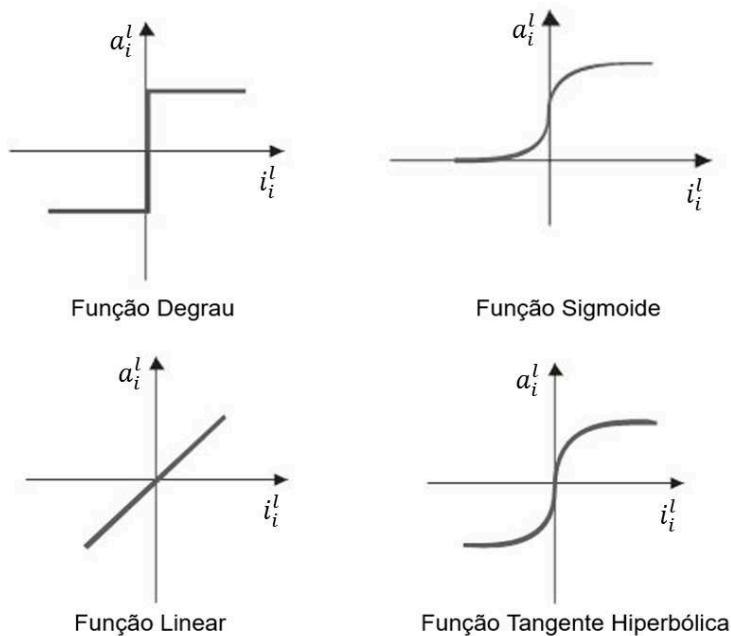


Figura 2.3: Funções de ativação usualmente utilizadas.

2.2.2. Treinamento da Rede Neural

Após a fixação do número de camadas e do número de neurônios em cada camada, procedimento realizado geralmente de forma intuitiva, é necessário determinar os valores dos parâmetros w_{ij}^l e b_i^l que conduzem ao melhor desempenho da rede, processo denominado treinamento da rede neural.

O treinamento da rede neural é tradicionalmente executado através de métodos de otimização inspirados nos métodos da máxima descida e de métrica variável – embora haja na literatura trabalhos em que métodos heurísticos, como a evolução diferencial e os algoritmos genéticos, passam a ser utilizados para tal função (Pereira et al., 2016) – e baseia-se na minimização da função custo C_f , Eq. (2.7):

$$C_f = \sum_{k=1}^{Nd} [y(k) - \hat{y}(k)]^2 \quad (2.7)$$

em que $y(k)$, $k=1, \dots, Nd$; sendo Nd o número de elementos da sequência $y(k)$, são o conjunto de dados utilizados para o treinamento da rede, e $\hat{y}(k)$, $k=1, \dots, Nd$; são os valores obtidos na saída da rede neural. Ao se realizar com sucesso o processo de minimização da função custo, espera-se que a rede neural, para os valores de entrada que geram $y(k)$, apresente valores de saída $\hat{y}(k)$ tais que $\hat{y}(k) \approx y(k)$, $k=1, \dots, Nd$.

O treinamento da RNA pode ser realizado de duas formas: em grupo, chamado alternativamente de treinamento off-line; e em linha, também chamado de on-line ou em tempo real.

O treinamento em grupo é utilizado para processos invariantes no tempo ou para obter valores iniciais dos parâmetros da rede. O valor de Nd deve ser da ordem das centenas, o que permite ter uma grande variedade de amplitudes e informações do processo. O número de iterações utilizado para obter a convergência da rede é variável e é função da respectiva rapidez de convergência do processo de otimização e do valor aceitável para Cf . Após o treino em grupo, as redes devem ser testadas utilizando outras sequências de entrada/saída para verificar a sua boa capacidade de generalização.

Já o treinamento em linha é necessário quando se precisa identificar um processo variante no tempo. Ressalta-se que para garantir a rápida adaptação da rede ao sistema, o número de iterações realizadas durante o treinamento por cada passo de execução do controle deverá ser no mínimo na ordem de dezenas, conforme é ilustrado na Fig. 2.4. Em sistemas não-lineares, porém, o treinamento em linha pode causar o esquecimento das características globais da rede, ficando esta mais adaptada apenas para pontos de funcionamento com características locais.

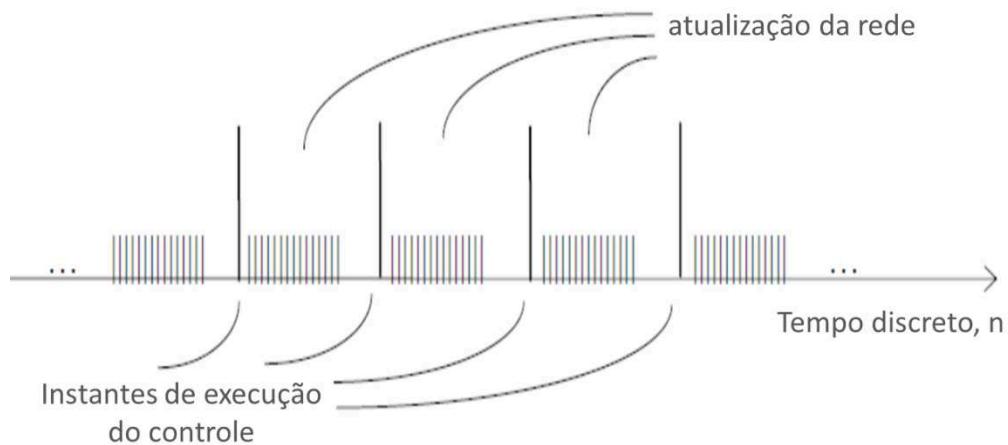


Figura 2.4: Relação temporal entre os instantes de execução do controle e o número de iterações realizadas para a atualização da rede (Silva, 2006).

2.2.3. Aplicação de Redes Neurais em Sistemas de Controle Adaptativos

De acordo com Haykin (2008), há basicamente duas formas de se trabalhar com redes neurais aplicadas a sistemas de controle adaptativos: controle direto adaptativo e indireto adaptativo.

Nos métodos de **controle indireto adaptativo**, a obtenção dos pesos e bias da rede neural não está ligada diretamente ao controle, mas sim à estimativa da dinâmica da planta, Fig. 2.5.

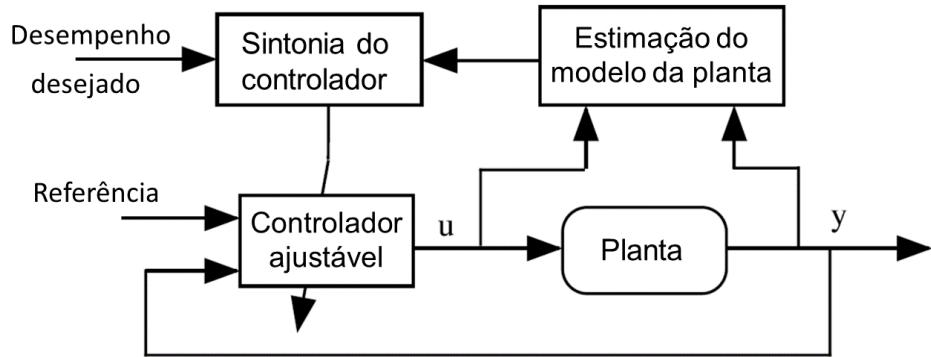


Figura 2.5: Princípio do controle indireto adaptativo (Landau et al., 2011).

Neste método de controle é utilizado um procedimento subdividido em 2 passos, a saber:

- Obtém-se um modelo da planta P através de uma amostra contendo dados de entrada e saída da mesma. Para exemplificar, considere um sistema de uma entrada $u(k)$ e uma saída $y(k)$ para o tempo discreto k . O modelo de identificação da rede neural artificial com realimentação da entrada, \hat{P} , pode ser dado da forma:

$$\hat{y}(k+1) = f(y(k), \dots, y(k-q+1), u(k), \dots, u(k-q+1)) \quad (2.8)$$

em que q é a ordem do sistema desconhecido e \hat{y} é a estimativa da saída y . A estimativa $\hat{y}(k+1)$ é então subtraída de $y(k+1)$ (saída real do sistema, obtida por meio de sensores) para produzir o sinal de erro:

$$e(k+1) = y(k+1) - \hat{y}(k+1). \quad (2.9)$$

O erro $e(k+1)$ é usado para ajustar os pesos sinápticos da rede neural, de modo a minimizar o erro na estimativa da saída y durante o processo de treinamento da estrutura. Ressalta-se que é a saída real do sistema (e não a saída do modelo de identificação) que é realimentada para a entrada do modelo. A Fig. 2.6 ilustra o procedimento de obtenção do modelo de identificação \hat{P} , considerando-se que o sistema apresenta ordem $q=1$.

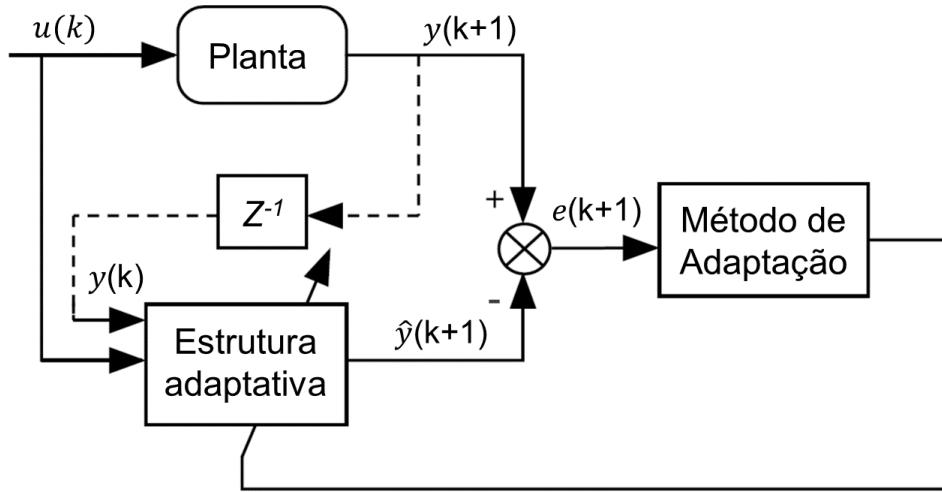


Figura 2.6: Esquema básico da estimação on-line dos parâmetros da rede neural.

- O modelo identificado da planta é utilizado para se obter as estimativas da saída do sistema em função da entrada imposta no mesmo. Dessa forma, a partir deste, é possível projetar controladores adaptativos, conforme verificado na Fig. 2.5, que obtêm o conjunto de ações de controle capaz de garantir que o sistema controlado atinja o desempenho desejado.

Já nos métodos de **controle direto adaptativo**, a dinâmica da planta é desconhecida, e dessa forma o controlador e a planta formam um sistema realimentado de laço fechado cujas entradas são o *setpoint* e uma versão atrasada da saída da planta; e a saída do sistema de controle é a correspondente ação de controle a ser aplicada no mesmo, Fig. 2.7.

Em (Haykin, 2008), os valores dos pesos e dos *bias* são diretamente ajustados para melhorar o desempenho do sistema, embora não existam métodos precisos para ajustar estes parâmetros, visto que a planta desconhecida encontra-se entre o controlador e o erro de saída.

O esquema do controle direto adaptativo, apresentado na Fig. 2.7, baseia-se na observação de que a diferença entre a saída da planta e a saída do modelo de referência, chamada subsequentemente de erro planta-modelo, é uma medida comparativa entre o desempenho real do controlador e o desejado. Esta informação, juntamente com outras, é utilizada pelo método de adaptação para ajustar diretamente em tempo real os parâmetros do controlador adaptativo, de modo a forçar, de forma assintótica, o erro planta-modelo para zero (Landau et al., 2011).

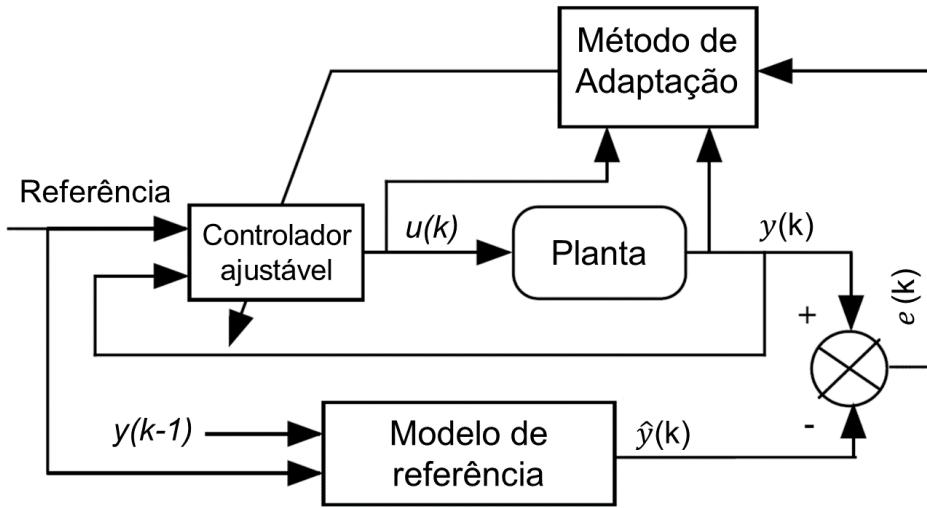


Figura 2.7: Esquema do controle direto adaptativo (Landau et al., 2011).

O controle direto adaptativo, porém, apresenta algumas limitações. Embora o desempenho do sistema possa, em muitos casos, ser especificado em termos de um modelo de referência, as condições para a existência de um controlador viável que permita, em malha fechada, seguir esse modelo são restritivas (Landau et al., 2011).

2.3. Teoria dos Conjuntos Fuzzy

Como referência para estudar a teoria dos conjuntos fuzzy, são utilizados (Jafelice, 2003 e Gomide et al., 2015), e alguns dos principais conceitos concernentes à área são mencionados em sequência.

Conjunto Fuzzy: Um subconjunto fuzzy F do conjunto universo U é definido em termos de uma função de pertinência u , que a cada elemento x de U , associa um número $u(x)$ entre zero e um, chamado de grau de pertinência de x a F . Assim, o conjunto fuzzy F é simbolicamente indicado por sua função de pertinência $u_F : U \rightarrow [0,1]$. Os valores $u_F(x) = 1$ e $u_F(x) = 0$ indicam, respectivamente, a pertinência plena e a não pertinência do elemento x a F .

Sistema Baseado em Regras Fuzzy: Sistemas baseados em regras fuzzy (SBRF) contêm quatro componentes: um processador de entrada que realiza a fuzzificação dos dados de entrada, uma coleção de regras nebulosas chamada base de regras, uma máquina de inferência fuzzy e um processador de saída que fornece um número real como saída. Uma

vez estabelecida uma base de regras, isto é, como relacionamos os conjuntos fuzzy pela forma Se...então..., um SBRF pode ser visto como um mapeamento entre a entrada e a saída da forma $y = f(x)$, $x \in \mathbb{R}^n$ e $y \in \mathbb{R}^m$. A Fig. 2.8 ilustra a arquitetura de um SBRF.

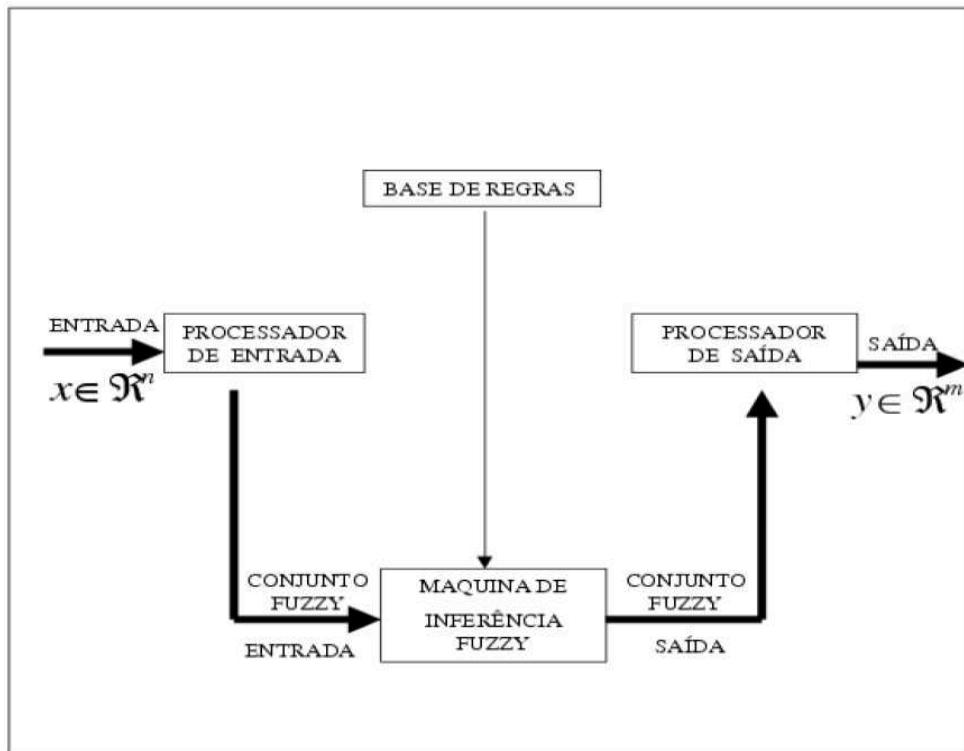


Figura 2.8: Ilustração da arquitetura de um SBRF (Jafelice, 2003).

Esta classe de sistema é amplamente utilizada em problemas de modelagem, controle e classificação. Os componentes do SBRF são descritos a seguir:

- **Processador de Entrada (Fuzzificação):** Neste componente as entradas do sistema são traduzidas em conjuntos fuzzy em seus respectivos domínios. É neste momento em que são construídas as funções de pertinências para a descrição das entradas.
- **Base de Regras:** Este componente, juntamente com a máquina de inferência, pode ser considerado o núcleo dos sistemas baseados em regras fuzzy. Ele é composto por uma coleção de proposições fuzzy na forma ‘Se...então...’.
- **Máquina de Inferência Fuzzy:** É neste componente que cada proposição fuzzy é traduzida matematicamente por meio de métodos de Inferência Fuzzy. Apresentam-se os métodos de Mamdani e de Takagi-Sugeno de Inferência Fuzzy:
 - **Método de Mamdani:** Este método agrupa cada uma das regras – ‘Se (antecedente) então (consequente)’ – através do operador lógico *OU*, que é modelado pelo operador

máximo e, em cada regra, o operador lógico E é modelado pelo operador mínimo. Como exemplo, citam-se as regras a seguir:

Regra 1: Se (x é A_1 e y é B_1) então (z é C_1).

Regra 2: Se (x é A_2 e y é B_2) então (z é C_2).

A Fig. 2.9 ilustra como uma saída real z de um sistema de inferência do tipo Mamdani é gerada a partir das entradas x e y reais e a regra de composição *max-min*. A saída $z \in R$ é obtida pela defuzzificação do conjunto fuzzy de saída $C = C'_1 \cup C'_2$ da Fig. 2.9.

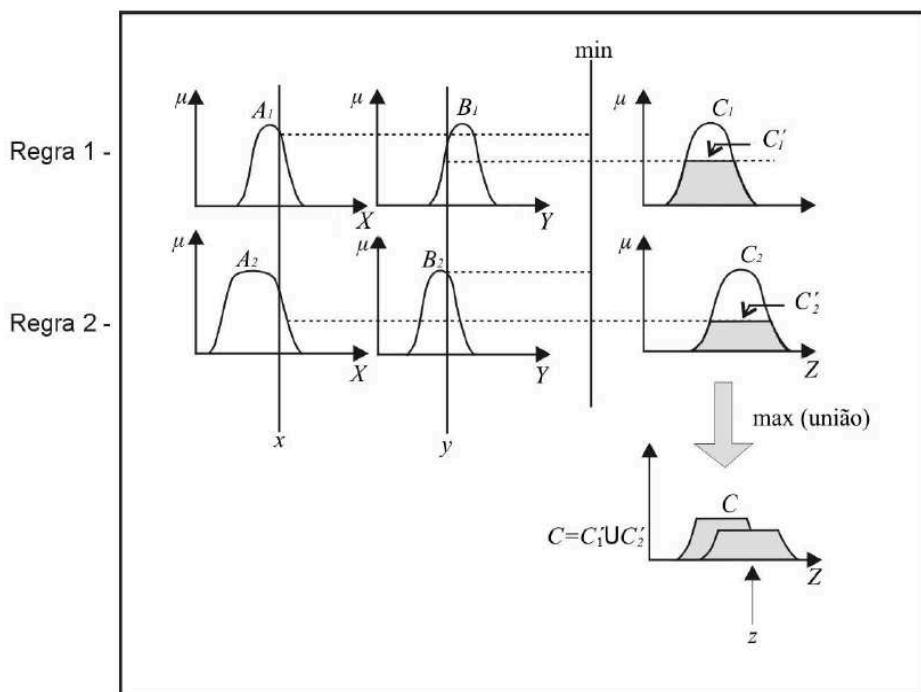


Figura 2.9: Ilustração do método de Mamdani (Jafelice, 2003).

- **Método de Takagi-Sugeno:** A única diferença deste método com relação ao método de Mamdani está na modelagem dos consequentes do SBRF, os quais são funções do tipo $f_i = f(x_1, x_2, \dots, x_n)$, sendo x_1, x_2, \dots, x_n as entradas do SBRF (Valle, 2015);
- **Processador de Saída (Defuzzificação):** Na teoria dos conjuntos fuzzy pode-se dizer que a defuzzificação é um processo de se representar um conjunto fuzzy por um número real. Em sistemas fuzzy, em geral a saída é um conjunto fuzzy. Assim, deve-se escolher um método para defuzzificar a saída e obter um número real que a represente. A seguir, apresentam-se os 2 métodos mais comuns de defuzzificação.
- **Média Ponderada:** É o método de defuzzificação mais usado quando se utiliza o método de inferência fuzzy de Takagi-Sugeno. A saída z para o SBRF é calculada a partir da Eq. (2.10):

$$z = \frac{\sum_{i=1}^k c_i(u) f_i}{\sum_{i=1}^k c_i(u)} \quad (2.10)$$

- **Centro de Gravidade:** Este método de defuzzificação é semelhante à média ponderada para distribuição de dados, com a diferença que os pesos são os valores $C(u)$ que indicam o grau de compatibilidade do valor u com o conceito modelado pelo conjunto fuzzy C , onde R é a região de integração. O centro de gravidade é a forma de defuzzificação mais utilizada quando a inferência fuzzy é dada a partir do método de Mamdani.

$$G(C) = \frac{\int_R u C(u) du}{\int_R C(u) du} \quad (2.11)$$

2.4. Sistema Neuro-Fuzzy

O sistema neuro-fuzzy adaptativo, Fig. 2.10, também conhecido como ANFIS (*Adaptive Neuro-Fuzzy Inference System*) surgiu da fusão da rede neural artificial ANN (*Artificial Neural Network*) com o Sistema de Inferência Fuzzy FIS (*Fuzzy Inference System*). Dessa forma, o ANFIS herda a estrutura do FIS, ou seja, contém o processo de fuzzificação, uma base de regras do tipo *Se ... Então*, uma máquina de inferência fuzzy e a defuzzificação; porém com a mesma representação da estrutura e capacidade de aprendizado e adaptação da ANN através do ajuste dos parâmetros relacionados ao sistema fuzzy.

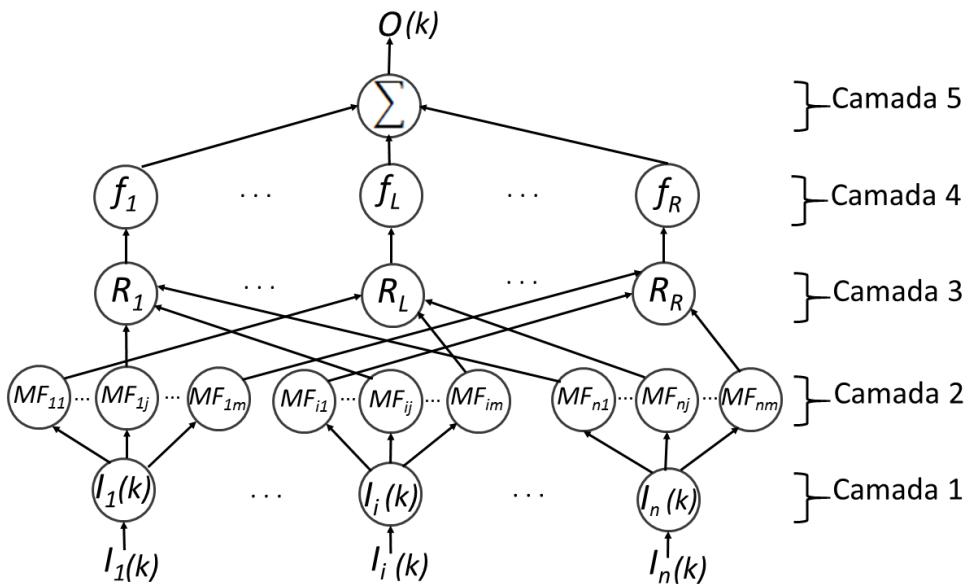


Figura 2.10: Representação de um sistema neuro-fuzzy.

A seguir é indicada a composição e a função de cada uma das camadas do ANFIS, considerando a utilização do método de inferência fuzzy de Takagi-Sugeno (Takagi e Sugeno, 1985; Chen, Lin e Lin, 2008):

- **Camada 1:** Cada nó da primeira camada recebe uma das variáveis de entrada $I_i(k)$ do sistema neuro-fuzzy, de sorte que a saída do nó i da camada 1 no tempo k , $u_i^{(1)}(k)$, é dada por:

$$u_i^{(1)}(k) = I_i(k) \quad (2.12)$$

- **Camada 2:** Realização da fuzzificação dos dados de entrada, ou seja, estes são traduzidos em conjuntos fuzzy em seus respectivos domínios. É neste momento em que são construídas as funções de pertinência MF para a descrição das entradas. Adotando funções de pertinência do tipo gaussiana, a saída do nó ij da camada 2 no tempo k , $u_{ij}^{(2)}(k)$, é dada por:

$$u_{ij}^{(2)}(k) = e^{\frac{(u_i^{(1)}(k) - m_{ij}(k))^2}{\sigma_{ij}^2(k)}} \quad (2.13)$$

em que $m_{ij}(k)$ e $\sigma_{ij}^2(k)$ são respectivamente a média e a variância da função de pertinência gaussiana MF_{ij} .

- **Camada 3:** Coleção de proposições do tipo Se ... Então, formando assim a base de regras do ANFIS. Para toda regra R_L , os operadores lógicos E e OU são modelados respectivamente através dos operadores matemáticos produto e máximo, de modo que a saída do nó L desta camada, $u_L^{(3)}(k)$, seja função das saídas da camada 2 selecionadas a partir da regra R_L .
- **Camada 4:** Cada nó desta camada, denominado nó consequente, realiza o papel de um consequente, que é definido como sendo uma função $f_L: R^n \rightarrow R$ tal que $f_L = f(I_1, \dots, I_i, \dots, I_n, w_{1L}, \dots, w_{jL}, \dots, w_{oL}, k)$, em que $w_{1L}, \dots, w_{jL}, \dots, w_{oL}$ são pesos a serem determinados na fase de treinamento do ANFIS. Dessa forma, a saída do nó L da camada 4 $u_L^{(4)}(k)$ é calculada por:

$$u_L^{(4)}(k) = u_L^{(3)}(k) f_L(I_1, \dots, I_i, \dots, I_n, w_{1L}, \dots, w_{jL}, \dots, w_{oL}, k) \quad (2.14)$$

- **Camada 5:** Determinação da saída do sistema neuro-fuzzy, dada pela Eq. (2.15):

$$O(k) = \frac{\sum_{L=1}^R u_L^{(4)}(k)}{\sum_{L=1}^R u_L^{(3)}(k)} \quad (2.15)$$

Considerando que o ANFIS seja invariante no tempo e que o método de inferência utilizado seja o de Takagi-Sugeno de primeira ordem, os consequentes passam a ser polinômios de primeira ordem, e assim tem-se que:

$$O = \frac{\sum_{L=1}^R u_L^{(4)}}{\sum_{L=1}^R u_L^{(3)}} = \frac{\sum_{L=1}^R u_L^{(3)}(w_{0L} + w_{1L}I_1 + \dots + w_{iL}I_i + \dots + w_{nL}I_n)}{\sum_{L=1}^R u_L^{(3)}} \quad (2.16)$$

Para que sejam obtidos os pesos do sistema neuro-fuzzy, é necessário que este seja treinado a partir de algum método de otimização que minimize a função de custo relacionada ao processo. A subseção 2.5 trata da evolução diferencial, método escolhido de busca dos pesos adequados ao ANFIS.

2.5. Evolução Diferencial

A evolução diferencial, algoritmo desenvolvido por Storn e Price (Storn e Price, 1996), é um método de minimização de funções, podendo ser até não lineares e não diferenciáveis, que se mostra capaz de se chegar ao valor ótimo global até mais facilmente e rapidamente que outros métodos clássicos e heurísticos. O método requer poucas variáveis de controle, é robusto, de fácil implementação e apresenta boa adaptabilidade à computação paralela (Storn e Price, 1996).

O método da evolução diferencial busca encontrar o valor mínimo global de uma função $f_{obj}(X)$, sendo X um vetor cujos elementos representam cada uma das variáveis independentes da função. As etapas e operadores constituintes do algoritmo estão descritos a seguir (Han, Lin e Chang, 2013) e representados na Fig. 2.11:

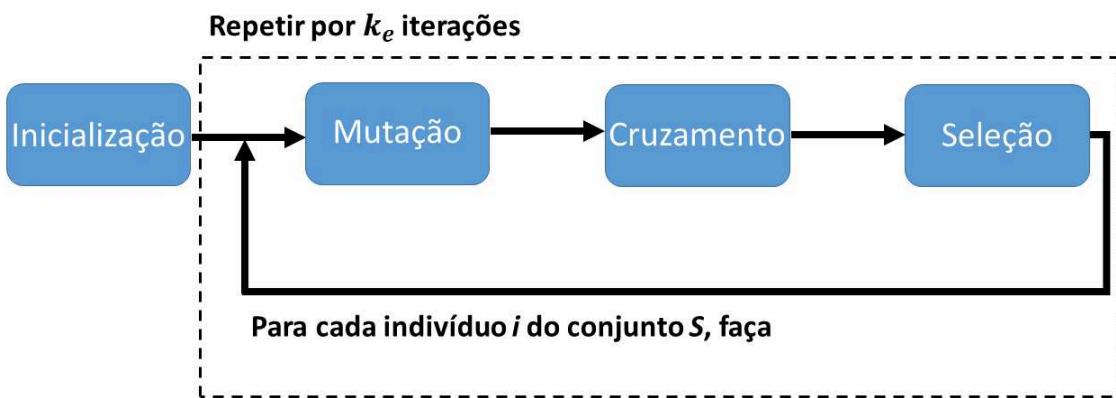


Figura 2.11: Representação da evolução diferencial, adaptada de Mór et al. (2015).

Durante a **inicialização** do algoritmo, os valores dos parâmetros utilizados durante a execução do método são definidos, e são também determinados os limites de busca para cada variável. Os vetores (ou indivíduos) $X_{i,0}$, cada um representando uma possível solução

para o problema de otimização e cujos elementos são os valores normalizados das variáveis do processo, são inicializados seguindo uma distribuição uniforme definida dentro do intervalo $[0,1]$, e em seguida é calculada a aptidão de todos os vetores com relação à função objetivo f_{obj} .

Na fase de **mutação**, utiliza-se a alteração dos vetores $X_{i,k}$ a partir da Eq. (2.17), em que $V_{i,k}$ é o vetor resultante obtido a partir do vetor $X_{i,k}$ na iteração k , F é o fator de perturbação do vetor $X_{i,k}$, geralmente apresentando valor contido no intervalo $[0,2]$, e $X_{j,k}$ e $X_{l,k}$ são outros vetores escolhidos aleatoriamente pertencentes ao conjunto (ou população) S das soluções possíveis para o problema de otimização também na iteração k . Esse procedimento é realizado n vezes, em que n é o número de elementos do conjunto S .

$$V_{i,k} = X_{i,k} + F(X_{j,k} - X_{l,k}) \quad (2.17)$$

Na fase de **cruzamento** (ou recombinação), busca-se mesclar as informações presentes nos vetores $V_{i,k}$ e $X_{i,k}$, garantindo maior diversidade na população. Para cada elemento e do vetor $V_{i,k}$, utiliza-se um gerador de valores uniformemente distribuídos dentro do intervalo $[0,1]$, $rand(0,1)$, e compara-se com o valor da taxa de cruzamento CR , também definido dentro do intervalo $[0,1]$. A tomada de decisão desse operador é especificada na Eq. (2.18), em que $C_{i,k}$ é o vetor candidato a participar do conjunto S na iteração $k+1$ e $randnum(1, n_e)$, sendo n_e o número de elementos dos vetores, é um gerador de números inteiros uniformemente distribuídos dentro do intervalo $[1, n_e]$. Esse procedimento é repetido para todos os vetores do conjunto S .

$$C_{e,i,k} = \begin{cases} V_{e,i,k}, & \text{SE } rand(0,1) \leq CR \text{ OU } e = randnum(1, n_e) \\ X_{e,i,k}, & \text{SE } rand(0,1) > CR \text{ E } e \neq randnum(1, n_e) \end{cases} \quad (2.18)$$

A **seleção** é o último operador padrão da evolução diferencial e que se baseia na análise dos vetores $C_{i,k}$ gerados no cruzamento. A Eq. (2.19) descreve o procedimento de seleção, a ser realizado também para todos os vetores do conjunto S . Após a seleção, avança-se para a iteração $k+1$, e é repetido todo o procedimento a partir do operador mutação até que na iteração k_e alguma condição de parada pré-determinada seja atendida.

$$X_{i,k+1} = \begin{cases} X_{i,k}, & \text{se } f_{obj}(X_{i,k}) \leq f_{obj}(C_{i,k}) \\ C_{i,k}, & \text{outro caso} \end{cases} \quad (2.19)$$

2.6. Regressão

A regressão é uma técnica estatística que se destina à obtenção de um modelo matemático que consiga se ajustar satisfatoriamente a um conjunto de dados, Fig. 2.12 (Montgomery e Runger, 2002).

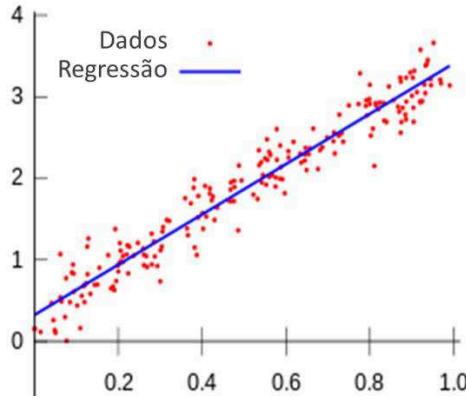


Figura 2.12: Modelo obtido a partir da técnica de regressão, adaptada de Berland (2007).

O processo de regressão dá-se inicialmente na determinação da estrutura do modelo a ser adotado para o sistema em análise. Em seguida, este deve ser escrito de acordo com a Eq. (2.20):

$$Y_{[1 \times 1]} = X_{[1 \times p]} b_{[p \times 1]} = x_1 b_1 + \dots + x_i b_i + \dots + x_p b_p \quad (2.20)$$

onde $Y_{[1 \times 1]}$ é a saída do modelo (variável dependente), $X_{[1 \times p]}$ é um vetor cujos elementos são associados às entradas do sistema (variáveis independentes), e o vetor $b_{[p \times 1]}$ armazena os p coeficientes do modelo matemático.

O exemplo a seguir ilustra a aplicação da escrita acima na equação: $y = c_1 x_1 + c_2 x_1^2 + c_3 x_1 x_2$.

$$Y_{[1 \times 1]} = X_{[1 \times p]} b_{[p \times 1]} \rightarrow y = (x_1 \ x_1^2 \ x_1 x_2) \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

Após a determinação da estrutura matemática e sua escrita da forma apresentada na Eq. (2.20), é necessário utilizar a mesma escrita na representação dos dados de entrada e saída disponíveis:

$$Y_{e[n \times 1]} = X_{[n \times p]} b_{[p \times 1]} + \varepsilon_{[n \times 1]} \quad (2.21)$$

em que n é o número de dados da amostra, que contém informações da entrada X e saída Y_e do sistema analisado, e $\varepsilon_{[nx1]}$ é um vetor cujos elementos representam os ruídos acrescidos devido a fatores não modelados e aos sensores que porventura sejam utilizados na coleta de dados.

A obtenção dos coeficientes do modelo é realizada partindo de 3 hipóteses:

- Os ruídos ε são considerados normalmente distribuídos com média zero e desvio padrão σ ;
- Os elementos da matriz $X_{[mxn]}$ são isentos de ruídos;
- $\begin{cases} COV(\varepsilon_i, \varepsilon_j) = 0, \text{ para } i \neq j \\ COV(\varepsilon_i, \varepsilon_j) = \sigma^2, \text{ para } i = j \end{cases}$, sendo $i, j \in [1, n]$.

e parte-se do princípio da minimização do resíduo SQ_{res} , definido como sendo o quadrado da diferença entre os dados da amostra, $Y_{e[nx1]}$, e seus respectivos valores estimados pelo modelo, $Y_{[nx1]}$:

$$SQ_{res} = (Y_{[nx1]} - Y_{e[nx1]})^T (Y_{[nx1]} - Y_{e[nx1]})$$

$$SQ_{res} = (X_{[nxp]} b_{[px1]} - Y_{e[nx1]})^T (X_{[nxp]} b_{[px1]} - Y_{e[nx1]}) \quad (2.22)$$

O processo de minimização do erro quadrático e a consequente obtenção dos coeficientes do modelo ($b_{[nx1]}$) são feitos a partir da resolução da Eq. (2.23).

$$\frac{dSQ_{res}}{db} = \frac{d(X_{[nxp]} b_{[px1]} - Y_{e[nx1]})^T (X_{[nxp]} b_{[px1]} - Y_{e[nx1]})}{db} = 0 \quad (2.23)$$

O resultado da resolução da equação acima encerra o processo de regressão, e é indicado na Eq. (2.24):

$$b_{[px1]} = (X_{[pxn]}^T X_{[nxp]})^{-1} X_{[pxn]}^T Y_{e[nx1]} \quad (2.24)$$

É necessário, porém, estimar a precisão do ajuste de curva realizado, e isso é feito através da contabilização do intervalo de confiança de cada um dos elementos do vetor $b_{[px1]}$ e da estimativa da taxa de ajuste do modelo aos dados disponíveis, denominado R^2 .

Um estimador para o erro experimental é o resíduo médio MQ_{res} , Eq. (2.25):

$$MQ_{res} = \frac{SQ_{res}}{n-p} \quad (2.25)$$

A partir desse estimador, é possível extrair os intervalos de confiança IC tanto para a saída $Y_{[nx1]}$, Eq. (2.26), quanto para os coeficientes $b_{[px1]}$, Eq. (2.28), do modelo.

$$IC: Y_{[nx1]} - t_{1-\frac{\alpha}{2}} s_e \leq Y_{r[nx1]} \leq Y_{[nx1]} + t_{1-\frac{\alpha}{2}} s_e \quad (2.26)$$

ou

$$IC: X_{[nxp]} b_{[px1]} - t_{1-\frac{\alpha}{2}} s_e \leq Y_{r[nx1]} \leq X_{[nxp]} b_{[px1]} + t_{1-\frac{\alpha}{2}} s_e \quad (2.27)$$

em que $t_{1-\frac{\alpha}{2}}$ é o valor, baseado na distribuição t de Student, que expande a incerteza para o nível de significância α , $Y_{r[nx1]}$ é o valor real da saída do sistema, e $s_e^2 = MQ_{res} diag(I)_{[nx1]}$, sendo I a matriz identidade.

Partindo da Eq. (2.27), é possível mostrar que os intervalos de confiança para coeficientes $b_{[px1]}$ são dados por:

$$IC: b_{[px1]} - t_{1-\frac{\alpha}{2}} s_e \leq \beta_{[px1]} \leq b_{[px1]} + t_{1-\frac{\alpha}{2}} s_e \quad (2.28)$$

onde $\beta_{[px1]}$ é o valor real dos coeficientes do sistema e $s_e^2 = MQ_{res} diag((X_{[pxn]}^T X_{[nxp]})^{-1})_{[px1]}$.

Por fim, a estimativa da taxa de ajuste do modelo aos dados disponíveis exige a definição de 3 variâncias relacionadas aos dados experimentais, determinadas a partir do desenvolvimento matemático a seguir:

- Inicia-se com a igualdade abaixo:

$$Y_{e[nx1]} - \bar{Y}_{e[nx1]} = Y_{e[nx1]} - \bar{Y}_{e[nx1]}$$

- Em seguida, é somado e subtraído o vetor $Y_{[nx1]}$ e rearranja-se o segundo membro da equação da forma indicada a seguir:

$$Y_{e[nx1]} - \bar{Y}_{e[nx1]} = (Y_{e[nx1]} - Y_{[nx1]}) + (Y_{[nx1]} - \bar{Y}_{e[nx1]})$$

- Elevam-se ambos os membros ao quadrado, e após algumas operações matemáticas e considerações, obtém-se a Eq. (2.29):

$$\begin{aligned} (Y_{e[nx1]} - \bar{Y}_{e[nx1]})^T (Y_{e[nx1]} - \bar{Y}_{e[nx1]}) &= (Y_{e[nx1]} - Y_{[nx1]})^T (Y_{e[nx1]} - Y_{[nx1]}) + \\ (Y_{[nx1]} - \bar{Y}_{e[nx1]})^T (Y_{[nx1]} - \bar{Y}_{e[nx1]}) \end{aligned} \quad (2.29)$$

- A partir da Eq. (2.29), definem-se as seguintes variâncias:

$$SQ_{tot} = SQ_{res} + SQ_{reg}$$

sendo:

$$SQ_{tot} = \left(Y_{e[nx1]} - \bar{Y}_{e[nx1]} \right)^T \left(Y_{e[nx1]} - \bar{Y}_{e[nx1]} \right) ,$$

$$SQ_{res} = \left(Y_{e[nx1]} - Y_{[nx1]} \right)^T \left(Y_{e[nx1]} - Y_{[nx1]} \right) , \text{ já utilizada na Eq. (2.22) e}$$

$$SQ_{reg} = \left(Y_{[nx1]} - \bar{Y}_{e[nx1]} \right)^T \left(Y_{[nx1]} - \bar{Y}_{e[nx1]} \right) .$$

Finalmente, através das variâncias, são apresentadas duas formas de se estimar a qualidade de ajuste do modelo aos dados disponíveis: a porcentagem explicável PE , Eq. (2.30), e a taxa de ajuste R^2 , Eq. (2.31).

$$PE = \frac{SQ_{reg}}{SQ_{tot}} \quad (2.30)$$

$$R^2 = \frac{SQ_{reg}}{SQ_{tot}} \left(\frac{n-p}{n} \right) \quad (2.31)$$

Ambos os métodos de avaliação são definidos dentro do intervalo $[0,1]$, sendo que valores próximos de 1 são indicativos de alto grau de qualidade de ajuste do modelo aos dados. Ressalta-se que devido à maior restitutividade do parâmetro R^2 , este habitualmente é preferível em relação à porcentagem explicável.

2.7. Método de Monte Carlo

Para o estudo do método de Monte Carlo, são utilizadas as referências (INMETRO, 2008; Arencibia, 2016).

O método de Monte Carlo é um procedimento numérico para a solução de problemas matemáticos por meio da simulação de variáveis aleatórias. Ele consiste em uma ferramenta que permite combinar duas ou mais distribuições estatísticas, associadas às variáveis de entrada de um modelo matemático definido previamente, e propagá-las à saída deste, gerando novas distribuições.

A Fig. 2.13 ilustra um processo estocástico, em que $VA_{E1} \dots VA_{EN}$ são as variáveis aleatórias do processo, e $VA_{S1} \dots VA_{SN}$ são as variáveis de saída. O método de Monte Carlo pode ser utilizado, por exemplo, quando as distribuições estatísticas das variáveis aleatórias

de entrada são conhecidas, assim como o modelo do processo, e se deseja conhecer as distribuições estatísticas das variáveis aleatórias de saída.

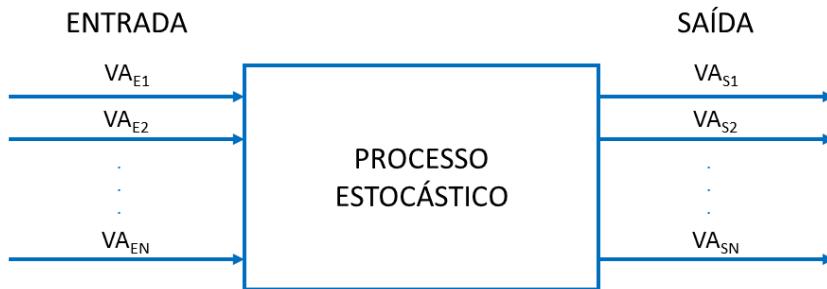


Figura 2.13: Representação de um processo estocástico.

O método de Monte Carlo utiliza a geração de números aleatórios para simular os valores de variáveis aleatórias, técnica simples, facilmente aplicável a sistemas complexos (em oposição ao método clássico de propagação de incerteza), tornando-se viável devido à alta capacidade de processamento dos atuais computadores.

A criação da técnica está ligada aos norte-americanos J. Von Neumann e S. Ulam, e data aproximadamente de 1944, motivada pelo desenvolvimento do computador. Os primeiros trabalhos, porém, que iniciaram sua utilização na avaliação da incerteza de medição foram publicados na década dos anos 90 (Arencibia, 2016).

Ressalta-se que o método de Monte Carlo não busca substituir e nem contradizer os procedimentos descritos pelo GUM (*Guide to the Expression of Uncertainty in Measurement*), guia padrão na contabilização e propagação das incertezas de sistemas de natureza estocástica. O procedimento numérico deve ser utilizado em conjunto com o GUM, ou de forma exclusiva apenas em sistemas de alto nível de complexidade (INMETRO, 2008).

Os passos utilizados na técnica de Monte-Carlo são descritos a seguir:

- **Passo 1:** Definição do mensurando ou variável de saída;
- **Passo 2:** Identificação das variáveis de entrada ou fatores de influência;
- **Passo 3:** Adoção de um modelo matemático que relate as variáveis de entrada à variável de saída;
- **Passo 4:** Identificação das funções densidade de probabilidade (FDP) e dos parâmetros estatísticos, para cada fonte de entrada;
- **Passo 5:** Avaliação da incerteza-padrão das variáveis de entrada;
- **Passo 6:** Definição do número de iterações n_{iter} da técnica de Monte Carlo;

- **Passo 7:** Criação das FDPs a partir da geração de n_{iter} valores para cada variável de entrada, seguindo suas respectivas FDPs;
- **Passo 8:** Propagação das FDPs das variáveis de entrada através do modelo matemático para obtenção da FDP da variável de saída;
- **Passo 9:** Identificação da FDP da variável de saída;
- **Passo 10:** Estimativa da incerteza expandida;
- **Passo 11:** Apresentação do resultado da medição.

2.8. Análise Estatística dos Resultados

A referência utilizada para o estudo das metodologias de análise estatística dos resultados é Montgomery e Runger (2002).

Os testes de hipóteses são utilizados na inferência de certas características de toda uma população, através de uma amostra da mesma contendo um número limitado de dados. A Fig. 2.14 ilustra a utilização das estatísticas média \bar{X} e variância s^2 amostrais na estimativa dos parâmetros média μ e variância σ^2 populacionais.

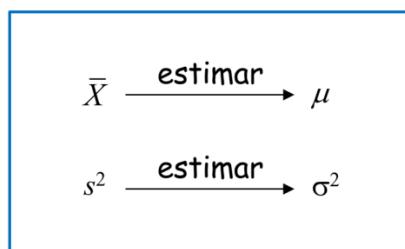


Figura 2.14: Estimação das propriedades de uma população.

O julgamento da população se dá por meio da elaboração de 2 hipóteses, sendo uma delas a crença inicial, denominada hipótese nula H_0 , e sua oposição, denominada hipótese alternativa H_1 . A Fig. 2.15 exemplifica dois testes de hipóteses.

Hipóteses (teste unilateral)	Hipóteses (teste bilateral)
$H_0: \mu = 100$ (Hipótese nula) $H_1: \mu > 100$ (Hipótese Alternativa)	$H_0: \mu = 100$ (Hipótese nula) $H_1: \mu \neq 100$ (Hipótese Alternativa)

Figura 2.15: Testes de hipóteses.

Por meio de testes de hipóteses, é possível realizar inferências sobre médias e variâncias populacionais, além da diferença entre médias e variâncias de duas diferentes populações, Fig. 2.16. A Fig. 2.16 também indica a distribuição estatística utilizada na análise de cada um dos testes de hipóteses, podendo ser as distribuições z (gaussiana normalizada), t de Student, qui-quadrado χ^2 , ou F de Snedecor. Os testes consistem na comparação de dois valores relativos à distribuição estatística adotada: o valor tabelado (v_{tab}) e o valor calculado (v_{calc}). Se $v_{calc} > v_{tab}$, o teste de hipóteses rejeita a hipótese nula com nível de significância α ; caso contrário, o teste não a rejeita com o mesmo nível de significância. Ressalta-se que os valores v_{tab} estão associados ao número de graus de liberdade, que é função do número de dados disponíveis n , e ao nível de significância α adotado.

	Valores Tabelados	Valores Calculados
Determinação de μ :	$\begin{cases} z & \text{se } \sigma^2 \text{ é conhecida} \\ t_{n-1} & \text{se } \sigma^2 \text{ é desconhecida} \end{cases}$	$z = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}} \quad t = \frac{\bar{X} - \mu}{\frac{s}{\sqrt{n}}}$
Determinação de σ^2 :	χ^2_{n-1}	$\chi^2 = \frac{(n-1)s^2}{\sigma^2}$
Determinação de $\mu_1 - \mu_2$:	$\begin{cases} z_{\mu_1 - \mu_2} & \text{se } \sigma_1^2 \text{ e } \sigma_2^2 \text{ são conhecidas} \\ t_{(homo)n_1+n_2-2} & \text{se } \sigma_1^2 \text{ e } \sigma_2^2 \text{ são desconhecidas, mas } \sigma_1^2 = \sigma_2^2 \\ t_{(hetero)n_1+n_2-2} & \text{se } \sigma_1^2 \text{ e } \sigma_2^2 \text{ são desconhecidas, mas } \sigma_1^2 \neq \sigma_2^2 \end{cases}$	$z_{\mu_1 - \mu_2} = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$ $t_{(homo)} = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2} \cdot \frac{1}{n_1} + \frac{1}{n_2}}}$ $t_{(hetero)} = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$
Determinação de $\frac{\sigma_1^2}{\sigma_2^2}$:	F_{n_1-1, n_2-1}	$F = \frac{s_1^2 \sigma_2^2}{s_2^2 \sigma_1^2}$

Figura 2.16: Esquema simplificado dos testes de hipóteses.

Para facilitar o entendimento do assunto, a seguir é apresentado um exemplo de aplicação de um teste de hipóteses (Ortiz, 2010):

- O esforço necessário para a ruptura dos cabos produzidos por um fabricante apresenta média (μ) de 1800 kgf e desvio padrão (σ) de 100 kgf. Mediante nova técnica de produção, proclamou-se que o esforço demandado pode ter aumentado. Para testar essa declaração, selecionou-se uma amostra de 50 cabos, chegando-se a uma média amostral de 1830 kgf. Pode-se confirmar a declaração ao nível de significância de 1%?
 - $H_0: \mu = 1800 \text{ kgf};$
 - $H_1: \mu > 1800 \text{ kgf};$

- Teste de hipóteses unilateral, $\alpha = 0,01$, σ conhecido \rightarrow distribuição estatística z;
- $z_{calc} = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}} = \frac{1830 - 1800}{\frac{100}{\sqrt{50}}} = 2,12$;
- Por meio de uma tabela da distribuição z, é possível verificar que para este caso, considerando a distribuição unilateral e $\alpha = 0,01$, tem-se $z_{tab} = 2,33$;
- $z_{calc} < z_{tab}$, portanto não se rejeita H_0 , ou seja, a nova técnica não melhora significativamente a 1% o esforço necessário para a ruptura dos cabos.

2.9. Modelagem de Veículo de Duas Rodas do Tipo Pêndulo Invertido

Como referência para o estudo da modelagem matemática de veículos de duas rodas do tipo pêndulo invertido, utiliza-se (Bonafilia et al., 2014).

De forma análoga ao princípio de funcionamento de um pêndulo, o pêndulo invertido, Fig. 2.17, apresenta oscilação da estrutura em torno de sua posição de equilíbrio, porém, ao contrário do pêndulo tradicional, o sistema em questão é marginalmente estável, pois qualquer perturbação submetida à estrutura é capaz de repelir a planta com relação à sua posição de equilíbrio. O principal objetivo dos controladores aplicados a plantas que seguem o princípio do pêndulo invertido está em minimizar o ângulo de inclinação θ bem como a translação x do veículo, garantindo assim sua estabilidade estática.

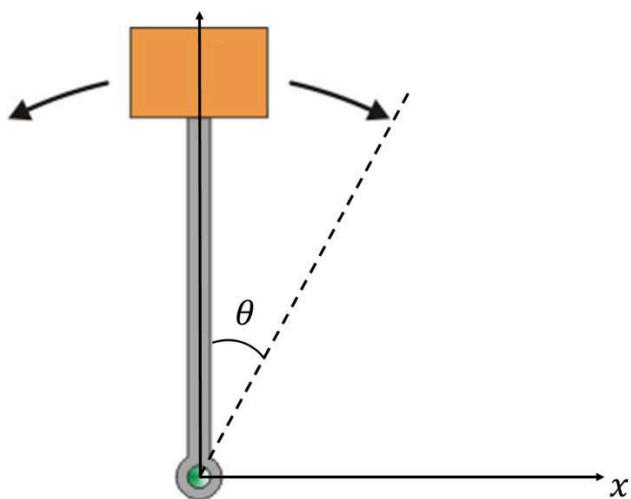


Figura 2.17: Representação de uma planta que segue o princípio do pêndulo invertido.

A Fig. 2.18a é a representação do veículo de duas rodas do tipo pêndulo invertido desenvolvido por Bonafilia et al. (2014), indicando a posição de componentes, como o

microprocessador, acelerômetro, giroscópio, rodas, motores e encoders; e a Fig. 2.18b refere-se ao circuito elétrico de seus motores, sendo R , L e u_e respectivamente a resistência, a indutância e a força contraeletromotriz associadas a cada um dos motores.

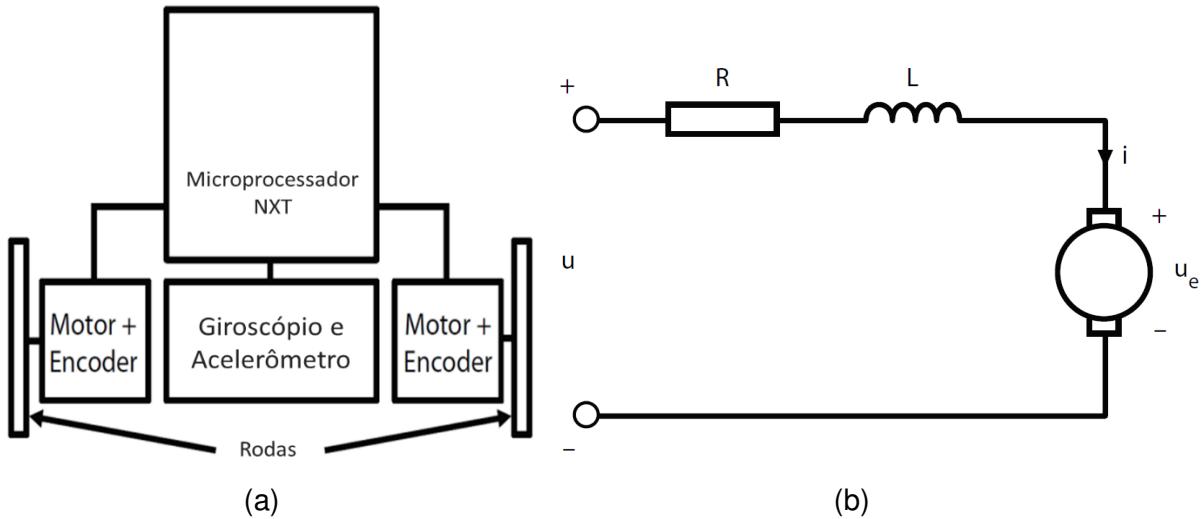


Figura 2.18: (a) Veículo de duas rodas e (b) o circuito elétrico de seus motores (Bonafilia et al., 2014).

As Figs. 2.19a e 2.19b referem-se respectivamente aos diagramas de corpo livre (DCLs) da estrutura do veículo e de suas rodas. A Tab. 2.1 e os enunciados das equações que se seguem descrevem os parâmetros e variáveis apresentadas nas figuras.

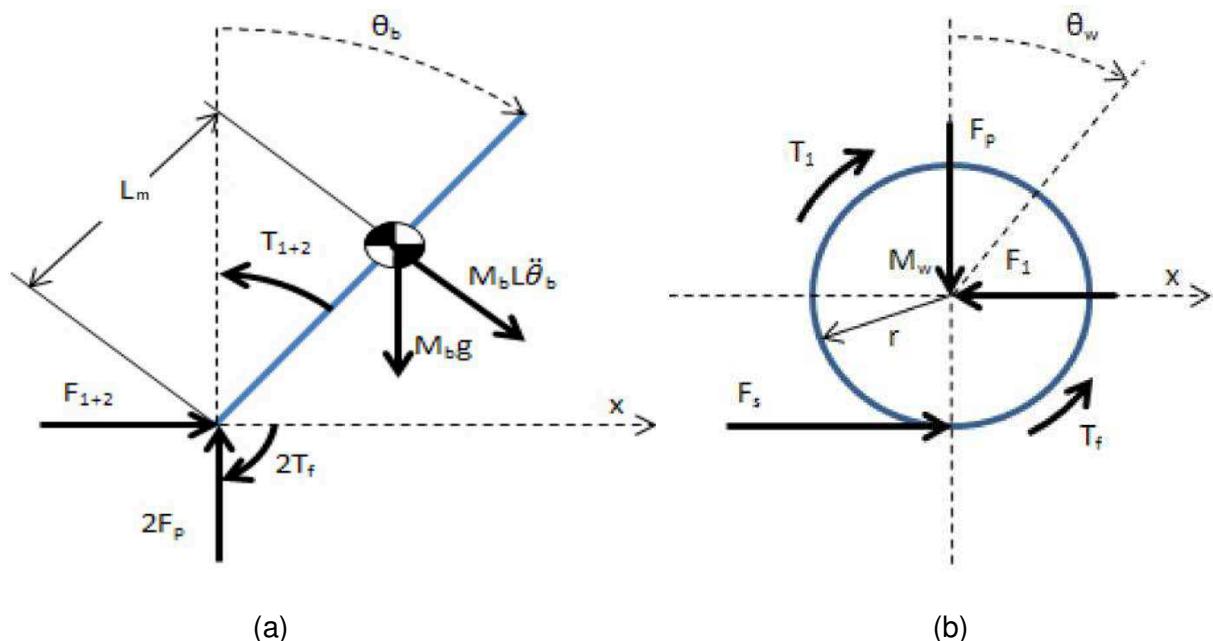


Figura 2.19: Diagramas de corpo livre (a) da estrutura do veículo e das rodas (b) (Bonafilia et al., 2014).

Os parâmetros envolvidos no sistema dinâmico são apresentados na Tab. 2.1, e a modelagem dinâmica do veículo de duas rodas é indicada a seguir:

Tabela 2.1: Parâmetros do processo.

Parâmetro	Descrição
M_b	Massa da estrutura do veículo
M_w	Massa das rodas do veículo
J_b	Momento de inércia da estrutura do veículo em relação ao seu centro de massa
J_w	Momento de Inércia da roda em relação ao seu centro de massa
r	Raio das rodas
L_m	Distância do eixo das rodas ao centro de massa do veículo
K_e	Razão entre a força contraeletromotriz gerada pelo motor e sua velocidade
K_m	Razão entre o torque gerado pelo motor e sua corrente de armadura
R	Resistência elétrica dos motores
b	Constante relacionada ao atrito viscoso dos motores
g	Aceleração da gravidade

A modelagem do sistema dinâmico é subdividida em 6 partes, sendo elas o subsistema elétrico, as rodas, a estrutura do veículo, a ligação entre a estrutura e a roda, a linearização do sistema resultante e a discretização do modelo.

O subsistema elétrico tem sua modelagem iniciada a partir da Eq. (2.32), que descreve a força contraeletromotriz (u_e) em função da velocidade angular da roda ($\dot{\theta}_w$) e da estrutura do veículo ($\dot{\theta}_b$):

$$u_e = K_e(\dot{\theta}_w - \dot{\theta}_b). \quad (2.32)$$

O torque (T_1) gerado pela passagem de corrente (i) na armadura do motor 1 é modelado através da Eq. (2.33):

$$T_1 = K_m i. \quad (2.33)$$

Relaciona-se a tensão (u) imposta aos terminais do motor à corrente (i) que passa pelo circuito através da Eq. (2.34):

$$u = Ri + L \frac{di}{dt} + u_e. \quad (2.34)$$

A partir das Eqs. (2.32, 2.33 e 2.34) e através da manipulação matemática indicada pela Eq. (2.35), encerra-se a modelagem do subsistema elétrico após a determinação de uma nova expressão para o torque T_1 , Eq. (2.36):

$$i = \frac{u-u_e}{R} = \frac{u}{R} - \frac{K_e}{R} (\dot{\theta}_w - \dot{\theta}_b) \quad (2.35)$$

$$T_1 = \frac{K_m}{R} u - \frac{K_e K_m}{R} (\dot{\theta}_w - \dot{\theta}_b) \quad (2.36)$$

As equações do movimento das rodas são determinadas utilizando-se da Segunda Lei de Newton-Euler e estão apresentadas a seguir. J_w é o momento de inércia da roda/eixo/motor, \ddot{x}_w é a aceleração tangencial das rodas, F_s é a força de atrito estático superfície/roda, T_f é o torque de fricção, em função da velocidade angular da roda, e F_1 é a força na direção horizontal exercida pela estrutura à roda 1:

$$J_w \ddot{\theta}_w = T_1 - rF_s - T_f \quad (2.37)$$

$$M_w \ddot{x}_w = F_s - F_1 \quad (2.38)$$

$$\ddot{x}_w = \ddot{\theta}_w r \quad (2.39)$$

As equações para a estrutura do veículo são obtidas a partir da decomposição das forças envolvidas na movimentação do corpo do pêndulo, dessa forma, a Eq. (2.40) é a representação matemática do movimento da estrutura do veículo na direção horizontal, sendo \ddot{x}_b a aceleração horizontal da estrutura do veículo; a Eq. (2.41) descreve o movimento em torno do centro de massa, sendo F_p a força de reação na direção vertical exercida pelas rodas; e a Eq. (2.42) modela o movimento perpendicular ao pêndulo.

$$M_b \ddot{x}_b = (F_1 + F_2) - M_b L_m \ddot{\theta}_b \cos \theta_b \quad (2.40)$$

$$J_b \ddot{\theta}_b = -(F_1 + F_2) L_m \cos \theta_b - (T_1 + T_2) + 2F_p L_m \sin \theta_b + 2T_f \quad (2.41)$$

$$M_b \ddot{x}_b \cos \theta_b = (F_1 + F_2) \cos \theta_b - M_b L_m \ddot{\theta}_b - 2F_p \sin \theta_b + M_b g \sin \theta_b \quad (2.42)$$

A ligação matemática entre a estrutura do veículo e as rodas é dada a partir da retirada da dependência das variáveis F_1 e F_2 , Eq. (2.43); e substituição da Eq. (2.41) na Eq. (2.42), gerando a Eq. (2.44), em que T_1 e T_2 são obtidos através da Eq. (2.36) e $T_f = b(\dot{\theta}_w - \frac{\dot{x}_w}{r})$:

$$M_b \ddot{x}_b + M_b L_m \ddot{\theta}_b \cos \theta_b = -2 \left(M_w + \frac{J_w}{r^2} \right) \ddot{x}_w - \frac{2K_e K_m}{r^2 R} \dot{x}_w + \frac{2K_e K_m}{r R} \dot{\theta}_b + \frac{2K_m}{r R} u - \frac{2T_f}{r} \quad (2.43)$$

$$(J_b + M_b L_m^2) \ddot{\theta}_b = M_b g L_m \sin \theta_b - M_b \ddot{x}_b L_m \cos \theta_b + 2T_f - (T_1 + T_2) \quad (2.44)$$

Ao se impor $\ddot{x}_w = \ddot{x}_b$ e sendo $\theta = \theta_b$ e $x = x_w$, onde x é a posição do veículo, \dot{x} é sua velocidade de translação, θ é o ângulo de elevação, $\dot{\theta}$ é a variação do ângulo de elevação no tempo, e u é a tensão de alimentação dos dois motores DC, obtém-se o modelo linearizado do veículo de duas rodas do tipo pêndulo invertido:

$$\dot{\xi} = A\xi + Bu \quad (2.45)$$

em que:

$$\xi = [x \ \dot{x} \ \theta \ \dot{\theta}]^T$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \alpha_M & \beta_M & -r\alpha_M \\ 0 & 0 & 0 & 1 \\ 0 & \gamma_M & \delta_M & -r\gamma_M \end{bmatrix}$$

$$B = [0 \ \alpha_M \varepsilon_M \ 0 \ \gamma_M \varepsilon_M]^T$$

sendo:

$$\alpha_M = \frac{2(Rb - K_e K_m)(M_b L_m^2 + M_b r L_m + J_b)}{R(2(J_b J_w + J_w L_m^2 M_b + J_b M_w r^2 + L_m^2 M_b M_w r^2) + J_b M_b r^2)}$$

$$\beta_M = \frac{-L_m^2 M_b^2 g r^2}{J_b(2J_w + M_b r^2 + 2M_w r^2) + 2J_w L_m^2 M_b + 2L_m^2 M_b M_w r^2}$$

$$\gamma_M = \frac{-2(Rb - K_e K_m)(2J_w + M_b r^2 + 2M_w r^2 + L_m M_b r)}{Rr(2(J_b J_w + J_w L_m^2 M_b + J_b M_w r^2 + L_m^2 M_b M_w r^2) + J_b M_b r^2)}$$

$$\delta_M = \frac{L_m M_b g (2J_w + M_b r^2 + 2M_w r^2)}{2J_b J_w + 2J_w L_m^2 M_b + J_b M_b r^2 + 2J_b M_w r^2 + 2L_m^2 M_b M_w r^2}$$

$$\varepsilon_M = \frac{K_m r}{Rb - K_e K_m}$$

Para que o modelo dinâmico possa ser utilizado em simulações do funcionamento do mecanismo, a Eq. (2.45) é discretizada considerando um segurador de ordem zero, da forma descrita a seguir:

$$\xi_{k+1} = G\xi_k + Hu_k. \quad (2.46)$$

As matrizes G e H são determinadas através das Eqs. (2.4) e (2.5):

$$G = e^{AT_s}$$

$$H = A^{-1}(e^{AT_s} - I)B$$

em que T_s é o tempo de amostragem e I é a matriz identidade.

2.10. Zona Morta de Atuadores

A zona morta de um atuador, Fig. 2.20, é caracterizada por ser o(s) intervalo(s) em que a variação da entrada não exerce influência na resposta do sistema. Peng e Dubay (2011) modelam um atuador dotado de zona morta a partir da Eq. (2.47):

$$y = \begin{cases} f_n(u), & u \leq \delta_n \\ 0, & \delta_n < u < \delta_p \\ f_p(u), & u \geq \delta_p \end{cases} \quad (2.47)$$

em que u é a ação de controle aplicada ao sistema, y é a saída, $f_n(u)$ e $f_p(u)$ são funções que relacionam a ação de controle à saída do sistema e os parâmetros δ_n e δ_p determinam a amplitude da zona morta.

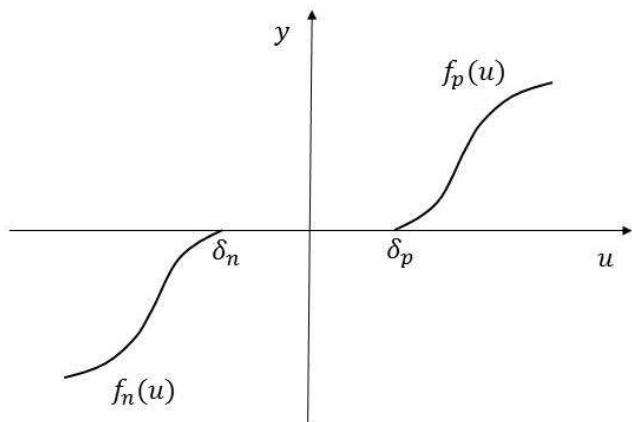


Figura 2.20: Representação da zona morta.

Os parâmetros δ_n e δ_p são de difícil obtenção, pois se alteram em função das cargas a que os atuadores estão submetidos, e também são função do desgaste mecânico dos mesmos.

Os motores DC de um veículo de duas rodas estão submetidos ao efeito da zona morta, e no intuito de se adicionar o efeito dessa não-linearidade na modelagem dinâmica apresentada na Eq. (2.46), realiza-se uma simplificação do modelo da Eq. (2.47). Dessa forma, a ação de controle percebida pela dinâmica do sistema e utilizada como sinal de entrada na Eq. (2.46), u_k , é modelada a partir da Eq. (2.48):

$$u_k = \begin{cases} u_k^0, & u_k^0 \leq -\delta \\ 0, & -\delta < u_k^0 < +\delta \\ u_k^0, & u_k^0 \geq +\delta \end{cases} \quad (2.48)$$

em que u_k^0 é a ação de controle de fato aplicada nos motores do veículo e δ é a amplitude da zona morta do mecanismo.

CAPÍTULO III

VEÍCULO DE DUAS RODAS DO TIPO PÊNDULO INVERTIDO

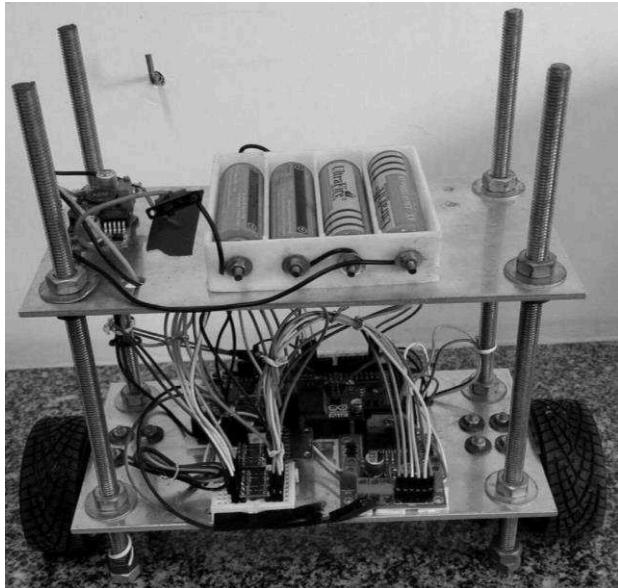
Este capítulo é dedicado à apresentação da bancada experimental utilizada no trabalho, aparato empregado nas aplicações das técnicas de modelagem e controle neuro-fuzzy. No intuito de se permitir a comparação dessas técnicas com os métodos clássicos de modelagem (equações do movimento - Eqs. (2.45) e (2.48)) e controle de sistemas, apresenta-se também neste capítulo a identificação do modelo dinâmico do veículo de duas rodas do tipo pêndulo invertido.

3.1. Descrição da Bancada Experimental

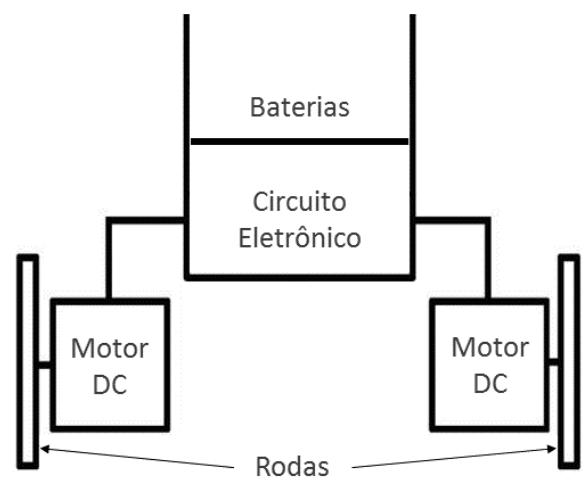
O veículo de duas rodas do tipo pêndulo invertido, cuja descrição é dada a seguir, é apresentado na Fig. 3.1, assim como um esquema mostrando a distribuição dos seus componentes:

- **Estrutura:** Composta por placas de alumínio, fusos rosados, porcas e arruelas M8;
- **Motor:** Corrente contínua, 6 V de tensão nominal;
- **Rodas:** Apresentam pneus de borracha, garantindo a aderência do veículo ao solo;
- **Módulo de baterias:** 4 baterias em série de 5800 mAh e 3,7 V, associadas a um módulo *buck-boost*, responsável por manter a tensão máxima de alimentação dos motores por volta de 6V;

- **Círculo Eletrônico:** Composto pela plataforma de prototipagem arduino DUE, o qual é dotado de um microcontrolador Atmel SAM3X8E ARM Cortex-M3 de 32 bits e 84 MHz de clock; pelo driver L298N para a amplificação da corrente que alimenta os motores; e uma IMU (*Inertial Measurement Unit*) MPU6050, que é capaz de mensurar o ângulo de inclinação do sistema.



(a)



(b)

Figura 3.1: Veículo de duas rodas – (a) foto e (b) representação dos seus componentes.

3.2. Modelagem da Bancada Experimental

A modelagem do veículo é realizada a partir da Eq. (2.45):

$$\dot{\xi} = A\xi + Bu$$

que discretizada, considerando que o circuito eletrônico envia aos motores um valor constante de tensão dentro de um intervalo de tempo T_s (segurador de ordem zero), tem-se – Eq. (2.46):

$$\xi_{k+1} = G\xi_k + Hu_k$$

O motor apresenta zona morta com amplitude δ , dessa forma, a ação de controle percebida pelo sistema dinâmico, u_k , é determinada a partir da Eq. (2.48), sendo u_k^0 a tensão de fato aplicada aos terminais dos motores do veículo.

$$u_k = \begin{cases} u_k^0, & u_k^0 \leq -\delta \\ 0, & -\delta < u_k^0 < +\delta \\ u_k^0, & u_k^0 \geq +\delta \end{cases}$$

Assim, após determinar as equações que regem o modelo dinâmico da bancada, é necessário obter os parâmetros relacionados ao mesmo, os quais se resumem aos especificados na Tab. 2.1, além da amplitude da zona morta δ .

A obtenção dos parâmetros é dada em duas etapas, sendo que a primeira consiste em estimar previamente os parâmetros mecânicos do veículo, e na segunda é realizado um refinamento dos mesmos e a estimativa dos parâmetros elétricos, ambos a partir da evolução diferencial.

Na primeira etapa, os parâmetros M_b e M_w são estimados por meio de uma balança, r é obtido a partir de uma régua, L_m - relacionado ao centro de massa da estrutura - é calculado pela equação $L_m = \frac{\sum CM_i M_i}{M_b}$ - em que M_i e CM_i referem-se, respectivamente, à massa e posição do centro de massa de cada um dos elementos que compõem o veículo com exceção das rodas -, J_w é estimado pela equação $J_w = \frac{1}{2} M_w (r_{int}^2 + r^2)$ - sendo r_{int} o raio interno das rodas -, e J_b é calculado de forma análoga a L_m , sendo estimado através da equação $J_b = \sum M_i (CM_i - L_m)^2$.

Na segunda etapa, utiliza-se o algoritmo de evolução diferencial com 80 indivíduos, número máximo de iterações igual a 100, fator de perturbação igual a 0,4 e taxa de cruzamento unitária. A função objetivo adotada como critério de minimização é o erro médio quadrático MSE , Eq. (3.1):

$$MSE = \frac{\sum_{k=1}^{k_{\max}} (\hat{\theta}_k - \theta_{\exp_k})^2}{k_{\max}} \quad (3.1)$$

em que $\hat{\theta}_k$ é o ângulo de inclinação estimado pelo modelo e θ_{\exp_k} é o valor do ângulo obtido experimentalmente, sendo k_{\max} o número total de observações.

No intuito de se estabilizar o veículo de duas rodas, maximizar a sua amplitude de oscilação e consequentemente permitir uma melhor estimativa do modelo da planta, adota-se um controlador proporcional com ganho crítico, o qual é definido como sendo o ganho necessário para levar o sistema à iminência da instabilidade. O valor do ganho crítico é obtido empiricamente através do aumento gradativo do valor do ganho até que o sistema se torne marginalmente estável.

A Fig. 3.2a ilustra os dados utilizados para a obtenção dos parâmetros (dados de trabalho) e na Fig. 3.2b são apresentados os dados de validação do modelo, sendo que em ambas as figuras a curva obtida através do modelo está representada em azul, e em vermelho a curva relativa aos dados experimentais.

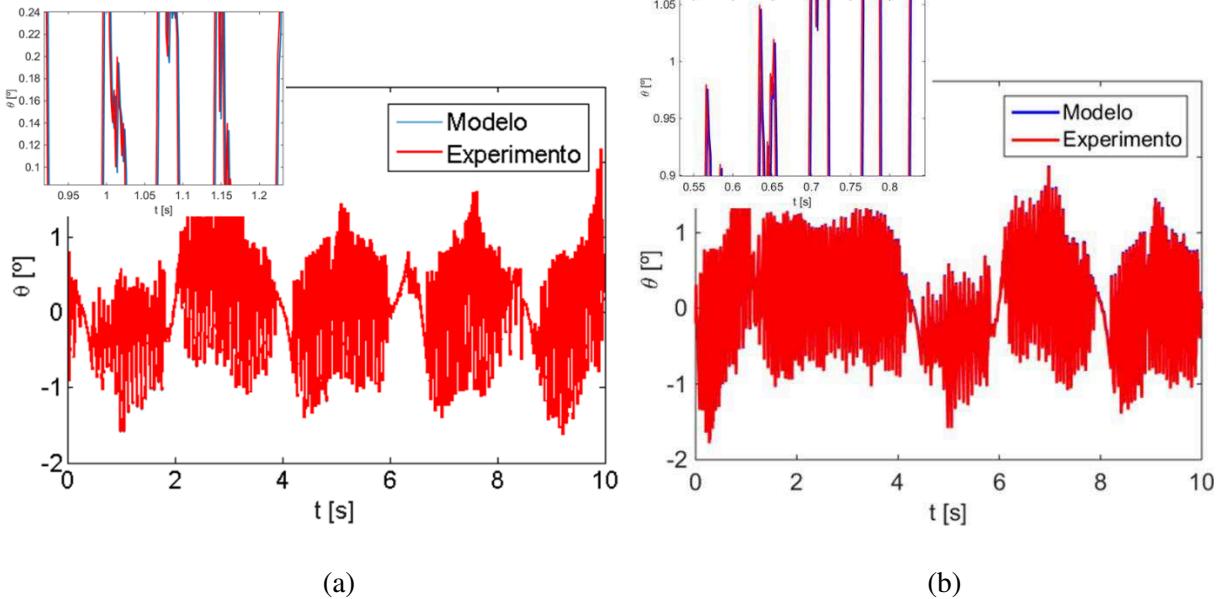


Figura 3.2: (a) Dados de obtenção e (b) validação do modelo.

A Fig. 3.3 apresenta a evolução de MSE com o passar das iterações e a Tab. 3.1 mostra o valor dos parâmetros que foram obtidos no processo de identificação da planta. Ressalta-se que o processo de otimização obteve taxa de ajuste do modelo aos dados experimentais, Eq. (2.30), de 93%.

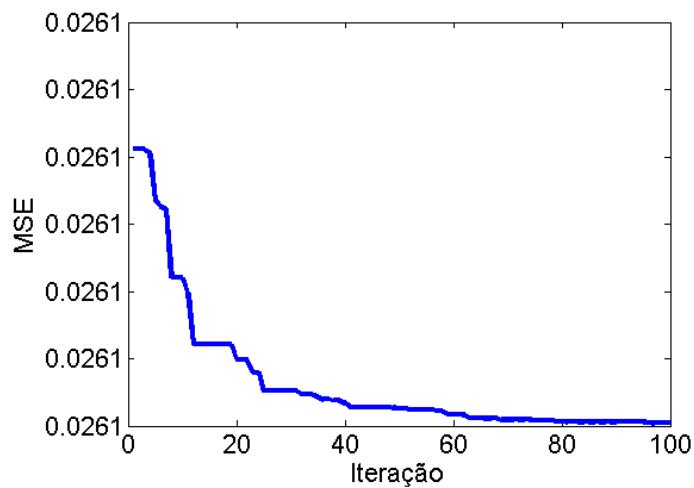


Figura 3.3: Evolução do erro médio quadrático no tempo.

Tabela 3.1: Valor dos Parâmetros do Processo.

Parâmetro	Valor
M_b	1,304 kg
M_w	0,031 kg
J_b	0,00480 kgm^2
J_w	0,00003 kgm^2
r	0,032 m
L_m	0,0748 m
K_e	0,3400 Vs/rad
K_m	0,3373 Nm/A
R	0,93 Ω
b	0,0001 Nms/rad
δ	0,12 V

Os resultados apresentados através das Figs. 3.2 e 3.3, além do valor obtido para a taxa de ajuste do modelo dinâmico aos dados experimentais, indicam a eficácia do método da evolução diferencial em minimizar o erro médio quadrático MSE associado ao processo de obtenção do modelo. Os resultados também sugerem que o modelo dinâmico linearizado, analisado próximo ao ponto de equilíbrio $\theta = 0^\circ$, é um bom estimador do comportamento da dinâmica da planta.

O projeto de um controlador, porém, que tenha em vista o modelo dinâmico de um veículo de duas rodas e as não-linearidades dos atuadores, deve levar em consideração uma série de parâmetros muitas vezes difíceis de serem calculados ou obtidos. Além disso, alguns destes parâmetros estão submetidos a variações durante a operação do veículo devido ao desgaste dos seus componentes, ao acréscimo de massa, à variação da posição do centro de gravidade, à redução gradativa da tensão da bateria, dentre outros fatores. Sendo assim, a metodologia proposta de modelagem e controle neuro-fuzzy, dada à sua versatilidade e capacidade de adaptação durante a operação dos sistemas, passa a ser uma opção de interesse por parte do projetista.

CAPÍTULO IV

MODELAGEM NEURO-FUZZY

Neste capítulo, no intuito de se levar em consideração todas as não-linearidades presentes na planta, propõe-se inicialmente a modelagem neuro-fuzzy a partir de dados experimentais do sistema, fazendo o uso da evolução diferencial como método de otimização.

4.1. Obtenção do Modelo Neuro-Fuzzy

O modelo neuro-fuzzy para o veículo de duas rodas pode ser implementado tanto off-line, a partir do processamento dos dados experimentais e obtenção do modelo anteriormente à operação do sistema, quanto on-line, em que periodicamente o modelo é alterado a partir dos dados experimentais obtidos em tempo real durante a operação do mesmo. Opta-se pela abordagem off-line devido à restrição da capacidade de processamento do microcontrolador utilizado no veículo.

Anteriormente à identificação do modelo, definem-se as entradas e as saídas do sistema neuro-fuzzy. A Fig. 4.1 ilustra o funcionamento do ANFIS, cujas variáveis de entrada são o ângulo de inclinação θ_k , obtido por meio da *IMU* presente no veículo, a variação do ângulo de inclinação $\Delta\theta_k = \theta_k - \theta_{k-1}$, e a ação de controle u_{k+1} imposta ao mecanismo no

tempo $k + 1$; e a variável de saída é o ângulo de inclinação estimado para o tempo $k + 1$, $\hat{\theta}_{k+1}$.



Figura 4.1: Estrutura do modelo neuro-fuzzy.

Para que seja possível o treinamento do modelo neuro-fuzzy, é necessário que se tenha um conjunto de dados de entrada e saída do veículo de duas rodas da forma:

- Entrada: u_{k+1} , $\Delta\theta_k$ e θ_k .
- Saída: θ_{k+1} .

Porém, devido à instabilidade natural do sistema, é necessário que este robô seja previamente controlado. Utiliza-se então o mesmo controlador proporcional aplicado na obtenção do modelo dinâmico apresentado no capítulo 3.

Duas amostras, contendo cada uma 10 s de coleta de dados com tempo de amostragem T_s de 0,005 s (no apêndice B está apresentado o estudo para a escolha do tempo de amostragem adequado para a planta), são extraídas do veículo, sendo que uma destas é utilizada para a obtenção do modelo, e com a outra realiza-se a validação do mesmo. Ressalta-se que em 10 s, utilizando o tempo de amostragem indicado, coletam-se 2000 dados, quantidade suficiente para se extrair informações sobre a planta.

As Figs. 4.2a e 4.2b apresentam, respectivamente, a tensão dos motores u , em volts, e o ângulo de inclinação θ , em graus, extraídos da amostra de obtenção do modelo.

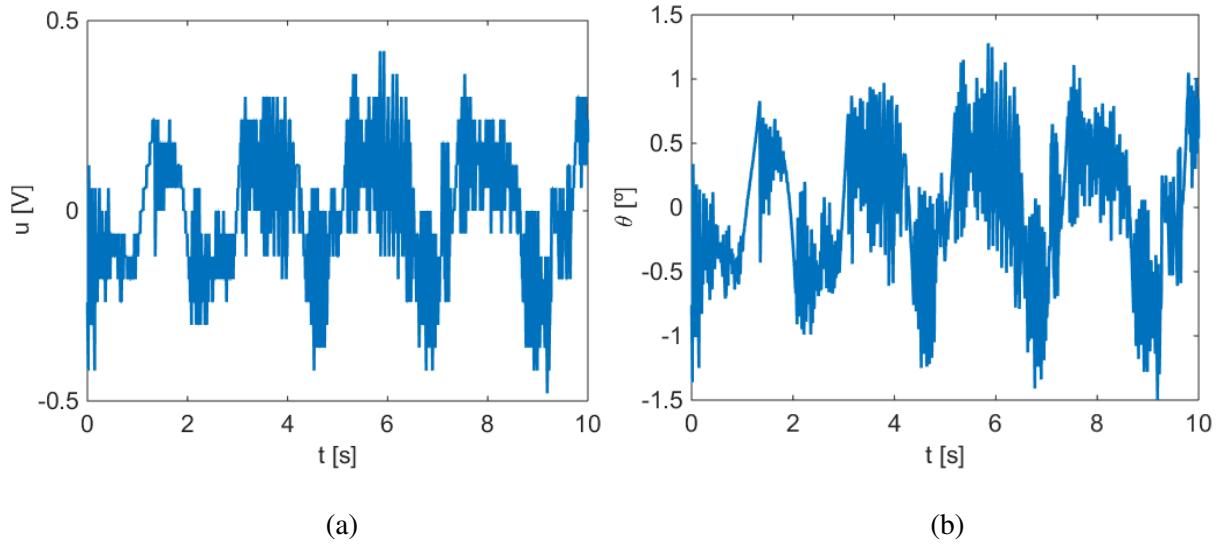


Figura 4.2: (a) Ação de controle e (b) ângulo de inclinação utilizados como dados de obtenção do modelo.

Já as Figs. 4.3a e 4.3b apresentam, respectivamente, a tensão dos motores e o ângulo de inclinação extraídos da amostra de validação.

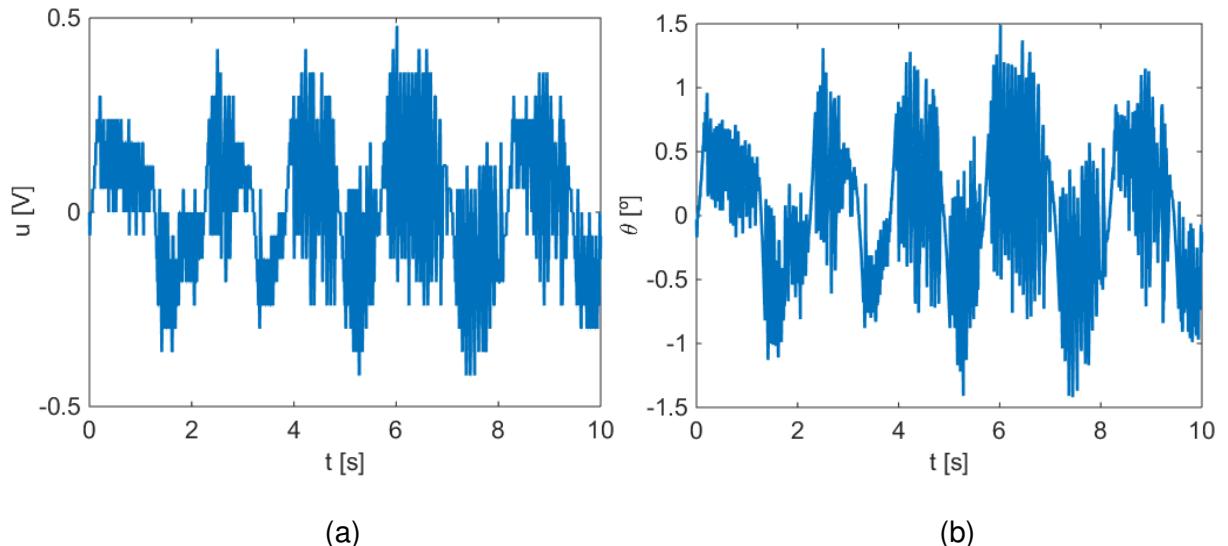


Figura 4.3: (a) Ação de controle e (b) ângulo de inclinação utilizados como dados de validação do modelo.

A partir dos dados acima, e tendo como base o conhecimento qualitativo da dinâmica do veículo, constrói-se o sistema neuro-fuzzy indicado a seguir:

- **Variáveis de entrada:** Entradas normalizadas e no tempo k , $u_{N,k+1}$, $\theta_{N,k}$ e $\Delta\theta_{N,k}$. Os parâmetros de normalização utilizados são respectivamente 6 (tensão máxima aplicada aos motores), 1,5 e 0,5 (valores próximos aos máximos observados nas amostras).
- **Funções de pertinência das variáveis de entrada:** As Figuras 4.4a, 4.4b e 4.4c indicam respectivamente as funções de pertinência relativas às entradas normalizadas $u_{N,k+1}$, $\theta_{N,k}$ e $\Delta\theta_{N,k}$.

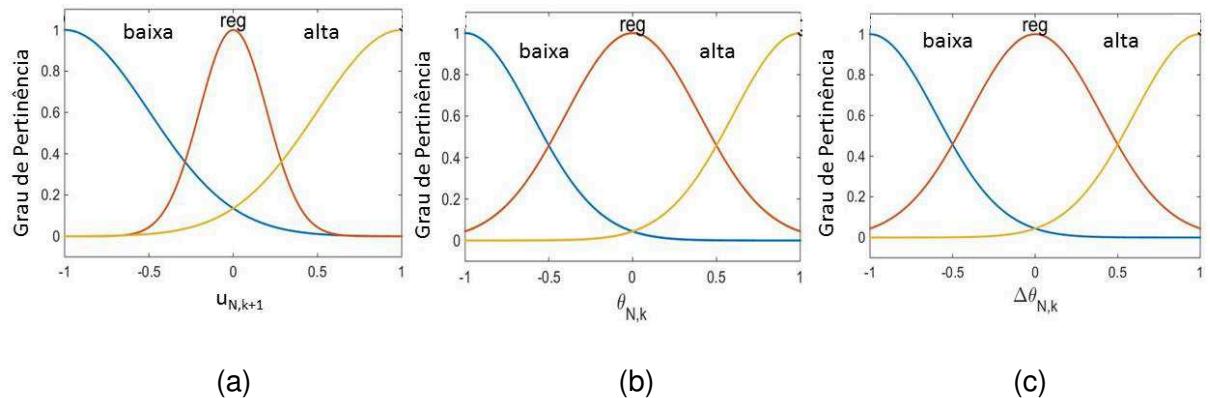


Figura 4.4: Funções de pertinência das variáveis de entrada (a) $u_{N,k+1}$, (b) $\theta_{N,k}$ e (c)

$$\Delta\theta_{N,k}.$$

- **Variável de saída:** Ângulo de inclinação estimado, normalizado e no tempo $k + 1$, $\hat{\theta}_{N,k+1}$.
- **Consequentes do sistema neuro-fuzzy:** Baseados no método de Takagi-Sugeno de primeira ordem (escolhido por aliar uma boa capacidade de resolução de problemas de otimização e controle, e um baixo custo computacional se comparado aos demais métodos de inferência fuzzy), e estão indicados em sequência:
 - $\hat{\theta}_{N,k+1}$ baixa: $f_1 = w_{01} + w_{11}u_{N,k+1} + w_{21}\theta_{N,k} + w_{31}\Delta\theta_{N,k}$;
 - $\hat{\theta}_{N,k+1}$ regular: $f_2 = w_{02} + w_{12}u_{N,k+1} + w_{22}\theta_{N,k} + w_{32}\Delta\theta_{N,k}$;
 - $\hat{\theta}_{N,k+1}$ alta: $f_3 = w_{03} + w_{13}u_{N,k+1} + w_{23}\theta_{N,k} + w_{33}\Delta\theta_{N,k}$.
- **Base de regras:** A base de regras do sistema neuro-fuzzy, construída a partir do conhecimento qualitativo do mecanismo, é apresentada na Tabela 4.1, em que N/A indica que a variável de entrada indicada não se aplica à regra em questão:

Tabela 4.1: Base de Regras do Modelo Neuro-Fuzzy.

SE			ENTÃO
$\theta_{N,k}$	$\Delta\theta_{N,k}$	$u_{N,k+1}$	$\hat{\theta}_{N,k+1}$
baixa	baixa	N/A	baixa
baixa	N/A	regular	baixa
N/A	N/A	alta	baixa
regular	regular	regular	regular
alta	regular	alta	regular
baixa	regular	baixa	regular
alta	alta	N/A	alta
alta	N/A	regular	alta
N/A	N/A	baixa	alta

- **Treinamento do sistema neuro-fuzzy:** Para a obtenção do modelo, são encontrados os pesos dos consequentes do sistema neuro-fuzzy a partir do algoritmo de evolução diferencial que tem como critério de minimização o erro médio quadrático MSE definido pela Eq. (3.1):

$$MSE = \frac{\sum_{k=1}^{k_{\max}} (\hat{\theta}_k - \theta_{\text{exp}_k})^2}{k_{\max}}$$

Adota-se no processo de otimização 80 vetores, 100 iterações, fator de perturbação igual a 0,4 e taxa de cruzamento unitária.

As figuras a seguir comparam a resposta do modelo neuro-fuzzy, em azul, com os dados experimentais retirados robô, em vermelho. Na Fig. 4.5a são apresentados os dados utilizados para a obtenção do modelo, e na Fig. 4.5b, os dados de validação do sistema neuro-fuzzy.

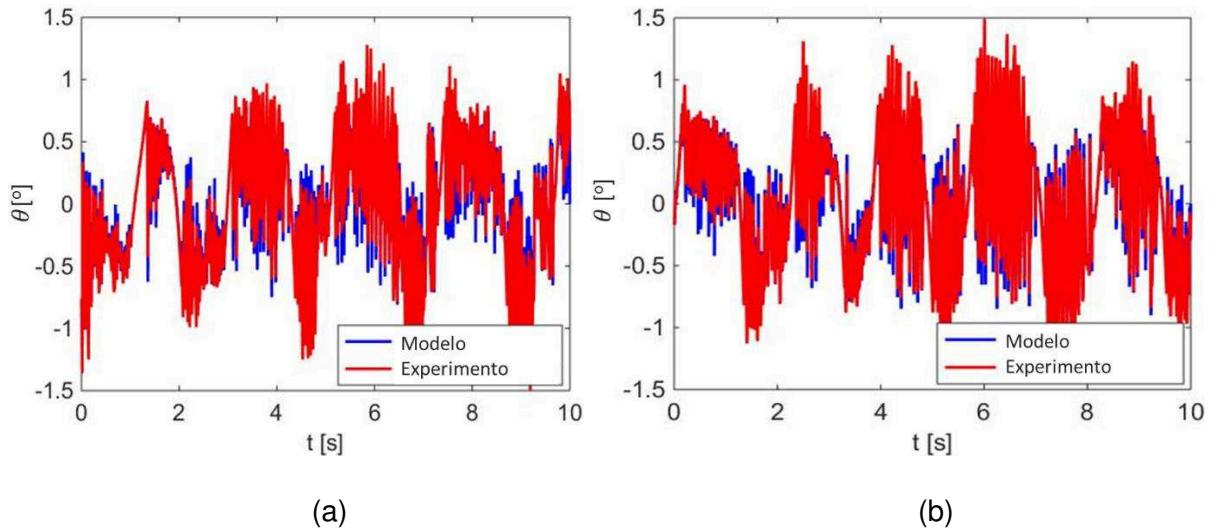


Figura 4.5: (a) Dados de obtenção do modelo e (b) dados de validação.

O método da evolução diferencial reduz o erro médio quadrático MSE com o passar das iterações, da forma apresentada na Fig. 4.6.

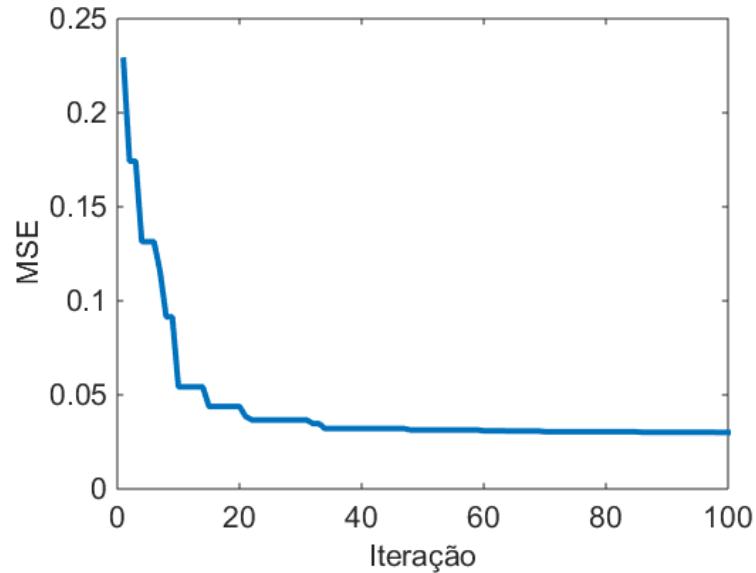


Figura 4.6: Evolução do processo de identificação dos parâmetros do modelo.

Ao fim da otimização do ANFIS, obtém-se o valor de MSE igual a 0,03 e taxa de ajuste do modelo aos dados experimentais, Eq. (2.33), de 89%, o que indica a alta capacidade do sistema neuro-fuzzy no ajuste de dados experimentais e na modelagem de sistemas.

4.2. Estudo das Incertezas Associadas aos Parâmetros do Modelo Neuro-Fuzzy

Esta subseção é dividida em dois subtópicos, sendo que o primeiro trata do estudo da incerteza associada à resposta da *IMU* utilizando para tal o método de Monte Carlo, e o segundo diz respeito propriamente ao estudo das incertezas associadas aos parâmetros do modelo neuro-fuzzy.

4.2.1. Estudo da Incerteza Associada à Resposta da *IMU*

Este subtópico tem como objetivo explanar a respeito da determinação da incerteza na estimativa do ângulo de inclinação θ do veículo de duas rodas tendo como valor convencional *VC* o ângulo obtido através de um esquadro de precisão calibrado.

A *IMU* MPU6050, Fig. 4.7, da fabricante InvenSense, é um sistema de baixo custo (aproximadamente R\$ 12,00) e se destina principalmente a smartphones, dispositivos móveis e em projetos de robótica de baixa precisão. Ela apresenta uma resolução de $0,01^\circ$ e faixa nominal de -90° a 90° , e a extração do ângulo de inclinação se dá através de uma biblioteca aberta e disponível aos desenvolvedores de programas na plataforma de prototipagem Arduino.

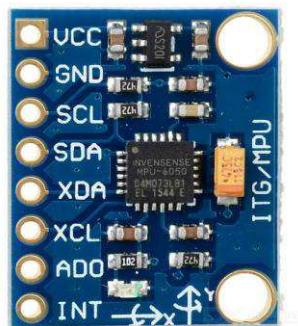


Figura 4.7: MPU6050.

O esquadro de precisão é da fabricante Mitutoyo, código 916 – 218, número de série 1000000, com dimensões nominais 400 x 250 mm, composto por aço inoxidável e apresenta o certificado de calibração número 09343/13. A Fig. 4.8 é uma foto do esquadro e a Fig. 4.9 apresenta o desvio de perpendicularidade do esquadro, extraído do certificado de calibração

do mesmo, além do sentido convencional do sinal tanto do esquadro, quanto do veículo de duas rodas.



Figura 4.8: Foto do esquadro de precisão.

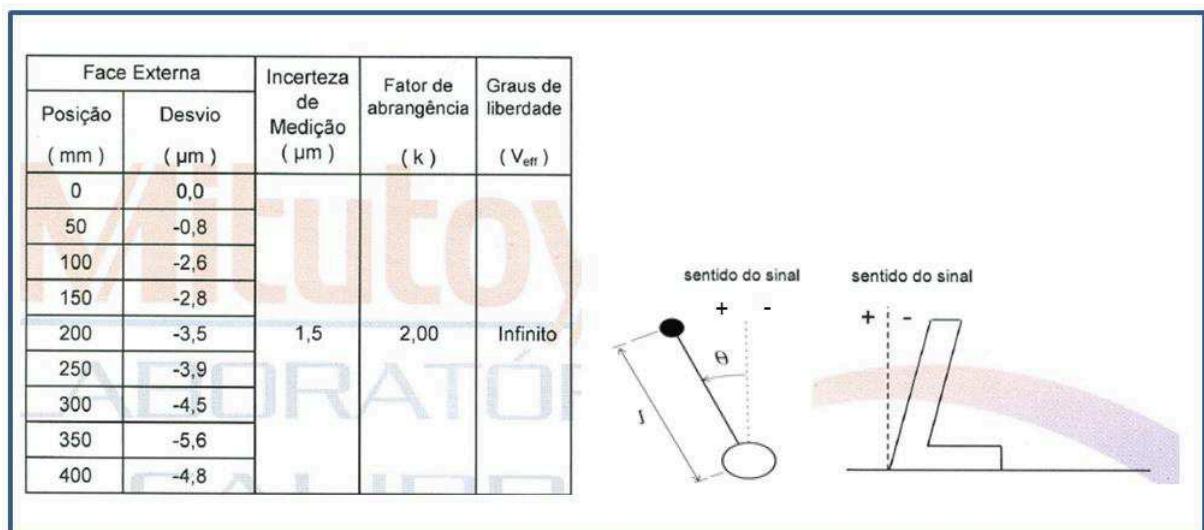


Figura 4.9: Desvio de perpendicularidade do esquadro de precisão.

Após o apoio do veículo na face externa do esquadro de precisão, a seguir são indicados os passos para o cálculo da incerteza de medição pelo método de Monte Carlo:

- **Definição da variável de saída:** A variável de saída analisada pelo método de Monte Carlo é o erro de medição (E);

- **Definição das variáveis de entrada:** As variáveis de entrada do problema são: a variabilidade das leituras da MPU6050 (\bar{X}_M), a resolução finita das leituras da MPU6050 R_M , a calibração da *IMU* (C_M), a calibração do esquadro de precisão (C_E) e a correção do esquadro de precisão (COR_E);
- **Modelo matemático:** O modelo matemático que relaciona a variável de saída às variáveis de entrada do processo estocástico é descrito a seguir:

$$E = \bar{X}_M - VC,$$

sendo VC o valor convencional do ângulo de inclinação do veículo. Dessa forma, tem-se que o erro E é dado por:

$$E = \bar{X}_M - (COR_E + \Delta C_E) + \Delta R_M + \Delta C_M; \quad (4.1)$$

- **Avaliação da incerteza padrão associada a cada variável de entrada:** A incerteza padrão de uma variável aleatória VA , $u(VA)$, define a abrangência do intervalo de confiança da variável ao nível padrão, ou seja, apresentando probabilidade de abrangência de 68,26%. A seguir são avaliadas as incertezas padrão associadas a cada variável de entrada:

- Variabilidade das leituras \bar{X}_M : A estimativa da incerteza padrão de \bar{X}_M se dá a partir de uma avaliação do tipo A, ou seja, é função dos dados disponíveis do ângulo de inclinação; apresenta distribuição normal, o valor da variável \bar{X}_M é igual a $0,0542^\circ$ e sua incerteza padrão é dada pela relação:

$$u(\bar{X}_M) = \frac{s}{\sqrt{n}} = \frac{s}{\sqrt{1}} = 0,0408^\circ,$$

em que s é o desvio padrão obtido através de 1168 medições e n é o número de medições utilizadas para cada tomada de decisão do veículo.

- Resolução finita da MPU6050 R_M : A estimativa da incerteza padrão $u(R_M)$ se dá através de uma avaliação do tipo B, ou seja, a avaliação não é função dos dados disponíveis do ângulo de inclinação, apresenta distribuição retangular e a incerteza padrão é calculada da forma indicada a seguir:

$$u(R_M) = \frac{R_M}{2\sqrt{3}} = \frac{0,01^\circ}{2\sqrt{3}} = 0,0028^\circ;$$

- Calibração da *IMU* C_M : Para o cálculo da incerteza padrão associada à calibração da MPU6050, não se dispõe de seu certificado de calibração, portanto, lança-se mão das técnicas apresentadas em (Vuolo, 1993), em que se sugere que a incerteza expandida $U(C_M)$ de um instrumento que não apresenta certificado de calibração seja dada pela relação $U(C_M) = \alpha R_M$, sendo α determinado intuitivamente de modo que a contribuição

de $u(C_m)$ na incerteza padrão combinada (incerteza padrão relacionada à variável de saída) esteja entre 20 e 30%. Assim, a avaliação da incerteza padrão $u(C_M)$ é do tipo B, segue distribuição normal e é dada a partir do desenvolvimento abaixo:

$$U(C_M) = 5R_M$$

$$u(C_M) = \frac{U(C_M)}{k} = \frac{5R_M}{2} = 0,0250^\circ,$$

sendo k o fator de abrangência, que neste caso transforma a incerteza expandida $U(C_M)$ (com probabilidade de abrangência de 95,45%) na incerteza padrão $u(C_M)$;

- Calibração do esquadro de precisão C_E : O esquadro de precisão, em contraste com a *IMU*, apresenta certificado de calibração. Dessa forma, tem-se que a avaliação da incerteza padrão relacionada ao esquadro de precisão é do tipo B, apresenta distribuição normal, e é calculada através da projeção angular da incerteza expandida do desvio de perpendicularidade $U(C_{PE})$ mencionado no certificado de calibração, Fig. 4.9, levando-se em conta a altura h do veículo de duas rodas:

$$U(C_{PE}) = 1,5 \mu m$$

$$h = 75 mm$$

$$U(C_E) = \operatorname{tg}^{-1} \left(\frac{U(C_{PE})}{h} \right)$$

$$k = 2$$

$$u(C_E) = \frac{U(C_E)}{k} = 0,0006^\circ$$

- Correção esquadro de precisão COR_E : Analogamente a C_E , para a determinação de COR_E , utiliza-se a projeção angular do desvio de perpendicularidade C_{PE} levando-se em consideração a altura h do veículo. Como não há precisamente o desvio de perpendicularidade para o valor de h no certificado de calibração, realiza-se uma interpolação linear entre os valores de 50 e 100 mm:

$$C_{PE} = f(x)$$

$$C_{PE}(50 mm) = -0,8 \mu m$$

$$C_{PE}(100 mm) = -2,6 \mu m$$

$$C_{PE}(h) = -1,7 \mu m$$

$$COR_E = \operatorname{tg}^{-1} \left(\frac{C_{PE}}{h} \right)$$

$$COR_E = -0,0013^\circ$$

- **Determinação do número de iterações n_{iter} :** Adota-se n_{iter} igual a 1 000 000, que equivale à quantidade de elementos a serem gerados aleatoriamente para cada variável de entrada, seguindo suas respectivas distribuições estatísticas;

- **Geração de números aleatórios:** Geração dos n_{iter} elementos para cada variável de entrada a partir dos comandos no MATLAB `rand()`, no caso de distribuições retangulares, e `randn()`, no caso de distribuições normais;
- **Avaliação da incerteza padrão combinada u_c :** A propagação das incertezas das variáveis de entrada é dada através da combinação dos números aleatórios a partir do modelo matemático da Eq. (4.1). A incerteza padrão combinada é então determinada pela relação $u_c(E) = s(E) = 0,0479^\circ$;
- **Calculo da incerteza expandida $U(E)$:** A incerteza expandida da variável de saída é dada pelo desenvolvimento matemático a seguir:

$$U(E) = k u_c(E)$$

$$U(E) = 2 \times 0,0479^\circ$$

$$U(E) = 0,0957^\circ$$

- **Resultado de medição RM :** O resultado de medição é expresso da forma apresentada abaixo:

$$RM = E \pm U(E)$$

$$RM = 0,0554^\circ \pm 0,0957^\circ, \text{ para } k=2 \text{ e probabilidade de abrangência de } 95,45\%.$$

A determinação da contribuição de cada variável de entrada na incerteza da resposta da *IMU* é apresentada em sequência:

$$C(\bar{X}_M) = \frac{\left(\frac{\partial E}{\partial \bar{X}_M}\right)^2 u^2(\bar{X}_M)}{u_c^2(E)} = 72,36 \%$$

$$C(R_M) = \frac{\left(\frac{\partial E}{\partial R_M}\right)^2 u^2(R_M)}{u_c^2(E)} = 0,36 \%$$

$$C(C_M) = \frac{\left(\frac{\partial E}{\partial C_M}\right)^2 u^2(C_M)}{u_c^2(E)} = 27,27 \%$$

$$C(C_E) = \frac{\left(\frac{\partial E}{\partial C_E}\right)^2 u^2(C_E)}{u_c^2(E)} = 0,01 \%$$

É possível verificar que a variabilidade das leituras do ângulo de inclinação é a variável de entrada que exerce maior contribuição na incerteza da resposta do sensor, o que se deve à baixa precisão da IMU utilizada.

A Fig. 4.10 é o histograma originado na saída da Eq. (4.1), obtida a partir do método de Monte Carlo. Quando o histograma se aproxima de uma distribuição normal, o número de

iterações do método é considerado adequado, e consequentemente há sucesso na determinação das incertezas.

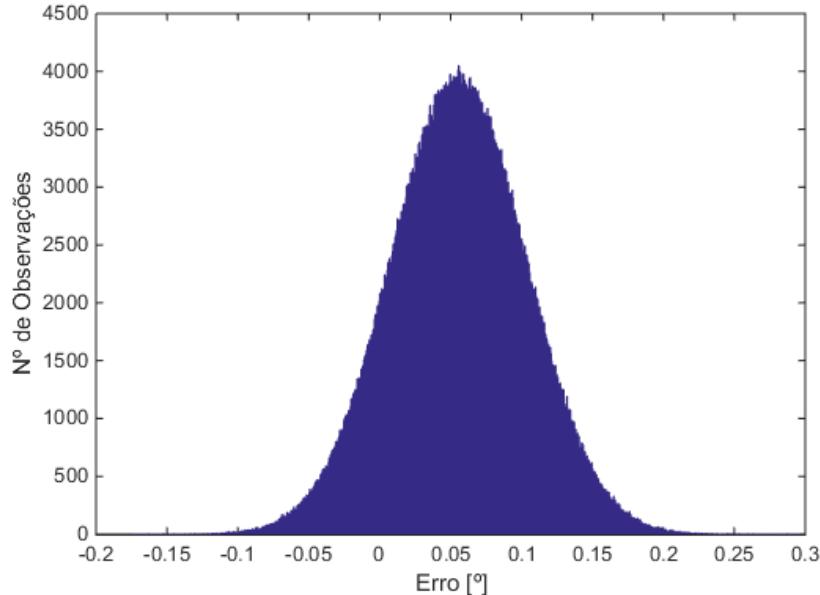


Figura 4.10: Histograma da saída do modelo do sensor.

A assimetria sk e curtoza $kurt$ dos dados apresentados na Fig. 4.10 são iguais respectivamente a -0,00007 e 2,9952, índices próximos aos apresentados por uma curva gaussiana ($sk = 0$ e $kurt = 3$). Realizam-se também testes de aderência do chi-quadrado e de Kolmogorov-Smirnov, e em ambos não foi rejeitada a hipótese nula de que, com nível de confiança de 95%, a resposta do modelo do sensor siga a distribuição normal. Esses são indicativos que, além da análise qualitativa do histograma, reforçam o sucesso da avaliação da incerteza expandida $U(E)$ associada à resposta da *IMU*.

A Tab. 4.2 é uma síntese dos resultados relativos às variáveis de entrada do processo.

Tabela 4.2: Variáveis de Entrada.

Variável de Entrada	\bar{X}_M	R_M	C_M	C_E
Estimativa	0,0542°	0,01°	0,05°	0,0012°
Tipo de Avaliação	A	B	B	B
Distribuição de Probabilidade	Normal	Retangular	Normal	Normal
Incerteza Padrão	0,0408°	0,0028°	0,0250°	0,0006°
Contribuição	72,36 %	0,36 %	27,27 %	0,01 %

Por fim, a Tab. 4.3 apresenta os resultados obtidos da incerteza associada à resposta da *IMU*.

Tabela 4.3: Incerteza da Resposta da *IMU*.

Incerteza padrão combinada (u_c)	0,0478°
Incerteza expandida (U)	0,0957°
Fator de abrangência (k)	2
Probabilidade de abrangência	95,45%

Após o estudo da incerteza ligada ao sistema de medição do ângulo de inclinação do veículo de duas rodas, o próximo subtópico trata especificamente da obtenção dos intervalos de confiança dos parâmetros do modelo neuro-fuzzy identificado na subseção 4.1.

4.2.2. Determinação dos Intervalos de Confiança dos Parâmetros do Modelo Neuro-Fuzzy

O modelo neuro-fuzzy adotado para o veículo de duas rodas pode ser representado de acordo com a Eq. (4.2):

$$\hat{\theta}_{N,k+1} = \frac{\sum_{L=1}^3 u_L^{(3)} (w_{0L} + w_{1L} u_{N,k+1} + w_{2L} \theta_{N,k} + w_{3L} \Delta \theta_{N,k})}{\sum_{L=1}^R u_L^{(3)}} \quad (4.2)$$

sendo $u_L^{(3)}$ a saída da terceira camada do ANFIS.

Para se extrair os intervalos de confiança dos pesos w_{jL} , utiliza-se a mesma metodologia usada no cálculo das incertezas dos coeficientes de uma curva obtida através do método da regressão (Montgomery e Runger, 2002). Dessa forma, é necessário que se apresente o modelo neuro-fuzzy conforme a Eq. (2.23):

$$Y_{[1x1]} = X_{[1xp]} b_{[px1]}$$

e os intervalos de confiança IC são então dados pela Eq. (2.31):

$$IC: b_{[px1]} - t_{1-\frac{\alpha}{2}} s_e \leq \beta_{[px1]} \leq b_{[px1]} + t_{1-\frac{\alpha}{2}} s_e$$

O modelo neuro-fuzzy representado de acordo com a Eq. (2.23) é apresentado a seguir:

$$\hat{\theta}_{N,k+1} = X_{[1xp]} W_{[px1]} \quad (4.3)$$

onde $W_{[px1]}$ é a matriz dos pesos do sistema neuro-fuzzy:

$$W_{[px1]} = \begin{pmatrix} w_{01} \\ w_{11} \\ w_{21} \\ w_{31} \\ w_{02} \\ w_{12} \\ w_{22} \\ w_{32} \\ w_{03} \\ w_{13} \\ w_{23} \\ w_{33} \end{pmatrix}$$

Nessa representação, porém, a matriz $X_{[1xp]}$ torna-se função das variáveis de entrada $u_{N,k+1}$, $\theta_{N,k}$ e $\Delta\theta_{N,k}$, sendo as duas últimas corrompidas pelo ruído inserido através da *IMU* do veículo de duas rodas. Essa situação, que contraria a restrição de que os elementos da matriz X devem ser isentos de ruídos, tem seu efeito minimizado a partir da aplicação de um filtro aos dados experimentais.

A Fig. 4.11 apresenta os dados brutos em azul, e em vermelho estão os dados filtrados a partir de um filtro de Kalman projetado através do método numérico exposto no apêndice A deste trabalho. Considera-se no projeto do filtro que A *IMU* esteja corrompida por ruído branco de média zero e desvio padrão σ_s igual à incerteza padrão combinada da *IMU*, ou seja, $\sigma_s = u_c = 0,0478^\circ$.

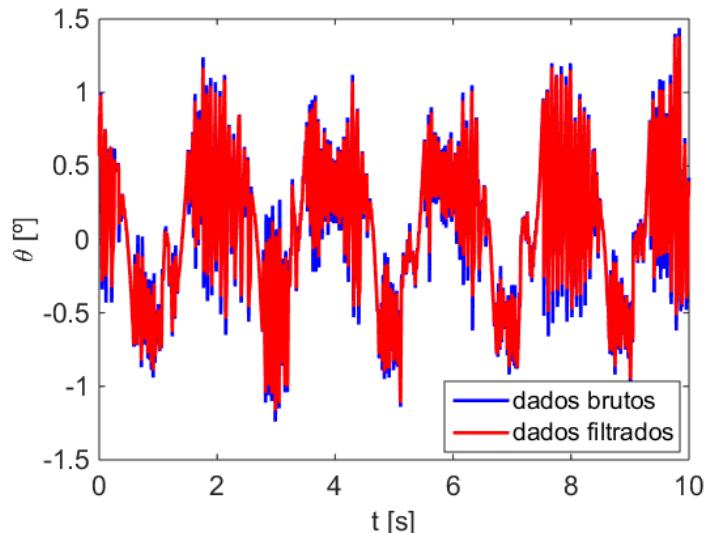


Figura 4.11: Representação da FDP da saída do modelo.

A partir dos dados filtrados, são extraídos os intervalos de confiança, Eq. (4.4), para cada um dos pesos do modelo neuro-fuzzy, com nível de confiança de 95%. A ordem da apresentação dos intervalos de confiança segue o ordenamento dos elementos da matriz $W_{[px1]}$:

$$IC: \begin{pmatrix} 0 \\ [0,00 \ 1,34] \\ [0,08 \ 0,12] \\ 0 \\ 0 \\ [0,83 \ 1,16] \\ [0,06 \ 0,10] \\ 0 \\ 0 \\ [0,05 \ 1,28] \\ [0,08 \ 0,12] \\ 0 \end{pmatrix} \quad (4.4)$$

Ressalta-se que os pesos que apresentavam intervalos de confiança que continham valores positivos e negativos, por não apresentarem efeito significativo na resposta do modelo, foram fixados iguais a zero.

4.3. Estudo da Incerteza Associada à Resposta do Modelo Neuro-Fuzzy

Após o cálculo das incertezas associadas à resposta da *IMU* e aos pesos do modelo neuro-fuzzy do veículo de duas rodas do tipo pêndulo invertido, resta avaliar a incerteza associada à resposta do modelo.

O procedimento de avaliação da incerteza relacionada à saída do modelo se pauta na propagação das incertezas ligadas ao modelo e à *IMU* por meio do método de Monte Carlo.

A seguir são indicados os passos para o cálculo da incerteza pelo método de Monte Carlo:

- **Definição da variável de saída:** Ângulo de inclinação estimado $\hat{\theta}_{k+1}$;
- **Definição das variáveis de entrada:** u_{k+1} , θ_k , $\Delta\theta_k$ e os pesos $w_{j,L}$;
- **Modelo matemático:**

$$\hat{\theta}_{k+1} = \frac{\sum_{L=1}^3 u_L^{(3)} (w_{0L} + w_{1L} u_{N,k+1} + w_{2L} \theta_{N,k} + w_{3L} \Delta\theta_{N,k})}{\sum_{L=1}^R u_L^{(3)}}$$

- **Avaliação da incerteza padrão associada a cada variável de entrada:**

- $u_{k+1}: u(u_{k+1}) = 0;$

- $\theta_k: u(\theta_k) = 0,0478^\circ$;
- $\Delta\theta_k: u(\Delta\theta_k) = u(\theta_k - \theta_{k-1}) = 0,0478\sqrt{2}^\circ$;
- $w_{j,L}$: A partir da Eq. (4.4) é possível extrair a incerteza padrão de cada um dos pesos $w_{j,L}$. A ordem da apresentação das incertezas padrão de cada um destes segue o ordenamento dos elementos da matriz $W_{[px1]}$:

$$u(w_{j,L}) = \begin{pmatrix} 0 \\ 0,34 \\ 0,01 \\ 0 \\ 0 \\ 0,08 \\ 0,01 \\ 0 \\ 0 \\ 0,31 \\ 0,01 \\ 0 \end{pmatrix}$$

- **Determinação do número de iterações n_{iter} :** Adota-se n_{iter} igual a 10000, que equivale à quantidade de simulações da operação do veículo, considerando que as entradas da planta são geradas aleatoriamente obedecendo as respectivas distribuições estatísticas e incertezas padrão de cada variável de entrada;
- **Geração de números aleatórios:** Geração dos valores aleatórios a partir do comando do MATLAB `randn()`;
- **Cálculo da incerteza padrão combinada u_c :** A propagação das incertezas das variáveis de entrada é dada através da combinação dos números aleatórios a partir do modelo matemático da Eq. (4.2). A incerteza padrão combinada é então calculada pela relação $u_c(\hat{\theta}_{k+1}) = s(\hat{\theta}_{k+1}) = 0,2867^\circ$;
- **Calculo da incerteza expandida $U(\hat{\theta}_{k+1})$:** A incerteza expandida da variável de saída é dada pelo desenvolvimento matemático a seguir:

$$U(\hat{\theta}_{k+1}) = k u_c(\hat{\theta}_{k+1})$$

$$U(\hat{\theta}_{k+1}) = 2 \times 0,2867^\circ$$

$$U(\hat{\theta}_{k+1}) = 0,5734^\circ$$

- **Resposta do modelo RM :** A resposta do modelo, considerando suas fontes de incerteza, é indica a seguir:

$$RM = \hat{\theta}_{k+1} \pm U(\hat{\theta}_{k+1})$$

$$RM = \hat{\theta}_{k+1} \pm 0,5734^\circ, \text{ para } k=2 \text{ e probabilidade de abrangência de } 95,45\%.$$

A Fig. 4.12 é um histograma da resposta do modelo em uma configuração em que a média da resposta $\bar{\theta}_{k+1} = 0,9^\circ$. A assimetria sk e curtose $kurt$ dos dados apresentados na Fig. 4.12 são iguais respectivamente a 0,0060 e 3,0920, índices próximos aos apresentados por uma curva gaussiana. Realizam-se também testes de aderência do chi-quadrado e de

Kolmogorov-Smirnov, e em ambos não foi rejeitada a hipótese nula de que, com nível de confiança de 95%, a resposta do modelo neuro-fuzzy siga a distribuição normal. Esses são indicativos que reforçam o sucesso da avaliação da incerteza expandida $U(\hat{\theta}_{k+1})$ associada à resposta do modelo da planta.

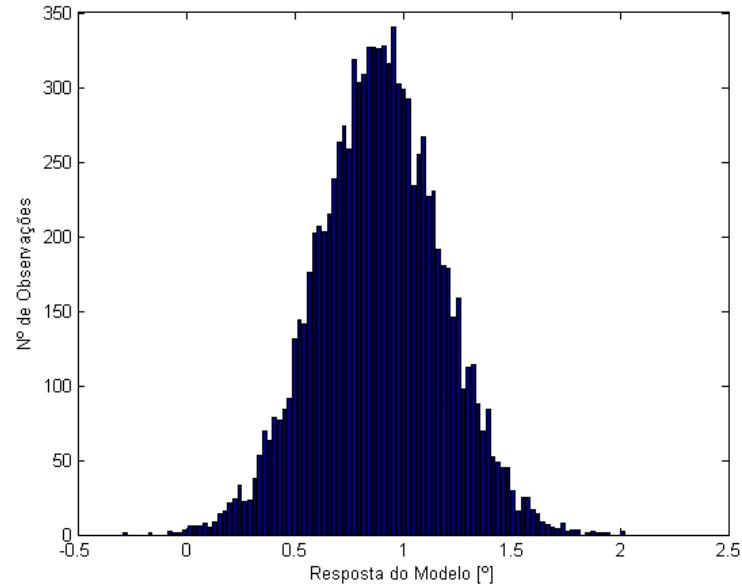


Figura 4.12: Histograma da saída do modelo da planta.

A Tab. 4.4 apresenta uma síntese dos resultados obtidos da incerteza associada à resposta do modelo neuro-fuzzy.

Tabela 4.4: Incerteza da Resposta do Modelo Neuro-Fuzzy.

Incerteza padrão combinada (u_c)	0,2867°
Incerteza expandida (U)	0,5734°
Fator de abrangência (k)	2
Probabilidade de abrangência	95,45%

Por fim, na Fig. 4.13 são mostrados os gráficos dos valores estimados pelo modelo neuro-fuzzy, em amarelo; os limites superior e inferior do intervalo de confiança para a resposta do modelo (com nível de confiança de 95,45%), respectivamente em vermelho e verde; e os dados experimentais extraídos do veículo de duas do tipo pêndulo invertido, em

azul. É possível verificar que os dados experimentais estão inteiramente situados dentro do intervalo de confiança, fato que evidencia a adequada identificação do modelo e de suas respectivas incertezas associadas.

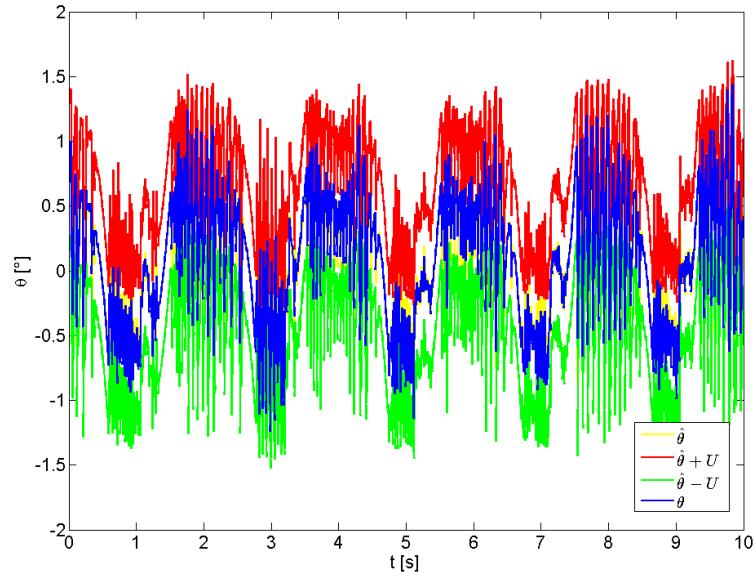


Figura 4.13: Resposta do modelo neuro-fuzzy.

É também possível verificar, a partir do procedimento de cálculo da incerteza, que a baixa precisão da IMU exerce grande influência na amplitude do intervalo de confiança associado à resposta do modelo. Sendo assim, a aquisição de um sensor inercial de maior qualidade poderia reduzir esse intervalo de confiança e consequentemente permitir o projeto e desenvolvimento de controladores mais eficientes para o veículo de duas rodas.

CAPÍTULO V

CONTROLE NEURO-FUZZY

Neste capítulo são apresentados os métodos utilizados de ajuste dos parâmetros dos controladores neuro-fuzzy, bem como os resultados obtidos via simulação e durante a operação do veículo para cada um dos métodos. É realizada a comparação das técnicas empregadas com os controladores clássicos, e por fim, desempenha-se uma análise estatística no intuito de se inferir sobre a eficiência de cada método testado.

Nas subseções 5.1 e 5.2, a sintonização dos parâmetros do controlador neuro-fuzzy é feita a partir de métodos de controle indireto adaptativo, nos quais os dados extraídos experimentalmente da bancada não estão ligados diretamente ao controle, mas sim à estimativa da dinâmica da planta, e dessa forma, podem ser executados em tempo real, ou de forma off-line, modo este escolhido devido às limitações de hardware do veículo de duas rodas.

Na subseção 5.3, porém, a sintonização dos parâmetros do controlador é realizada por meio de um método de controle direto adaptativo, ou seja, os dados extraídos da planta estão diretamente ligados à estimativa dos parâmetros do sistema de controle, e o método deve ser executado em tempo real.

5.1. Sintonização a Partir do Modelo Dinâmico

O controlador neuro-fuzzy, nesta subseção, é obtido através da simulação das condições de operação do veículo de duas rodas, levando-se em conta o modelo dinâmico apresentado na Eq. (2.46) e a zona morta de seus atuadores, Eq. (2.48). Este controlador é denominado DEDMN (Differential Evolution and Dynamical Model Based Neuro-Fuzzy Controller).

$$\xi_{k+1} = G\xi_k + Hu_k$$

$$u_k = \begin{cases} u_k^0, & u_k^0 \leq -\delta \\ 0, & -\delta < u_k^0 < +\delta \\ u_k^0, & u_k^0 \geq +\delta \end{cases}$$

A descrição da simulação para as condições de operação do veículo é apresentada a seguir:

- São simulados $T_t = 20\text{ s}$ de operação do sistema (tempo suficiente para verificar a capacidade do controlador em estabilizar estaticamente o veículo), em que cada iteração k da simulação, com um total de k_{\max} iterações, equivale ao tempo de amostragem $T_s = 0,005\text{ s}$;
- Assume-se que a condição inicial do veículo seja apenas um ângulo de inclinação θ_0 ;
- Admite-se que a *IMU* seja corrompida por ruído branco de média zero e desvio padrão σ_{acc} ;
- Tanto θ_k , quanto $\Delta\theta_k$, são corrompidos respectivamente por ruído branco de média zero e desvio padrão σ_θ e $\sigma_{\Delta\theta}$;
- Aplicam-se perturbações (i.e., funções impulso) ao ângulo de inclinação θ_k do sistema, com intensidade i_θ e repetidas n_d vezes;
- A aplicação das perturbações e ruídos mencionados acima ocorre até o tempo T_p , $T_p < T_t$; A partir de T_p , verifica-se a capacidade do controlador em retornar o sistema ao ponto de estabilidade sem a presença de fatores externos;
- A determinação das propriedades cinemáticas da estrutura no tempo kT_s é dada pela Eq. (2.46);
- Se o valor da ação de controle calculada pelo controlador no tempo k , u_k , estiver contido no intervalo $-\delta < u_k < +\delta$, impõe-se que o mesmo seja igual a zero, de acordo com a Eq. (2.48), de modo a se contabilizar na simulação a ação da zona morta dos motores no comportamento dinâmico do veículo;

A simulação é reproduzida para cada possível solução encontrada a partir do método da evolução diferencial, o qual busca minimizar a função custo (C_f) associada ao processo, Eq. (5.1):

$$C_f = \frac{\sum_{k=1}^{k_{max}} \lambda_1(u_k)^2 + \lambda_2(\theta_k - \theta_{ref})^2 + \lambda_3(\Delta\theta_k - \Delta\theta_{ref})^2}{k_{max}} \quad (5.1)$$

em que θ_{ref} é o ângulo de inclinação de referência, considerado nulo, $\Delta\theta_{ref}$ é a variação do ângulo de inclinação de referência, também considerado nulo, e as constantes λ_1 , λ_2 e λ_3 são utilizadas na imposição de prioridades no procedimento de minimização da função custo do processo. A constante λ_1 é ligada ao esforço de controle e à eficiência energética do sistema, enquanto as constantes λ_2 e λ_3 , por estarem relacionadas respectivamente ao ângulo de inclinação e à variação do ângulo de inclinação, estão ligadas à minimização da amplitude de oscilação do veículo. De modo a priorizar a minimização esforço de controle e a maximização da eficiência energética do sistema, adotam-se, por meio de tentativa e erro, os valores de λ_1 , λ_2 e λ_3 respectivamente iguais a 20, 2 e 0,5.

A Fig. 5.1 é um diagrama de blocos que representa matematicamente o sistema, em que θ é o ângulo de inclinação da planta, $\Delta\theta$ é a variação do ângulo de elevação, u é a ação de controle da planta, que corresponde à tensão de alimentação dos motores do veículo, k indica a iteração corrente, sendo que o intervalo entre duas iterações consecutivas é equivalente ao tempo de amostragem T_s adotado, e o bloco Z^{-D} é um indicativo de atraso de D iterações do sinal de entrada do mesmo.

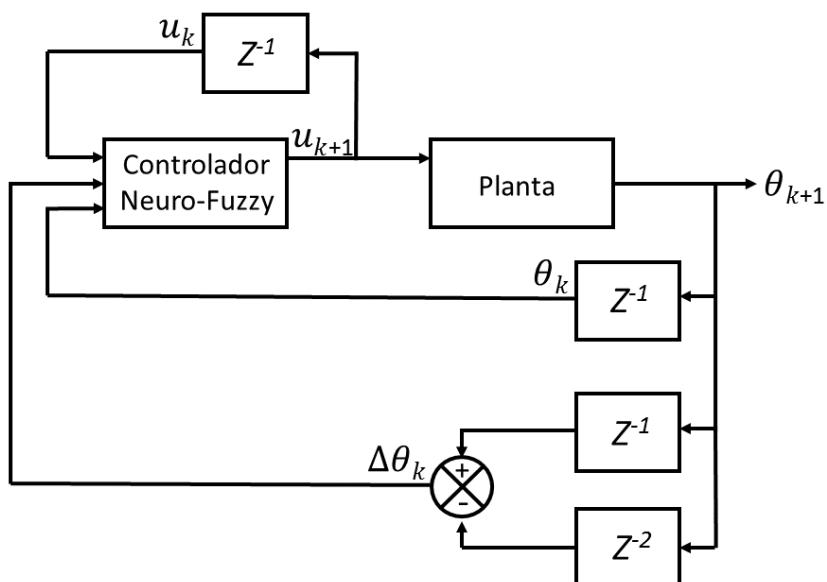


Figura 5.1: Representação do sistema em malha fechada.

A estrutura do controlador neuro-fuzzy projetado é indicada a seguir:

- **Variáveis de entrada:** Entradas normalizadas e na iteração k , $u_{N,k}$, $\theta_{N,k}$ e $\Delta\theta_{N,k}$.
- **Funções de pertinência das variáveis de entrada:** As Figs. 5.2a, 5.2b e 5.2c indicam, respectivamente, as funções de pertinência relativas às entradas normalizadas $u_{N,k}$, $\theta_{N,k}$ e $\Delta\theta_{N,k}$.

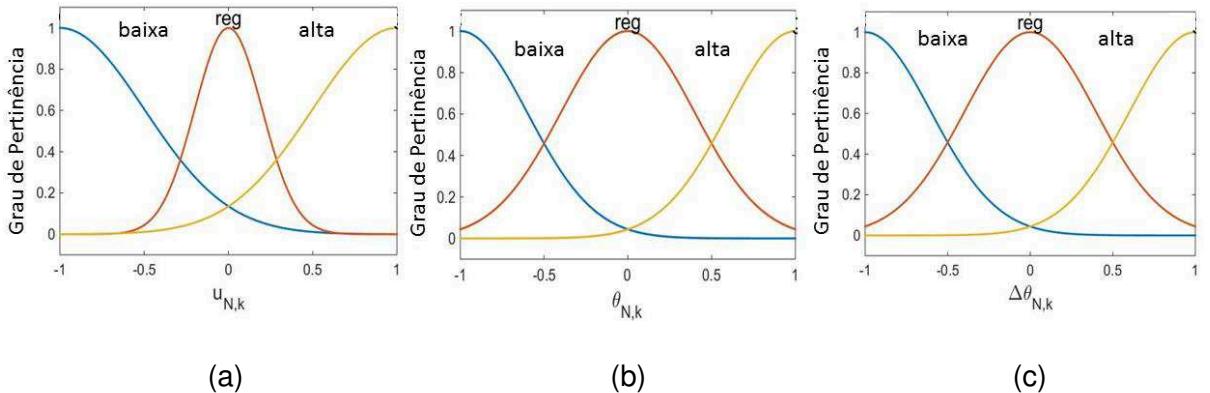


Figura 5.2: Funções de pertinência das variáveis de entrada (a) $u_{N,k}$, (b) $\theta_{N,k}$ e (c) $\Delta\theta_{N,k}$.

- **Variável de saída:** Tensão estimada para alimentação dos motores, normalizada e na iteração $k + 1$, $\hat{u}_{N,k+1}$;
- **Consequentes do sistema neuro-fuzzy:** Baseados no método de Takagi-Sugeno de primeira ordem e indicados em sequência:
 - $\hat{u}_{N,k+1}$ baixa: $f_1 = w_{01} + w_{11}u_{N,k} + w_{21}\theta_{N,k} + w_{31}\Delta\theta_{N,k}$;
 - $\hat{u}_{N,k+1}$ regular: $f_2 = w_{02} + w_{12}u_{N,k} + w_{22}\theta_{N,k} + w_{32}\Delta\theta_{N,k}$;
 - $\hat{u}_{N,k+1}$ alta: $f_3 = w_{03} + w_{13}u_{N,k} + w_{23}\theta_{N,k} + w_{33}\Delta\theta_{N,k}$.
- **Base de regras:** A base de regras do sistema neuro-fuzzy, construída a partir do conhecimento qualitativo do mecanismo, é apresentada na Tabela 5.1, em que N/A indica que a variável de entrada indicada não se aplica à regra em questão:

Tabela 5.1: Base de Regras do Controlador Neuro-Fuzzy.

SE			ENTÃO
$\theta_{N,k}$	$\Delta\theta_{N,k}$	$u_{N,k}$	$\hat{u}_{N,k+1}$
baixa	N/A	N/A	baixa
N/A	baixa	N/A	baixa
regular	regular	baixa	baixa
regular	regular	regular	regular
alta	N/A	N/A	alta
N/A	alta	N/A	alta
regular	regular	alta	alta

- **Treinamento do sistema neuro-fuzzy:** Para a obtenção do controlador, são encontrados os pesos dos consequentes do sistema neuro-fuzzy a partir do algoritmo de evolução diferencial que utiliza como função objetivo (f_{obj}) a função custo (Cf) associada ao processo, Eq. (5.1).

Anteriormente à realização dos testes experimentais, analisa-se em simulação o comportamento do veículo considerando a existência da zona morta em seus atuadores, condição esta que exerce forte influência no comportamento dinâmico da planta no que se refere à sua estabilização e no aumento de sua amplitude de oscilação (Peng e Dubay, 2011).

Dessa forma, no intuito de se analisar a influência da zona morta na resposta da planta, e a capacidade do sistema neuro-fuzzy de controlar um sistema submetido a essa não-linearidade, realizam-se simulações da operação do veículo de duas rodas, fixando o modelo identificado no capítulo III, porém variando o valor da amplitude da zona morta (δ) .

A ação de controle normalizada e a inclinação do veículo obtidas durante a simulação, utilizada no processo de obtenção dos parâmetros do controlador neuro-fuzzy, são apresentadas a seguir, em que nas Figs. 5.3a e 5.3b considera-se $\delta = 0 V$, $\delta = 0,6 V$ nas Figs. 5.4a e 5.4b, $\delta = 1,2 V$ nas Figs. 5.5a e 5.5b, e $\delta = 1,8 V$ nas Figs. 5.6a e 5.6b.

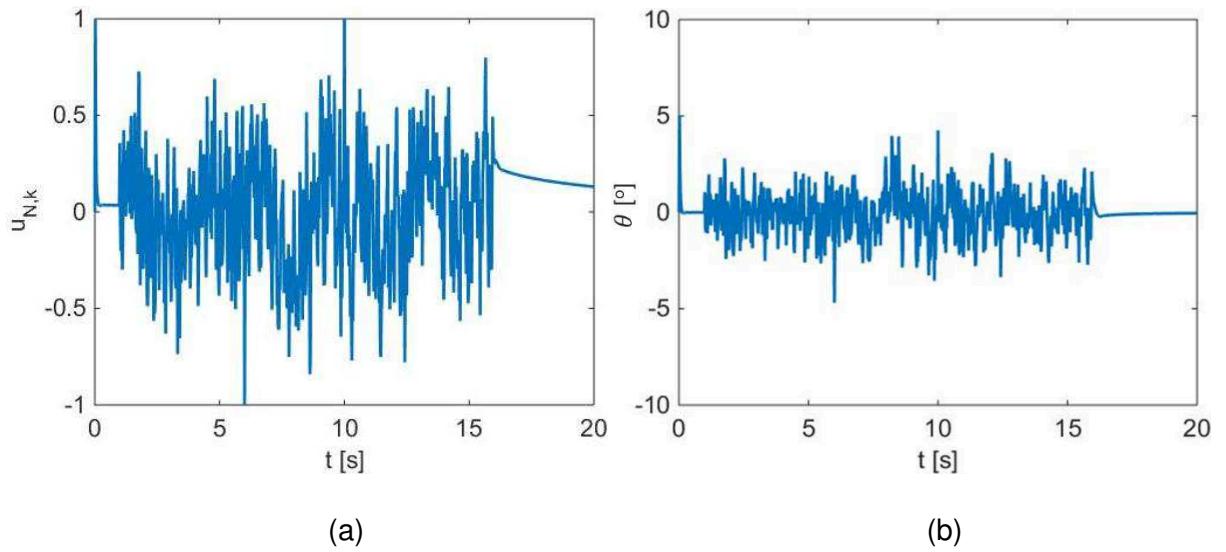


Figura 5.3: (a) Ação de controle normalizada e (b) ângulo de inclinação do veículo para $\delta = 0 V$.

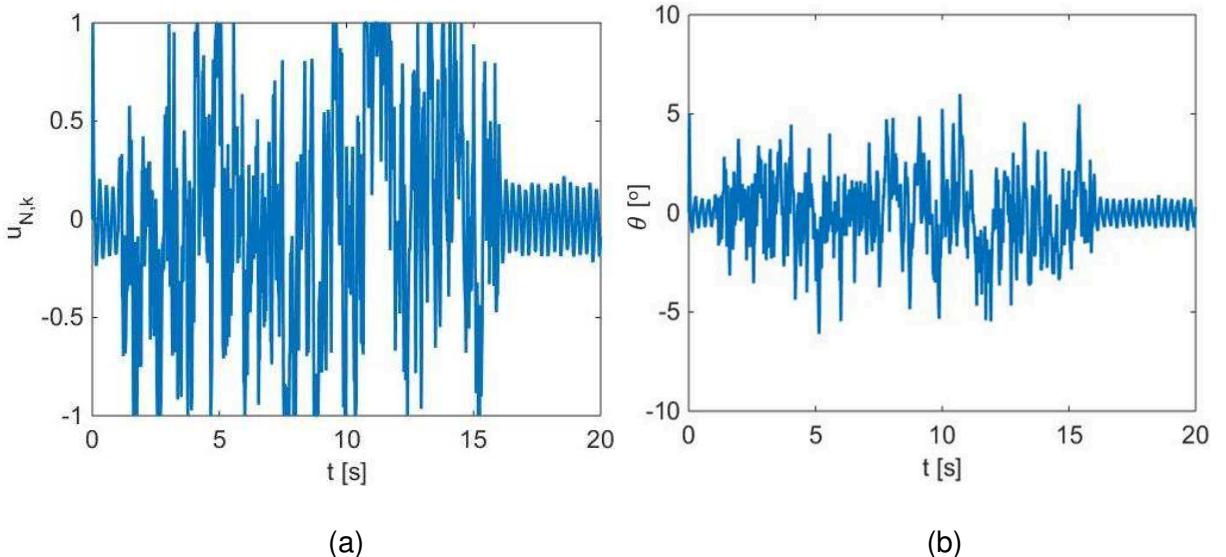


Figura 5.4: (a) Ação de controle normalizada e (b) ângulo de inclinação do veículo para $\delta = 0,6 V$.

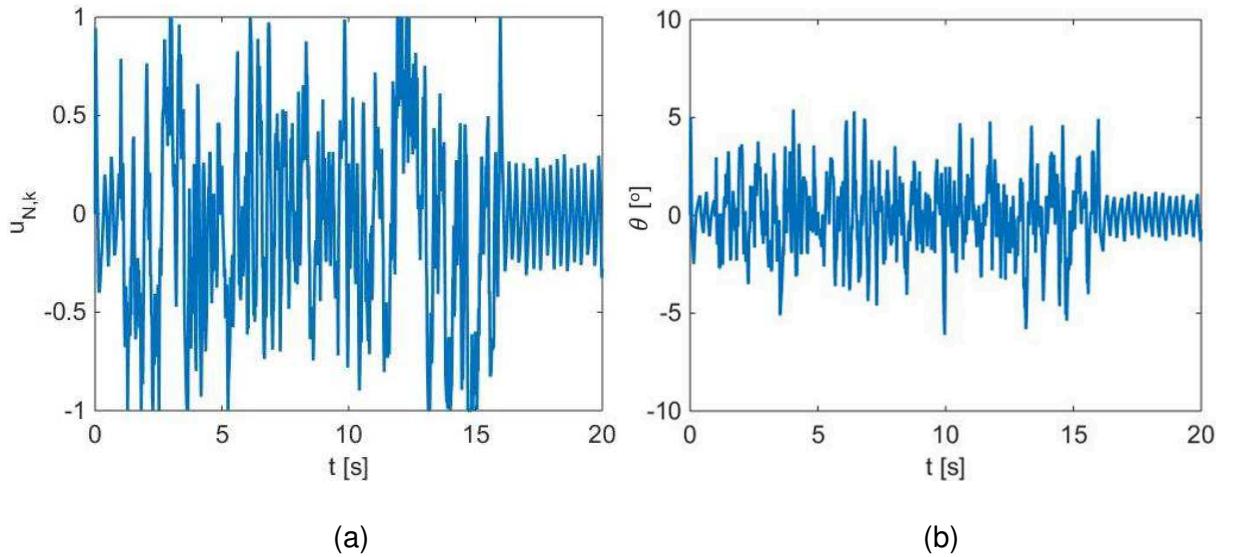


Figura 5.5: (a) Ação de controle normalizada e (b) ângulo de inclinação do veículo para $\delta = 1,2 V$.

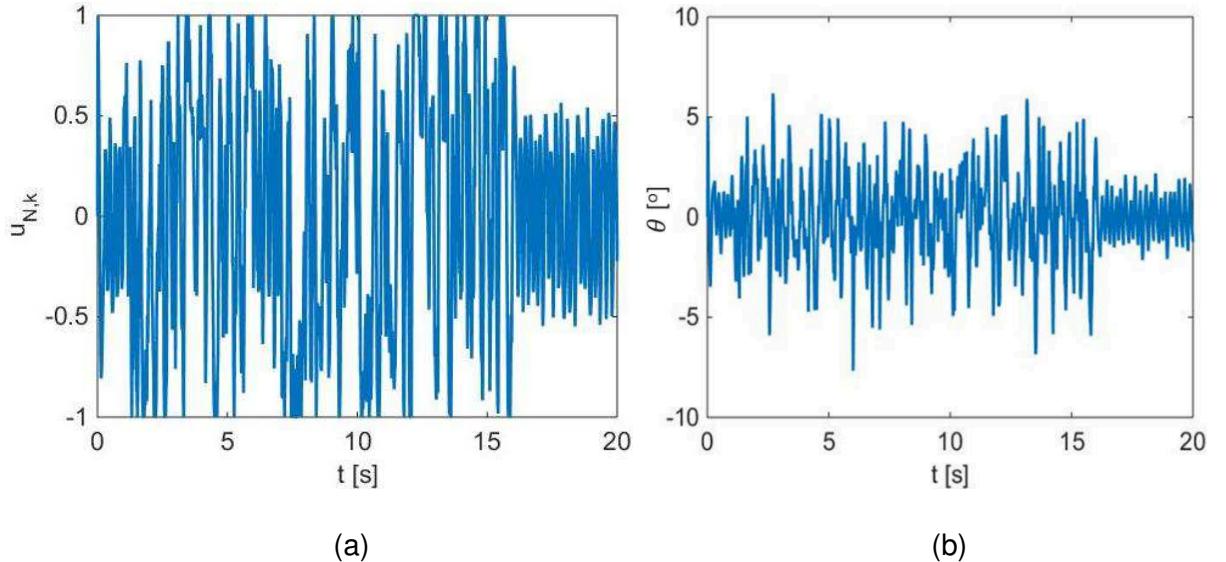


Figura 5.6: (a) Ação de controle normalizada e (b) ângulo de inclinação do veículo para $\delta = 1,8 V$.

A partir das Figs. 5.3, 5.4, 5.5 e 5.6, é possível verificar que à medida em que se aumenta a amplitude da zona morta dos atuadores, eleva-se a amplitude de oscilação da planta, o que está de acordo com a conclusão extraída em Peng e Dubay (2011). Não há, porém, nenhuma evidência de que o controlador neuro-fuzzy não seja capaz de controlar um sistema corrompido por tal não-linearidade.

Apresenta-se na Fig. 5.7 o gráfico da função custo C_f em função da evolução das iterações do algoritmo de evolução diferencial com 40 vetores, 50 iterações, fator de perturbação igual a 0,4 e taxa de cruzamento unitária. O modelo adotado do veículo considera a amplitude da zona morta igual ao valor identificado no capítulo III, ou seja, $\delta = 0,12 V$, e a partir dessa simulação, são determinados os parâmetros do controlador neuro-fuzzy que será implementado no veículo de duas rodas do tipo pêndulo invertido, indicados na Eq. (5.2).

$$\begin{pmatrix} w_{01} \\ w_{11} \\ w_{21} \\ w_{31} \\ w_{02} \\ w_{12} \\ w_{22} \\ w_{32} \\ w_{03} \\ w_{13} \\ w_{23} \\ w_{33} \end{pmatrix} = \begin{pmatrix} 0,20 \\ 0,02 \\ 0,11 \\ 0 \\ 0,16 \\ 0,10 \\ 0,07 \\ 0 \\ 0,20 \\ 0,02 \\ 0,11 \\ 0 \end{pmatrix} \quad (5.2)$$

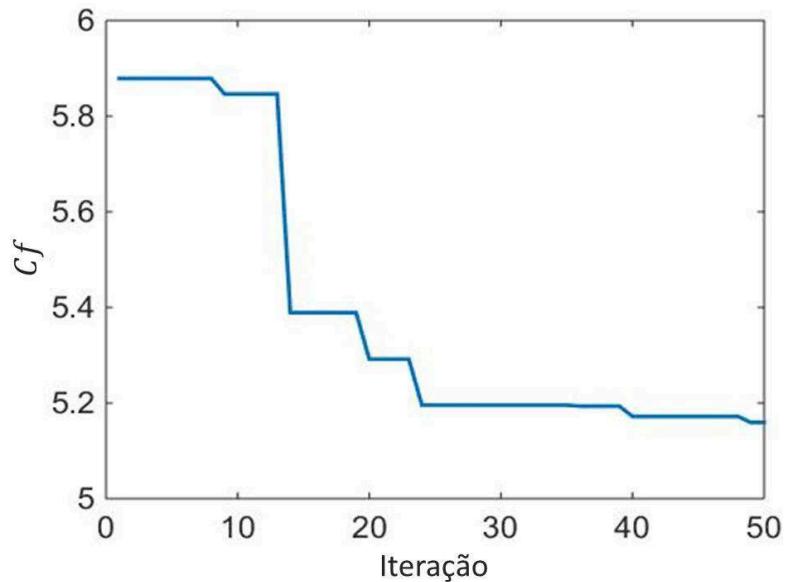


Figura 5.7: Evolução de C_f no processo de otimização dos parâmetros do controlador.

Nas Figs. 5.8a e 5.8b são apresentados respectivamente a ação de controle e o ângulo de inclinação obtidos, experimentalmente, a partir da implementação do controlador DEDMNF no veículo de duas rodas do tipo pêndulo invertido.

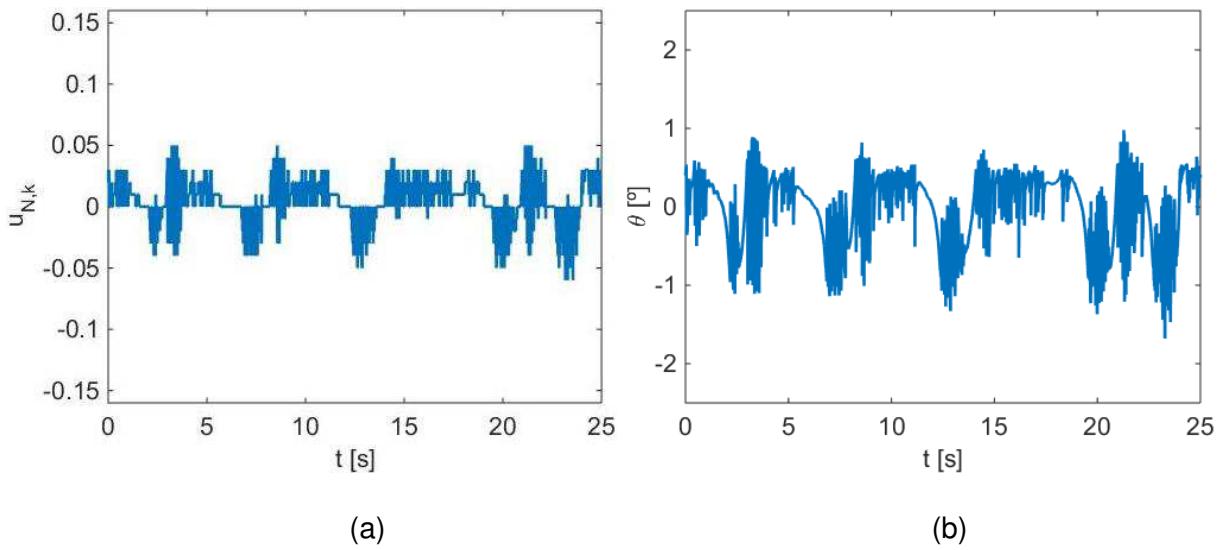


Figura 5.8: (a) Ação de controle e (b) ângulo de inclinação experimentais do veículo de duas rodas utilizando o controlador DEDMN.

5.2. Sintonização a Partir do Modelo Neuro-Fuzzy

Apresenta-se nesta subseção o controlador neuro-fuzzy obtido através da simulação das condições de operação do veículo de duas rodas, levando-se em conta o modelo neuro-fuzzy identificado no capítulo IV. O controlador proposto nesta subseção é denominado DENFNF (*Differential Evolution and Neuro-Fuzzy Model Based Neuro-Fuzzy Controller*).

Há vantagens em se utilizar o modelo neuro-fuzzy ao invés do modelo dinâmico do sistema. O modelo neuro-fuzzy é uma estrutura genérica não-linear, e portanto, permite que as não-linearidades intrínsecas ao sistema, como a zona morta, sejam incorporadas a ela durante seu treinamento. Além disso, se o ajuste do modelo for realizado em tempo real, este é capaz de contemplar as alterações impostas à planta, como por exemplo o acréscimo de massa ao sistema, e consequentemente, permitir o projeto de controladores que levem em consideração as mudanças impostas ao veículo.

O controlador DENFNF apresenta as mesmas variáveis de entrada e saída apresentadas na Fig. 5.1 e também a mesma estrutura que o controlador DEDMN, diferenciando unicamente o método de sintonização aplicado a cada um.

A descrição da simulação para as condições de operação do veículo, utilizada na sintonização dos parâmetros do controlador, é apresentada a seguir:

- São simulados $T_t = 10\text{ s}$ de operação do robô, em que cada iteração k da simulação, com um total de k_{\max} iterações, equivale ao tempo de amostragem $T_s = 0,005\text{ s}$;
- Assume-se que a condição inicial do veículo seja apenas um ângulo de inclinação θ_0 ;
- Admite-se que a *IMU* seja corrompida por ruído branco de média zero e desvio padrão σ_{acc} ;
- Tanto θ_k quanto $\Delta\theta_k$ são corrompidos respectivamente por ruído branco de média zero e desvio padrão σ_θ e $\sigma_{\Delta\theta}$;
- Aplicam-se perturbações (i.e., funções impulso) ao ângulo de inclinação θ_k do sistema, com intensidade i_θ e repetidas n_d vezes;
- A aplicação das perturbações e ruídos mencionados acima ocorre até o tempo T_p , $T_p < T_t$. A partir de T_p , verifica-se a capacidade do controlador em retornar o sistema ao ponto de estabilidade sem a presença de fatores externos.
- A determinação das propriedades cinemáticas da estrutura no tempo kT_s é dada pelo modelo neuro-fuzzy identificado no capítulo IV;

A simulação é reproduzida para cada possível solução encontrada a partir do método da evolução diferencial, o qual busca minimizar a função custo Cf associada ao processo, Eq. (5.1):

$$Cf = \frac{\sum_{k=1}^{k_{\max}} \lambda_1(u_k)^2 + \lambda_2(\theta_k - \theta_{ref})^2 + \lambda_3(\Delta\theta_k - \Delta\theta_{ref})^2}{k_{\max}}$$

Adotam-se os mesmos valores de λ_1 , λ_2 e λ_3 assumidos para o controlador DEDMF, ou seja, respectivamente iguais a 20, 2 e 0,5.

Os resultados da simulação são apresentados a seguir, em que na Eq. (5.3) são indicados os parâmetros obtidos para o controlador DENFNF por meio da evolução diferencial, a Fig. 5.9a apresenta a ação de controle normalizada do sistema em função do tempo ($u_{N,k}$) e a Fig. 5.9b representa o ângulo de inclinação do veículo de duas rodas.

$$\begin{pmatrix} w_{01} \\ w_{11} \\ w_{21} \\ w_{31} \\ w_{02} \\ w_{12} \\ w_{22} \\ w_{32} \\ w_{03} \\ w_{13} \\ w_{23} \\ w_{33} \end{pmatrix} = \begin{pmatrix} 0,05 \\ 0,05 \\ 0,08 \\ 0 \\ 0,05 \\ 0,05 \\ 0,07 \\ 0 \\ 0,05 \\ 0,05 \\ 0,08 \\ 0 \end{pmatrix} \quad (5.3)$$

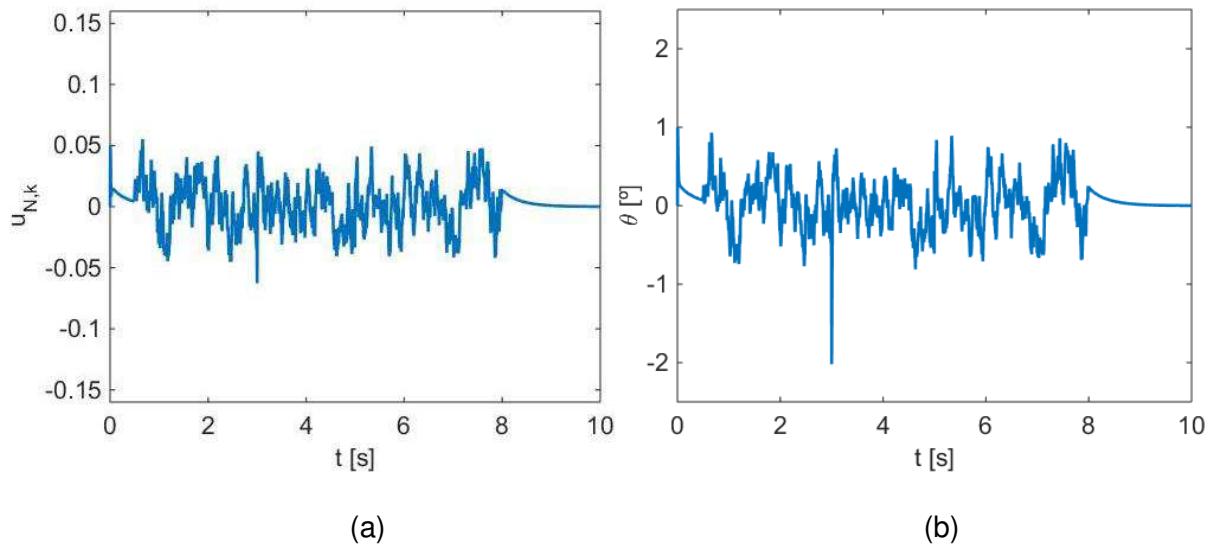


Figura 5.9: (a) Ação de controle e (b) ângulo de inclinação simulados do veículo de duas rodas.

A evolução da minimização da função custo C_f , Eq. (5.1), em função da iteração do processo é apresentada na Fig. 5.10. Utiliza-se para tal o algoritmo de evolução diferencial com 50 vetores, 60 iterações, fator de perturbação igual a 0,4 e taxa de cruzamento unitária.

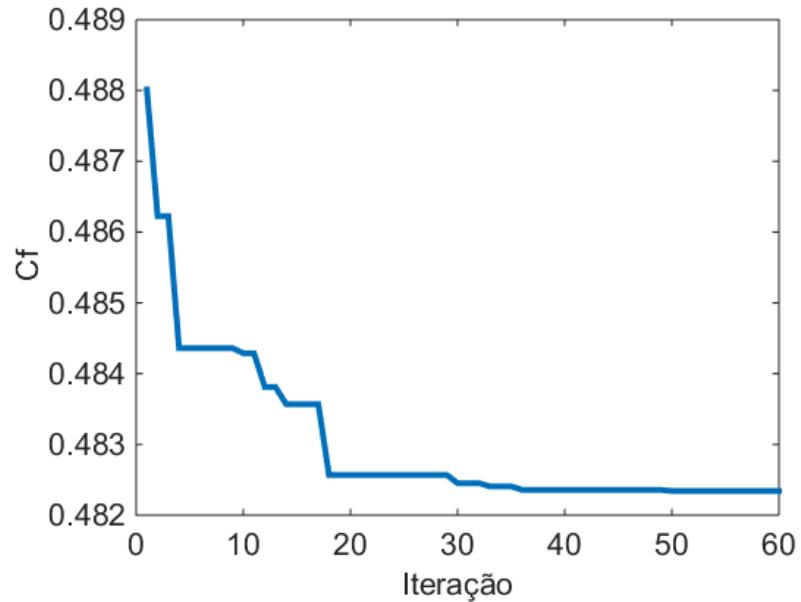


Figura 5.10: Evolução de C_f no processo de otimização dos parâmetros do controlador.

Os parâmetros obtidos através da simulação são utilizados experimentalmente no veículo de duas rodas, e durante a operação do mesmo, extraem-se os dados da ação de

controle normalizada, Fig. 5.11a, e do ângulo de inclinação, Fig. 5.11b. É válido ressaltar que a estabilização estática do sistema em valores diferentes de zero se dá devido à compensação que o torque do motor fornece à componente da força peso que tende a retirar o sistema da posição de equilíbrio estático $\theta = 0$.

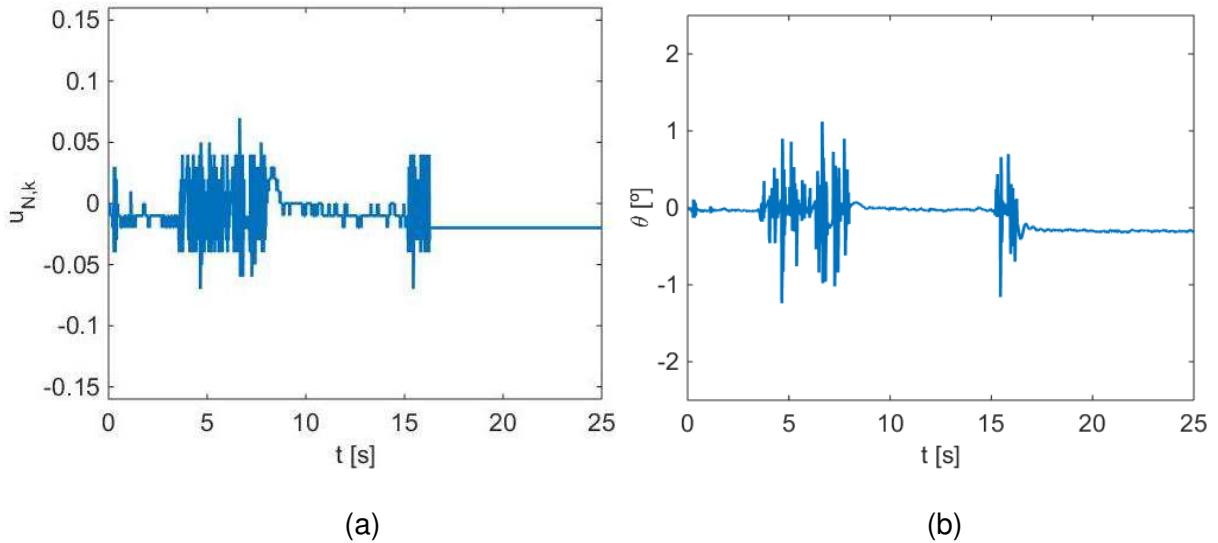


Figura 5.11: (a) Ação de controle e (b) ângulo de inclinação experimentais do veículo de duas rodas utilizando o controlador DENFNF.

Ao se comparar qualitativamente os resultados experimentais dos controladores DEDMNF e DENFNF, apresentados respectivamente nas Figs. 5.8 e 5.11, é possível notar que o controlador que leva em conta o modelo neuro-fuzzy da planta obtém maior êxito, tanto na redução da amplitude de oscilação do veículo, quanto na minimização do esforço de controle requerido para a estabilização estática do sistema.

5.3. Sintonização a Partir do Método OnMoDE

Essa subseção é a única que trata de um método de controle direto adaptativo, ou seja, os dados extraídos da planta estão diretamente ligados à estimativa dos parâmetros do sistema de controle, e deve ser executado em tempo real.

O método desenvolvido é denominado OnMoDE (*Online Modified Differential Evolution*) e objetiva sintonizar os parâmetros de controladores inteligentes a partir de uma

adaptação da evolução diferencial. Isto permite que esta técnica possa ser utilizada em tempo real em problemas de controle de sistemas não-lineares e sobretudo variantes no tempo.

O método é dividido em 2 etapas, sendo que a primeira lida com a geração inicial dos vetores, e a segunda, com o processo de otimização do controlador inteligente, e que dura durante toda a operação do veículo após a geração inicial dos vetores.

A Fig. 5.12 mostra um fluxograma representativo da etapa 1 do algoritmo. Cada passo desse algoritmo é detalhado em sequência.

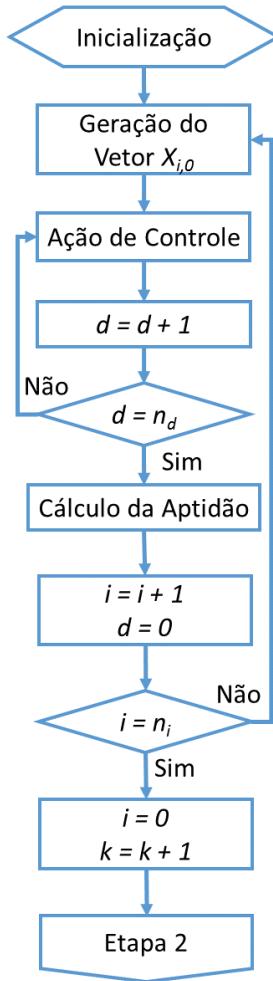


Figura 5.12: Fluxograma da etapa 1 do algoritmo OnMoDE.

- **Inicialização:** Nesse passo, o número de vetores n_i do conjunto S e o número de dados n_d utilizados para calcular a aptidão de cada vetor são definidos. É também definido o valor inicial da probabilidade de mutação crítica $prob_{cmut}$, e finalmente, as variáveis i , d e k são inicializadas com valor nulo;

- **Geração do Vetor $X_{i,0}$:** A geração de cada elemento dos vetores seguem uma distribuição uniforme definida entre o intervalo $[0,1]$, rand $(0,1)$, em que os valores limite 0 e 1 são relacionados aos limites do intervalo de busca de cada variável relacionada ao problema de otimização;
- **Ação de Controle:** Esse passo é relacionado com a determinação da ação de controle aplicada em tempo real no sistema de controle baseada nos parâmetros obtidos a partir do vetor $X_{i,k}$. Além disso, são coletados dados experimentais para a avaliação da performance do controlador;
- **Cálculo da Aptidão de $X_{i,0}$:** A estimativa da aptidão do vetor $X_{i,0}$ com respeito à função objetivo f_{obj} é computada. Para este propósito, os dados experimentais extraídos do passo anterior são utilizados.

A segunda etapa do algoritmo OnMoDE é representada na Fig. 5.13, e em sequência é apresentado cada passo da etapa 2.

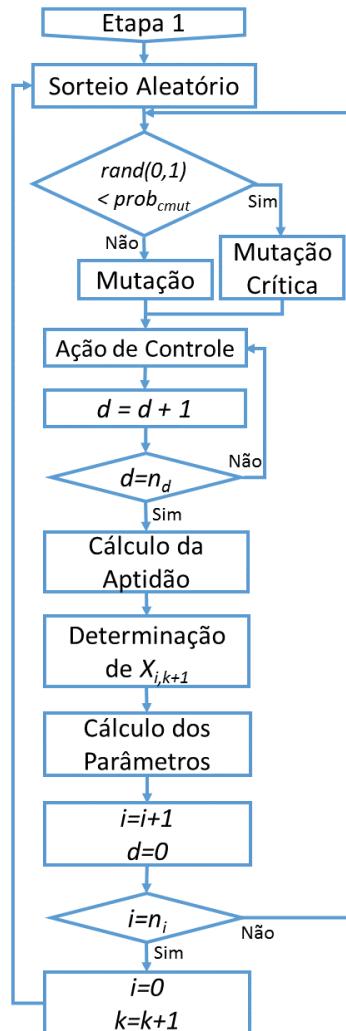


Figura 5.13: Fluxograma da etapa 2 do algoritmo OnMoDE.

- **Sorteio Aleatório:** Nesse passo, todas as possíveis mutações na k -ésima iteração são pré-determinadas baseadas no sorteio dos vetores $X_{j,k}$ e $X_{l,k}$;
- **Mutação:** Consiste na obtenção do vetor candidato a participar da iteração $k + 1$, $C_{i,k}$, a partir da Eq. (5.4), onde F também segue distribuição uniforme definida dentro do intervalo $[0,1]$:

$$C_{i,k} = X_{i,k} + F(X_{j,k} - X_{l,k}) \quad (5.4)$$

Na intenção de reduzir o custo computacional do algoritmo, não se realiza o procedimento padrão de mutação do método da evolução diferencial, Eqs. (2.17) e (2.18):

$$V_{i,k} = X_{i,k} + F(X_{j,k} - X_{l,k})$$

$$C_{e,i,k} = \begin{cases} V_{e,i,k}, & \text{SE } \text{rand}(0,1) \leq CR \text{ OU } e = \text{randnum}(1, n_e) \\ X_{e,i,k}, & \text{SE } \text{rand}(0,1) > CR \text{ E } e \neq \text{randnum}(1, n_e) \end{cases}$$

Essa alteração reduz consideravelmente o uso de laços condicionais na constituição do algoritmo, e consequentemente reduz o tempo de execução do método. Para garantir maior diversidade do mesmo, é aplicada também uma outra técnica de mutação, denominada mutação crítica;

- **Mutação Crítica:** A mutação crítica promove maior alteração nos vetores que a operação clássica de mutação. Esse passo pode ser desenvolvido de várias formas, dentre as quais:

$$\circ \quad C_{i,k} = \text{rand}(0,1) \quad (5.5)$$

$$\circ \quad C_{i,k} = N(0.5, \sigma_{cmut}) \quad (5.6)$$

$$\circ \quad C_{i,k} = X_{i,k} + N(0, \sigma_{cmut}) \quad (5.7)$$

- **Ação de Controle:** Procedimento idêntico ao passo de mesmo nome na etapa 1 do algoritmo;
- **Cálculo da Aptidão do Vetor $C_{i,k}$:** Procedimento análogo ao “Cálculo da Aptidão do Vetor $X_{i,0}$ ” da etapa 1 do método OnMoDE;
- **Determinação de $X_{i,k+1}$:** A determinação do vetor $X_{i,k+1}$ é dada pela Eq. (2.19):

$$X_{i,k+1} = \begin{cases} X_{i,k}, & \text{se } f_{obj}(X_{i,k}) \leq f_{obj}(C_{i,k}) \\ C_{i,k}, & \text{outro caso} \end{cases}$$

- **Cálculo dos Parâmetros:** Esse passo usa o histórico de informações da função objetivo f_{obj} associada ao vetor $X_{i,k}$ para atualizar o valor dos parâmetros σ_{cmut} e $prob_{cmut}$. Exemplos de métodos de atualização dos mesmos são indicados abaixo:

$$\circ \quad \sigma_{cmut} = \begin{cases} \sigma_1, & \text{if } \frac{f_{obj}(X_{i,k+1}) - f_{obj}(X_{i,k})}{f_{obj}(X_{i,k})} > ref_{cmut} \\ \sigma_2, & \text{if } \frac{f_{obj}(X_{i,k+1}) - f_{obj}(X_{i,k})}{f_{obj}(X_{i,k})} \leq ref_{cmut} \end{cases} \quad (5.8)$$

$$\circ \quad prob_{cmut} = \begin{cases} prob_1, & \text{if } \frac{f_{obj}(X_{i,k+1}) - f_{obj}(X_{i,k})}{f_{obj}(X_{i,k})} > ref_{cmut} \\ prob_2, & \text{if } \frac{f_{obj}(X_{i,k+1}) - f_{obj}(X_{i,k})}{f_{obj}(X_{i,k})} \leq ref_{cmut} \end{cases} \quad (5.9)$$

em que os valores de σ_1 , σ_2 , $prob_1$, $prob_2$ and ref_{cmut} são definidos no processo de inicialização do algoritmo.

A efetividade da nova metodologia de ajuste de parâmetros de controladores inteligentes é verificada a partir da estabilização estática do veículo de duas rodas e na comparação dos resultados desse método com as demais metodologias de controle. Nesse tipo de problema, pois, é altamente desejável que o controlador seja adaptativo devido ao fato de que as propriedades mecânicas e elétricas do sistema possam ser alteradas durante a operação do mesmo.

O controlador desta subseção apresenta as mesmas variáveis de entrada e saída apresentadas na Fig. 5.1 e também a mesma estrutura que os controladores DEDMF e DENFNF, diferenciando unicamente o método de sintonização aplicado.

A descrição da simulação para as condições de operação do veículo, utilizada neste caso apenas na verificação da convergência do método (visto que o ajuste dos parâmetros do controlador é realizado em tempo real durante a operação do veículo), é apresentada a seguir:

- A operação do veículo é simulada por $T_t = 80$ s (opta-se por um tempo de simulação maior em relação aos demais controladores no intuito de se verificar a adaptabilidade do método), com taxa de amostragem $T_s = 0,005$ s;
- Assume-se que a condição inicial do veículo seja apenas um ângulo de inclinação θ_0 ;
- A *IMU* é corrompida por ruído branco de média zero e desvio padrão σ_{acc} ;
- Tanto θ_k quanto $\Delta\theta_k$ são corrompidos respectivamente por ruído branco de média zero e desvio padrão σ_θ e $\sigma_{\Delta\theta}$;
- Aplicam-se perturbações (i.e., funções impulso) ao ângulo de inclinação θ_k do sistema, com intensidade i_θ e repetidas n_d vezes;

- A aplicação das perturbações e ruídos mencionados acima ocorre até o tempo $T_p, T_p < T_t$. A partir de T_p , verifica-se a capacidade do controlador em retornar o sistema ao ponto de estabilidade sem a presença de fatores externos;
- A determinação das propriedades cinemáticas da estrutura no tempo kT_s é dada pela Eq. (2.46);
- Se o valor da ação de controle calculada pelo controlador no tempo k , u_k , estiver contido no intervalo $-\delta < u_k < +\delta$, impõe-se o mesmo igual a zero, de acordo com a Eq. (2.48), de modo a se contabilizar na simulação a ação da zona morta dos motores no comportamento dinâmico do veículo;

A simulação considera a planta de acordo com o modelo apresentado no capítulo III, e a atualização dos parâmetros do controlador neuro-fuzzy é realizada a partir do método OnMoDE. Neste, utiliza-se como função objetivo f_{obj} a função custo Cf associada ao processo, Eq. (5.1):

$$Cf = \frac{\sum_{k=1}^{k_{max}} \lambda_1(u_k)^2 + \lambda_2(\theta_k - \theta_{ref})^2 + \lambda_3(\Delta\theta_k - \Delta\theta_{ref})^2}{k_{max}}$$

Adotam-se os mesmos valores de λ_1 , λ_2 e λ_3 assumidos para os controladores DEDMNF e DENFNF, ou seja, respectivamente iguais a 20, 2 e 0,5.

Os resultados da simulação do uso do método OnMoDE são apresentados a seguir, em que na Figura 5.14a é apresentada a ação de controle normalizada do sistema em função do tempo, $u_{N,k}$, e na Figura 5.14b está representado o ângulo de inclinação do veículo de duas rodas.

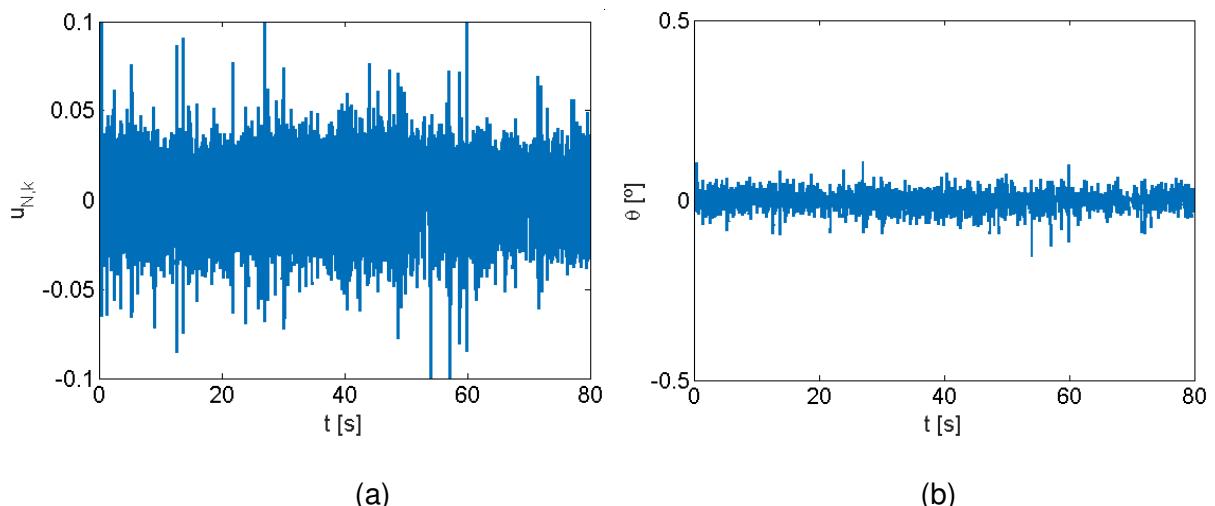


Figura 5.14: (a) Ação de controle e (b) ângulo de inclinação simulados do veículo de duas rodas.

A evolução da função custo no tempo é mostrada na Fig. 5.15. É possível verificar a tendência de redução da amplitude da função custo à medida que se evolui o tempo. Isso evidencia a capacidade do método OnMoDE em ajustar os parâmetros do controlador neuro-fuzzy de modo a garantir melhor performance do sistema.

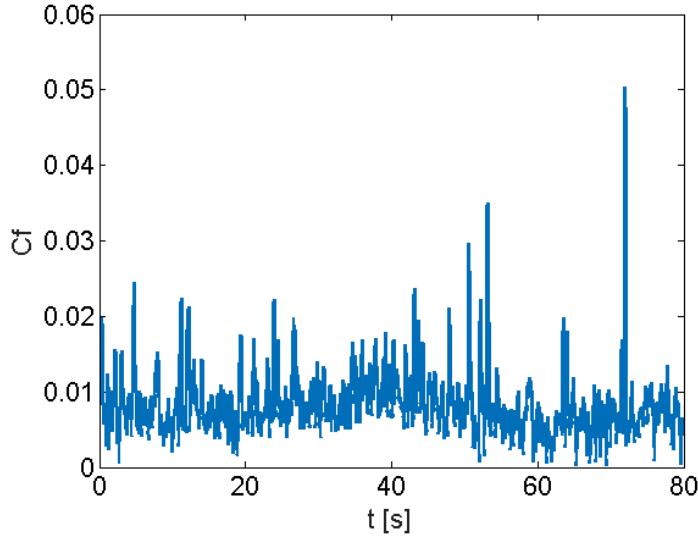


Figura 5.15: Evolução do valor da função custo no tempo obtida durante a simulação.

Após verificar em simulação que o método OnMoDE é capaz de manter o veículo de duas rodas em operação, implementa-se o mesmo na bancada experimental. Os dados experimentais de $u_{N,k}$ e θ_k extraídos do pêndulo invertido ao se utilizar o método OnMoDE são apresentados respectivamente nas Figs. 5.16a e 5.16b.

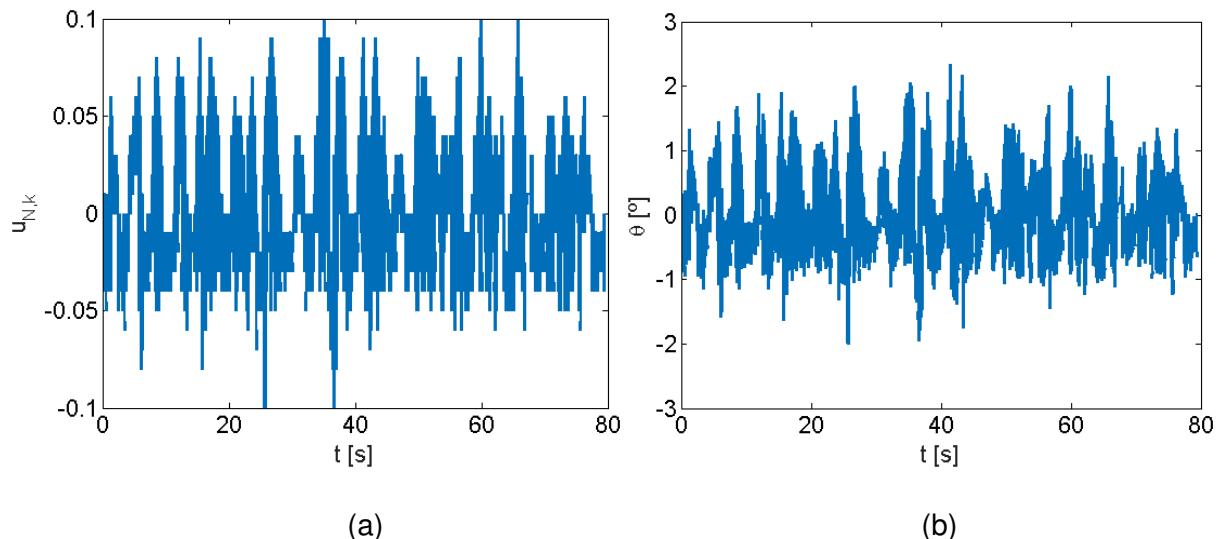


Figura 5.16: (a) Ação de controle e (b) ângulo de inclinação experimentais do veículo de duas rodas.

Ao se comparar qualitativamente os resultados experimentais dos controladores DEDMNf e DENFNf com o método OnMoDE, apresentados respectivamente nas Figs. 5.8, 5.11 e 5.16, é possível notar que o método proposto nesta subseção apresenta resultados inferiores aos demais. Isso se deve sobretudo ao fato de que as soluções iniciais ($X_{i,0}$), impostas no início do método OnMoDE, podem estar distantes da solução ótima do problema de controle e assim causar maior divergência do resultado.

A evolução de C_f em função do tempo é indicada na Fig. 5.17. Ambas, a função custo e a inclinação do veículo de duas rodas do tipo pêndulo invertido são minimizadas com a progressão do tempo, o que são indícios da capacidade de adaptação do controlador neuro-fuzzy aliado ao método OnMoDE.

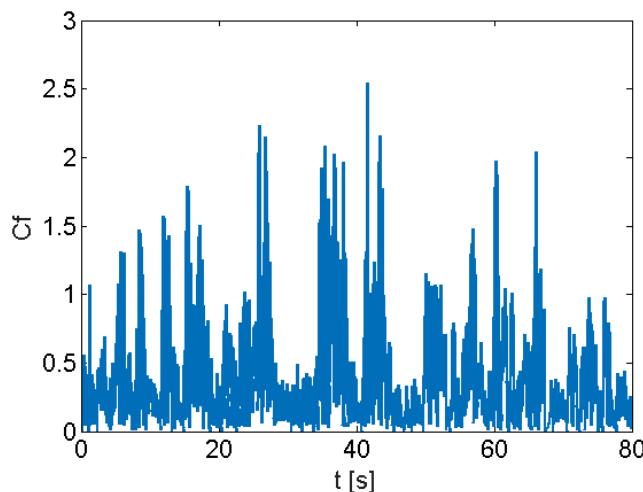


Figura 5.17: Evolução do valor da função custo no tempo obtida experimentalmente durante a operação do veículo de duas rodas.

5.4. Controladores Clássicos

Para que seja possível avaliar o desempenho dos controladores neuro-fuzzy projetados, realiza-se uma análise comparativa entre os mesmos e outras metodologias de controle, a saber:

- Controlador proporcional cujo ganho corresponde ao ganho crítico determinado empiricamente;
- Controlador PID obtido pelo método de Ziegler Nichols modificado, denominado ZNPID;
- Controlador PID obtido pelo método do lugar das raízes, RLPID, levando em consideração o modelo identificado no capítulo III;

As Figs. 5.18a e 5.18b apresentam respectivamente a ação de controle normalizada do sistema em função do tempo, e o ângulo de inclinação, ambos obtidos experimentalmente a partir da utilização do controlador proporcional.

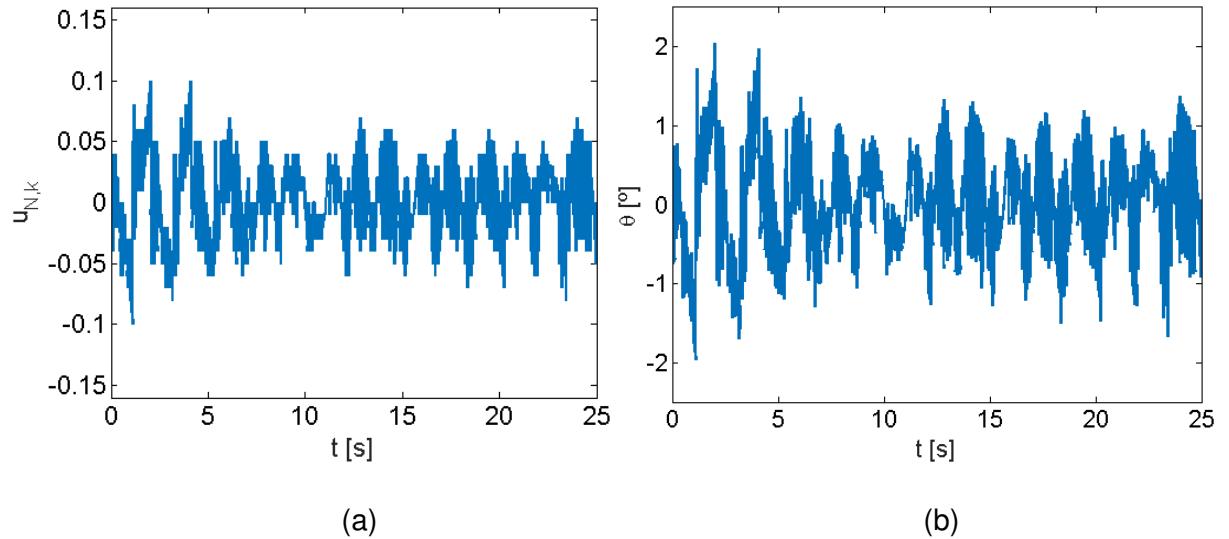


Figura 5.18: (a) Ação de controle e (b) ângulo de inclinação experimentais do veículo de duas rodas utilizando o controlador proporcional.

Da mesma forma, as Figs. 5.19a e 5.19b apresentam respectivamente a ação de controle e o ângulo de inclinação do sistema a partir da utilização do controlador ZNPID.

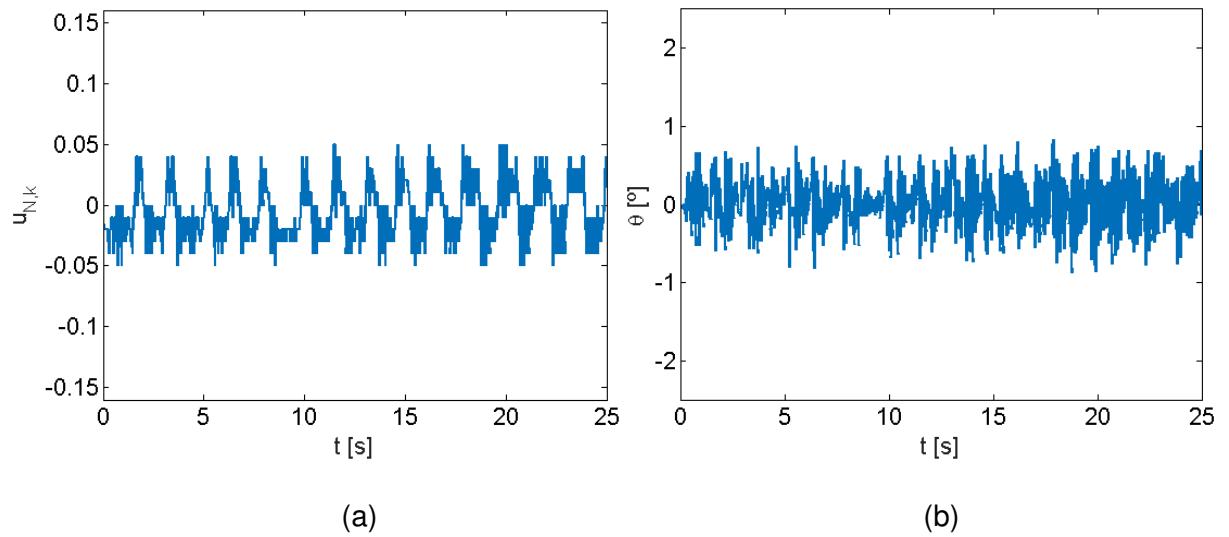


Figura 5.19: (a) Ação de controle e (b) ângulo de inclinação experimentais do veículo de duas rodas utilizando o controlador ZNPID.

Por fim, as Figs. 5.20a e 5.20b são gráficos relativos ao desempenho do controlador RLPID.

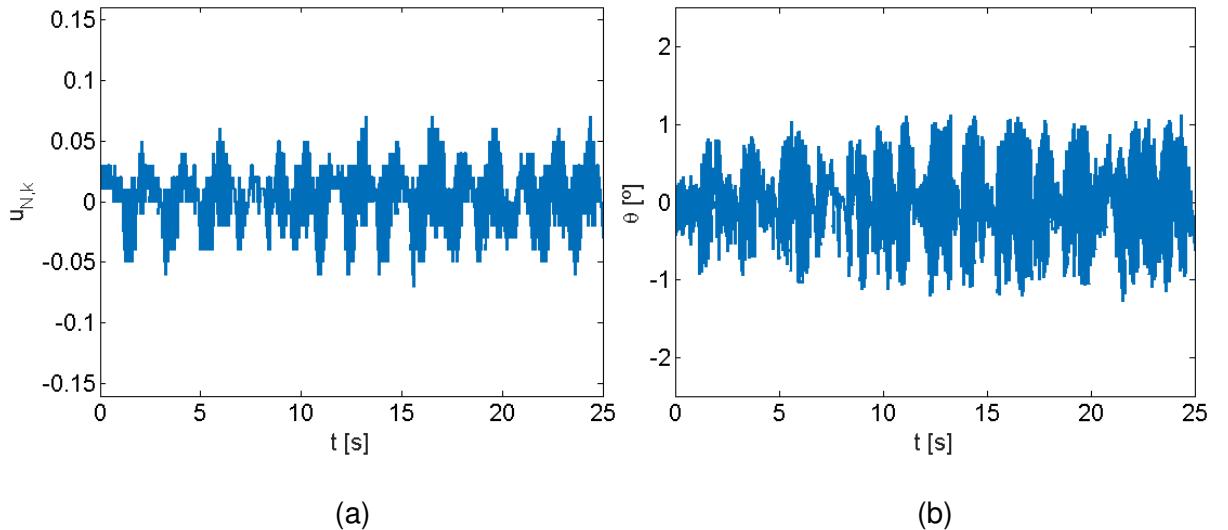


Figura 5.20: (a) Ação de controle e (b) ângulo de inclinação experimentais do veículo de duas rodas utilizando o controlador RLPID.

A partir dos resultados apresentados nas Figs. 5.18, 5.19 e 5.20, é possível verificar que o controlador ZNPID minimiza de forma mais efetiva o grau de dispersão da ação de controle e do ângulo de inclinação do veículo do que as demais metodologias de controle apresentadas nessa subseção.

Ao se comparar qualitativamente o desempenho dos controladores ZNPID, DEDMNF e DENFNF, verifica-se que embora o controlador DENFNF ainda apresente os melhores resultados, o ZNPID obtém um desempenho expressivamente superior ao DEDMNF. Esse comportamento sugere que embora o modelo dinâmico da planta tenha se ajustado satisfatoriamente aos dados experimentais extraídos do veículo, ele deixa de levar em conta algumas não-linearidades que afetam o desempenho dos controladores que são dimensionados tendo como base o modelo dinâmico, como por exemplo o RLPID e o DEDMNF.

5.5. Análise Estatística dos Resultados

Nesta subseção são analisados respectivamente, a partir dos dados extraídos em simulação, o efeito da zona morta sob a dinâmica do sistema e a capacidade do controlador neuro-fuzzy, projetado a partir do modelo dinâmico da planta, em manter o veículo oscilando em torno da posição de equilíbrio estático para qualquer valor de δ .

Em seguida, apresenta-se uma tabela comparativa dos resultados experimentais de cada metodologia de controle utilizada, e embora o controlador DENFNF tenha obtido o melhor desempenho nos experimentos, no intuito de se comparar estatisticamente os resultados do mesmo frente às demais metodologias de controle, são realizados testes de hipóteses que comparam o método DENFNF com o que obteve resultados imediatamente inferiores em cada uma das duas grandezas analisadas, sendo elas as variâncias do ângulo de inclinação e da ação de controle normalizada.

A Tab. 5.2 apresenta os valores extraídos em simulação da média e desvio padrão amostrais da ação de controle e do ângulo de inclinação em função de δ .

Tabela 5.2: Desempenho do sistema em função de δ .

$\delta [V]$	\bar{u}_N	s_{u_N}	$\bar{\theta} [^\circ]$	$s_\theta [^\circ]$
0	0,035	$\pm 0,27$	-0,059	$\pm 1,11$
0,6	0,017	$\pm 0,49$	-0,014	$\pm 1,71$
1,2	-0,041	$\pm 0,44$	0,002	$\pm 1,73$
1,8	0,013	$\pm 0,57$	0,026	$\pm 2,10$

A partir dos valores apresentados na Tab. 5.2, realizam-se testes estatísticos (Montgomery e Runger, 2002) no intuito de se analisar, com certo nível de confiança estatística, o efeito da zona morta no desempenho do veículo e a capacidade do controlador neuro-fuzzy de fazer com que o sistema siga o valor de referência pré-determinado.

Para o teste estatístico do efeito da zona morta, a partir da distribuição estatística F de Snedecor, 2 hipóteses são elaboradas:

- **Hipótese nula:** A variância da resposta do veículo é insensível à existência de zona morta nos atuadores;

- **Hipótese alternativa:** A variância da resposta do veículo é maior quando os atuadores apresentam zona morta.

A hipótese nula foi rejeitada para qualquer valor de δ analisado diferente de 0 com nível de confiança de 95%, o que indica que a zona morta de fato contribui para o aumento da amplitude de vibração do sistema em torno do valor de referência.

Já para o teste estatístico da capacidade do controlador neuro-fuzzy de fazer com que o sistema siga o valor de referência pré-determinado, a partir da distribuição estatística t de Student, 2 hipóteses são elaboradas:

- **Hipótese nula:** A média da resposta do veículo é igual ao valor de referência zero ao se utilizar o controlador neuro-fuzzy;
- **Hipótese alternativa:** A média da resposta do veículo é diferente do valor de referência zero ao se utilizar o controlador neuro-fuzzy.

Para qualquer valor de δ analisado, a hipótese nula não foi rejeitada com nível de confiança de 95%, ou seja, não há qualquer indício de que o controlador neuro-fuzzy não seja capaz de estabilizar e controlar o ângulo de inclinação do veículo de duas rodas.

A partir dos dados extraídos experimentalmente da planta, a Tab. 5.3 apresenta para cada controlador utilizado os valores obtidos para os intervalos de confiança das médias μ_θ e μ_u relativas respectivamente ao ângulo de inclinação e à ação de controle normalizada, com nível de confiança de 95,45%, bem como os valores dos desvios padrão amostrais correspondentes, s_θ e s_u .

Tabela 5.3: Desempenho dos controladores.

Controlador	μ_θ [°]	s_θ [°]	μ_u	s_u
DEDMNF	$0,00 \pm 0,88$	$\pm 0,44$	$0,01 \pm 0,03$	$\pm 0,02$
DENFNF	$-0,12 \pm 0,36$	$\pm 0,18$	$-0,01 \pm 0,03$	$\pm 0,01$
OnMoDE	$0,03 \pm 1,12$	$\pm 0,56$	$0,00 \pm 0,06$	$\pm 0,03$
OnMoDE (últimos 5 segundos)	$-0,08 \pm 0,84$	$\pm 0,42$	$0,00 \pm 0,04$	$\pm 0,02$
Proporcional	$0,04 \pm 1,18$	$\pm 0,59$	$0,00 \pm 0,06$	$\pm 0,03$
ZNPID	$0,00 \pm 0,48$	$\pm 0,24$	$-0,01 \pm 0,04$	$\pm 0,02$
RLPID	$0,00 \pm 0,89$	$\pm 0,44$	$0,00 \pm 0,04$	$\pm 0,02$

Ao se analisar os valores apresentados na Tab. 5.3 verifica-se, com nível de confiança de 95,45%, que não há motivos para rejeitar a hipótese de que todos os controladores testados tendam a manter o ângulo de inclinação e a ação de controle oscilando aproximadamente em torno do valor nulo, condições necessárias para a estabilidade estática da planta.

Para o teste de hipótese do grau de dispersão do ângulo de elevação do veículo de duas rodas a partir dos controladores ZNPID e DENFNF, duas hipóteses são elaboradas:

- **Hipótese nula:** A variância do ângulo de elevação do veículo é igual tanto com o controlador ZNPID, quanto com o controlador DENFNF.
- **Hipótese alternativa:** A variância do ângulo de elevação do veículo é maior quando se utiliza o controlador ZNPID, do que quando se utiliza o controlador DENFNF.

Levando em consideração a distribuição estatística F de Snedecor, com nível de confiança de 95%, a hipótese nula foi rejeitada, sendo assim, é válido afirmar com o mesmo nível de confiança que a utilização do controlador proposto garante menor grau dispersão do ângulo de elevação do sistema em torno de sua posição de estabilidade estática em relação aos demais controladores testados.

Para o teste de hipótese do grau de dispersão da ação de controle normalizada do veículo de duas rodas a partir dos controladores DEDMNF e DENFNF, duas hipóteses são elaboradas:

- **Hipótese nula:** A variância da ação de controle normalizada do veículo é igual tanto com o controlador DEDMNF, quanto com o controlador DENFNF.
- **Hipótese alternativa:** A variância da ação de controle normalizada do veículo é maior quando se utiliza o controlador DEDMNF, do que quando se utiliza o controlador DENFNF.

Levando em consideração novamente a distribuição estatística F de Snedecor, com nível de confiança de 95%, a hipótese nula foi rejeitada, sendo assim, é válido afirmar com o mesmo nível de confiança que a utilização do controlador proposto garante menor grau dispersão da ação de controle aplicada ao sistema em relação aos demais controladores testados.

CAPÍTULO VI

CONCLUSÕES

O trabalho apresenta a aplicação de técnicas de controle inteligente na análise e controle da estabilização estática de um veículo de duas rodas do tipo pêndulo invertido, com ênfase no desenvolvimento de algoritmos e na contribuição nessa área de estudo. Para isso, realiza-se a modelagem dinâmica e neuro-fuzzy do veículo de rodas, determinam-se as incertezas relacionadas ao processo, e por fim são realizadas simulações e implementações de controladores neuro-fuzzy que utilizam a evolução diferencial como método de busca e otimização de seus parâmetros.

Na fase de modelagem obtém-se, a partir do conjunto de dados de validação do modelo, uma taxa de ajuste aos dados experimentais de 89%, o que indica a capacidade do ANFIS tanto no ajuste de dados experimentais quanto na modelagem de sistemas dinâmicos não-lineares.

A análise de variância e o método de Monte Carlo são ferramentas capazes de calcular as incertezas relacionadas a sistemas neuro-fuzzy. A análise de variância presta-se à determinação do intervalo de confiança para cada um dos parâmetros dos consequentes do ANFIS, enquanto que o método de Monte Carlo destina-se ao estudo das incertezas relacionadas às respostas dos sensores que porventura sejam utilizados, e também ao estudo da incerteza ligada à saída do sistema neuro-fuzzy.

Verifica-se que a zona morta presente nos motores DC de um veículo de duas é uma não-linearidade que afeta a amplitude da oscilação do ângulo de inclinação do sistema, o que reduz a controlabilidade do mesmo a partir das técnicas clássicas de controle.

Conhecendo-se os valores da amplitude δ da região de zona morta dos atuadores envolvidos no processo, é possível projetar um controlador neuro-fuzzy que leve em consideração a não-linearidade em questão e seja capaz de estabilizar e controlar o ângulo de inclinação do veículo de duas rodas do tipo pêndulo invertido.

Ao comparar os valores de s_θ obtidos nos métodos OnMoDE e RLPID, verifica-se que o controlador RLPID apresenta melhores resultados durante praticamente todo o experimento. No entanto, nos últimos 5 segundos, a abordagem proposta possui um grau de dispersão do ângulo de inclinação mais baixo, o que indica a eficácia do método na adaptação dos pesos dos consequentes do controlador neuro-fuzzy de modo a torná-lo mais específico e eficiente para a aplicação proposta na medida em que se evolui o tempo.

Os resultados e análises sugerem que o método OnMoDE, devido à sua capacidade de conferir adaptabilidade a uma metodologia de controle, é uma ferramenta útil especialmente quando se deseja controlar sistemas complexos, com difícil obtenção dos parâmetros ou que estes variam durante o processo; sistemas em ambiente com alto grau de perturbações; ou quando o modelo dinâmico da planta é desconhecido.

Baseando-se nos resultados experimentais e nos testes estatísticos relativos ao ângulo de inclinação e à ação de controle do veículo de duas rodas do tipo pêndulo invertido, é possível verificar a superioridade do controlador DENFNF em relação aos demais métodos testados em garantir a estabilidade estática e a eficiência energética do sistema, sendo dessa forma o método recomendado para aplicações envolvendo o controle de sistemas dinâmicos não-lineares.

A utilização do modelo neuro-fuzzy como ferramenta para a simulação da operação do veículo e para a obtenção dos parâmetros do controlador neuro-fuzzy, devido à sua estrutura genérica não-linear, tende a permitir a absorção de maior parte das não-linearidades existentes e que podem ser mascaradas pelo modelo dinâmico linearizado da planta.

Os resultados obtidos reforçam a importância de se levar em conta a modelagem e o controle neuro-fuzzy quando se deseja controlar um sistema dinâmico em que se conhece qualitativamente seu comportamento, é possível extrair dados experimentais do mesmo, e sobretudo quando não se dispõe de um modelo matemático preciso que consiga considerar suas principais restrições e não-linearidades.

CAPÍTULO VII

TRABALHOS FUTUROS

De acordo com dados extraídos do Cenipa (Cenipa, 2016) (Centro de Investigação e Prevenção de Acidentes Aeronáuticos), órgão da Aeronáutica responsável por apurar acidentes e elaborar relatórios apontando seus principais motivos, de 2012 a 2016 houve o registro de 609 mortes devido a acidentes aéreos, e cerca de 70% dos 617 acidentes aéreos ocorridos no Brasil entre os anos 2000 e 2014, e que já tiveram investigações concluídas, apresentam apenas 3 causas relatadas, sendo elas:

- Perda de controle da aeronave - em voo ou em solo: 33% dos casos;
- Falha do motor em voo: 24% dos casos;
- Colisão em voo: 10% dos casos, dado que 90% destes são causados por aves, sobretudo nas operações de pouso ou decolagem, sendo os motores as partes mais atingidas.

Na intenção de se reduzir o número de acidentes aéreos acometidos principalmente pelas 3 causas mencionadas acima, e dessa forma buscar contribuir na garantia de maiores condições de segurança aos usuários de meios de transporte aéreos, propõe-se o desenvolvimento e aplicação de métodos de controle inteligente adaptativo que consigam auxiliar no controle da atitude de aeronaves sujeitas a operações críticas, como por exemplo em pouso de emergência em que há perda de potência no motor.

Assim, propõe-se o desenvolvimento e implementação de técnicas de controle e sintonização on-line de seus parâmetros, baseadas em métodos heurísticos e evolutivos; a

realização de simulações em ambiente virtual para validação numérica das técnicas desenvolvidas; a implementação das técnicas em um veículo de duas rodas do tipo pêndulo invertido; e por fim a execução de ensaios a partir de um aeromodelo construído pela equipe tucano de aerodesign da Universidade Federal de Uberlândia, Fig. 7.1, para validação experimental dos controladores propostos.



Figura 7.1: Aeromodelo construído pela equipe Tucano.

As técnicas de controle a serem desenvolvidas partem do conhecimento adquirido neste trabalho, e também visam suprir algumas das limitações das técnicas descritas no texto, principalmente no que se refere à adaptabilidade e convergência do método OnMoDE. Estão em fase de desenvolvimento uma estrutura matemática genética inspirada no sistema neuro-fuzzy, mas adaptada especificamente a problemas de controle, e também um método de otimização baseado no reconhecimento simulado, mas também mais específico a problemas dessa área.

Além disso, o controle adaptativo de sistemas exige o desenvolvimento de novas sistemáticas de avaliação dos resultados. Estas sistemáticas envolvem a área de *machine learning* e abrem portas para novas descobertas.

REFERÊNCIAS BIBLIOGRÁFICAS

- AQIL, M. et al. Analysis and Prediction of Flow from Local Source in a River Basin Using a Neuro-Fuzzy Modeling Tool. **Journal of Environmental Management**, Japan, v. 85, p. 215-223, nov. 2006.
- AREN CIBIA, R. V. **Avaliação da Incerteza de Medição Utilizando o Método de Monte Carlo**, Uberlândia, nov. 2016. 34 p.
- BABUŠKA, Robert; VERBRUGGEN, Henk. Neuro-fuzzy methods for nonlinear system identification. **Annual Reviews in Control**, Netherlands, v. 27, p. 73-85, jun. 2003. [https://doi.org/10.1016/S1367-5788\(03\)00009-9](https://doi.org/10.1016/S1367-5788(03)00009-9)
- BERLAND. **Illustration of linear regression on a data set**, 2007. Disponível em: <https://commons.wikimedia.org/wiki/File:LinearRegression.svg#/media/File:Linear_Regression_.svg>. Acesso em: 13 jun. 2017.
- BONAFILIA, B. et al. **Self-balancing two-wheeled robot**. p. 1-11, fev. 2014. Disponível em: <<https://pdfs.semanticscholar.org/ad7a/3b833695e8e01c1d41043c75c770276c923c.pdf>>. Acesso em: 16 set. 2015.
- BOSKOVICH, Scott M. A Two Wheeled Robot Control System. **WESCON/'95. Conference record. 'Microelectronics Communications Technology Producing Quality Products Mobile and Portable Power Emerging Technologies'**, California, p. 750-755, nov. 1995. Disponível em: <<http://ieeexplore.ieee.org/document/485495/>>. Acesso em: 04 out. 2016.
- BROWN, R.G.; HWANG, P. Y. C. **Introduction to Random Signals and Applied Kalman Filtering**. 4 ed. EUA: John Wiley & Sons, 2012. 383 p.
- BURAGOHAIN, Mrinal; MAHANTA, Chitralekha. A Novel Approach for ANFIS Modelling based on Full Factorial Design. **Applied Soft Computing**, India, v. 8, p. 609-625, jun. 2007.
- CENIPA. **Ocorrências Aeronáuticas**: Panorama Estatístico da Aviação Brasileira, Folheto do Comando da Aeronáutica, 2016.
- CHEN, Cheng-Hung; LIN, Cheng-Jian; LIN, Chin-Teng. Nonlinear System Control Using Adaptive Neural Fuzzy Networks Based on a Modified Differential Evolution. **IEEE Transactions on Systems, Man, and Cybernetics — Part C: Applications and Reviews**, Taiwan, v. 39, n. 4, p. 459-473, mai. 2008.
- CHEN, Min-You; LINKENS, D. A. A Systematic Neuro-Fuzzy Modeling Framework With Application to Material Property Prediction. **IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics**, United Kingdom, v. 31, n. 5, p. 781-790, mai. 2001.

DENAÏ, Mouloud A.; PALIS, Frank; ZEGHBIB, Abdelhafid. ANFIS Based Modelling and Control of Non-linear Systems : A tutorial. **IEEE International Conference on Systems, Man and Cybernetics**, Germany, p. 3433-3438, jul. 2004.

GODJEVAC, Jelena; STEELE, Nigel. Neuro-Fuzzy Control of a Mobile Robot. **Neurocomputing**, Switzerland, v. 28, p. 127-143, jul. 1999. [https://doi.org/10.1016/S0925-2312\(98\)00119-2](https://doi.org/10.1016/S0925-2312(98)00119-2)

GOMIDE, F. A. C. et al. **Conceitos Fundamentais da Teoria de Conjuntos Fuzzy**, Lógica Fuzzy e Aplicações, Unicamp, 2015, 39 p.

GRABOWSKI, P. Z. et al. A Simple Direct-Torque Neuro-Fuzzy Control of PWM-Inverter-Fed Induction Motor Drive. **IEEE Transactions on Industrial Electronics**, Poland, v. 47, n. 4, p. 863-870, mai. 2000.

HAN, Ming-Feng; LIN, Chin-Teng; CHANG, Jyh-Yeong. Differential Evolution With Local Information for Neuro-Fuzzy Systems Optimisation. **Knowledge-Based Systems**, Taiwan, v. 44, p. 78-89, fev. 2013.

<https://doi.org/10.1016/j.knosys.2013.01.023>

HASILOGLU, A. et al. Adaptive neuro-fuzzy modeling of transient heat transfer in circular duct air flow. **International Journal of Thermal Sciences**, Turkey, v. 43, p. 1075-1090, mai. 2004.

HAYKIN, Simon. **Redes neurais: Princípios e Prática**. 2 ed. Canada: Bookman, 2008. 899 p.

INMETRO. **A Estimativa da Incerteza de Medição pelos Métodos do ISSO GUM 95 e de Simulação de Monte Carlo**, Brasil, set. 2008, 33 p.

JAFELICE, R. S. M. **Modelagem Fuzzy para Dinâmica de Transferência de Soropositivos para HIV em Doença Plenamente Manifesta**. Campinas, SP, 2003, 169 p.

JANG, J. S. Roger. Fuzzy Controller Design without Domain Experts. **IEEE International Conference on Fuzzy Systems**, Berkeley, p. 289-296, mar. 1992a.

JANG, Jyh-Shing R. Self-Learning Fuzzy Controllers Based on Temporal Back Propagation. **IEEE Transactions on Neural Networks**, UC Berkeley, v. 3, n. 5, p. 714-723, set. 1992b. <https://doi.org/10.1109/72.159060>

JANG, Jyh-Shing Roger; SUN, Chuen-Tsai. Neuro-Fuzzy Modeling and Control. **Proceedings of the IEEE**, UC Berkeley, v. 83, n. 3, p. 378-406, mar. 1995. Disponível em: <<http://ieeexplore.ieee.org/abstract/document/364486/>>. Acesso em: 23 mar. 2016.

JUANG, Hau-Shiue; LUM, Kai-Yew. Design and Control of a Two-Wheel Self-Balancing Robot using the Arduino Microcontroller Board. **10th IEEE International Conference on Control and Automation (ICCA)**, China, p. 634-639, abr. 2013.

LANDAU, I. D. et al. **Adaptive Control: Algorithms, Analysis and Applications**, EUA: Springs, 2011, 34 p.
<https://doi.org/10.1007/978-0-85729-664-1>

MONTGOMERY, Douglas C. **Diseño y Análisis de Experimentos**: Versión Autorizada en Español. 2 ed. Arizona, EUA: Limusa Wiley, 2004. 686 p.

MÓR, F. N. et al. **Aborgadem Paralela da Evolução Diferencial em GPU**. Gramado, RS: XV Escola Regional de Alto Desempenho, 2015. 11 p.

MONTGOMERY, Douglas C; RUNGER, George C. **Applied Statistics and Probability for Engineers**. 3 ed. Arizona, EUA: John Wiley & Sons, 2002. 706 p.

NAAWI, S. W.; Ahmad, M. N.; Osman, J. H. S. Real-Time Control of a Two-Wheeled Inverted Pendulum Mobile Robot. **Proceedings of World Academy of Science, Engineering and Technology**, v. 29, p. 214-220, mai. 2008.

NAYAK, P. et al. A neuro-fuzzy computing technique for modeling hydrological time series. **Journal of Hydrology**, India, v. 291, p. 52-66, mai. 2004.

NEJADFARD, Atabak; YAZDANPANAH, M. J.; HASSANZADEH, Iraj. Friction compensation of double inverted pendulum on a cart using locally linear neuro-fuzzy model. **Neural Comput & Applic**, London, v. 22, p. 337-347, jul. 2013.
<https://doi.org/10.1007/s00521-011-0686-3>

ORLOWSKA-KOWALSKA, Teresa; SZABAT, Krzysztof. Control of the Drive System With Stiff and Elastic Couplings Using Adaptive Neuro-Fuzzy Approach. **IEEE Transactions on Industrial Electronics**, Poland, v. 54, n. 1, mai. 2008.

ORTIZ, R. J. F.; TOSCANO, E. M. M. T. **Estatística Usando Minitab**. Brasil, 2010, 77 p.

PARTAL, Turgay; KISI, Özgür. Wavelet and Neuro-Fuzzy Conjunction Model for Precipitation Forecasting. **Journal of Hydrology**, Turkey, v. 342, n. 1, p. 199-212, mai. 2007.

PENG, Jinzhu; DUBAY, Rickey. Identification and adaptive neural network control of a DC motor system with dead-zone characteristic. **ISA Transactions**, Canada, v. 50, n. 4, p. 588-598, jul. 2011. Disponível em:
<http://www.sciencedirect.com/science/article/pii/S0019057811000711>. Acesso em: 31 mar. 2016.

PEREIRA, B. L. et al. Two-Wheeled Vehicle Control Using Neural Network and Differential Evolution. **Postgraduate Symposium in Mechanical Engineering of Federal University of Uberlândia (POSMEC)**, Brazil, p. 1-7, dez. 2016. Disponível em:
<https://ssl4799.websiteseugro.com/swge5/PROCEEDINGS/PDF/POSMEC2016-0037.pdf>. Acesso em: 24 dez. 2016.

PETCHINATHAN, G. et al. Local Linear Model Tree and Neuro-fuzzy System for Modelling and Control of an Experimental pH Neutralization Process. **Brazilian Journal of Chemical**

Engineering, India, v. 31, n. 2, p. 483-495, abr./jun. 2014.
<https://doi.org/10.1590/0104-6632.20140312s00002287>

ROVERE, Emilio Lèbre La; WILLS, William. **Eficiência energética e emissões de gases de efeito estufa (gee):** Estimativa de redução das emissões de GEE até 2030, a partir de cenários de eficiência energética para veículos leves no Brasil. 1 ed. Rio de Janeiro: GREENPEACE, 2014. 39 p.

RUSU, Petru; PETRIU, Emil M. Behavior-Based Neuro-Fuzzy Controller for Mobile Robot Navigation. **IEEE Transactions on Instrumentation and Measurement**, Canada, v. 52, n. 4, p. 1335-1340, mai. 2008.

SAIFIZUL, A. A. et al. Intelligent Control for Self-erecting Inverted Pendulum Via Adaptive Neuro-fuzzy Inference System. **American Journal of Applied Sciences**, Malaysia, v. 3, n. 4, p. 1795-1802, jul. 2006.

<https://doi.org/10.3844/ajassp.2006.1795.1802>

SALERNO, Alessio; ANGELES, Jorge. A New Family of Two-Wheeled Mobile Robots: Modeling and Controllability. **IEEE Transactions on Robotics**, v. 23, n. 1, p. 169-173, fev. 2007.

SENG, Teo Lian; KHALID, Marzuki Bin; YUSOF, Rubiyah. Tuning of a Neuro-Fuzzy Controller by Genetic Algorithm. **IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics**, Malaysia, v. 29, n. 2, mai. 2008.

SHEN, W. X. et al. Adaptive Neuro-Fuzzy Modeling of Battery Residual Capacity for Electric Vehicle. **IEEE Transactions on Industrial Electronics**, Hong Kong, v. 49, n. 3, p. 677-684, mai. 2002.

SILVA, Gustavo Vitorino Monteiro Da. **Controlo não linear:** Edição Digital. 3 ed. Lisboa: Fxs, 2006. 219 p.

STORN, Rainer; PRICE, Kenneth. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. **Journal of Global Optimization**, Netherlands, v. 11, p. 341-359, nov. 1996.

SU, Kuo-Ho; CHEN, Yih-Young; SU, Shun-Feng. Design of Neuro-Fuzzy-Based Controller For Two Autonomously Driven Wheeled Robot. **Neurocomputing**, Taiwan, v. 73, p. 2478-2488, mai. 2010.

SU, M. et al. A Rule-Based Symbiotic MOdified Differential Evolution for Self-Organizing Neuro-Fuzzy Systems. **Applied Soft Computing**, Taiwan, v. 11, p. 4847-4858, jul. 2011.
<https://doi.org/10.1016/j.asoc.2011.06.015>

TAKAGI, Tomohiro; SUGENO, Michio. Fuzzy Identification of Systems and Its Applications to Modeling and Control. **IEEE Transactions on Systems, Man, and Cybernetics**, Japan, v. 15, n. 1, p. 116-132, jan./fev. 1985.

TANAKA, Kazuo; SANO, Manabu; WATANABE, Hiroyuki. Modeling and Control of Carbon Monoxide Concentration Using a Neuro-Fuzzy Technique. **IEEE Transactions on Fuzzy Systems**, Japan, v. 3, n. 3, p. 271-279, mai. 1995.

TAO, S. et al. Modelling and control PEMFC using fuzzy neural networks. **Journal of Zhejiang University SCIENCE**, China, v. 6, n. 10, p. 1084-1089, abr. 2005. <https://doi.org/10.1631/jzus.2005.A1084>

TATIKONDA, Ravi Chandra; BATTULA, Venkata Praveen; KUMAR, Vijay. Control of Inverted pendulum using Adaptive Neuro Fuzzy Inference Structure (ANFIS). **Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)**, India, p. 1348-1351, ago. 2010.

TSAI, Ching-Chih; HUANG, Hsu-Chih; LIN, Shui-Chun. Adaptive Neural Network Control of a Self-Balancing Two-Wheeled Scooter. **IEEE Transactions on Industrial Electronics**, Taiwan, v. 57, n. 4, p. 1420-1428, mai. 2008.

VALLE, M. E. **O Método de Inferência de Takagi-Sugeno para Sistemas Baseados em Regras Fuzzy**. Unicamp, 2015, 43 p.

VUOLO, J. H. **Fundamentos da Teoria de Erros**. 1 ed. USP, 1993, 250 p.

WANNIGE, C. T.; SONNADARA, D. U. J. Adaptive Neuro-Fuzzy Traffic Signal Control for Multiple Junctions. **IEEE International Conference on Industrial and Information Systems (ICIIS)**, University of Colombo, v. 4, p. 262-267, dez. 2009. <https://doi.org/10.1109/ICIINFS.2009.5429853>

XU, Jian-Xin; GUO, Zhao-Qin; LEE, Tong Heng. Design and Implementation of Integral Sliding-Mode Control on an Underactuated Two-Wheeled Mobile Robot. **IEEE Transactions on Industrial Electronics**, Japan, v. 61, n. 7, p. 3671-3681, jul. 2014. <https://doi.org/10.1109/TIE.2013.2282594>

ZADEH, L. A. Fuzzy Sets. **Information and Control**, Berkeley, v. 8, p. 338-353, mar. 1965. [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)

ZHANG, Jie; MORRIS, A. Julian. Recurrent Neuro-Fuzzy Networks for Nonlinear Process Modeling. **IEEE Transactions on Neural Networks**, United Kingdom, v. 10, n. 2, p. 313-326, mai. 2008.

APÊNDICES

A) Filtro de Kalman

O filtro de Kalman foi desenvolvido no ano de 1960 e até hoje é o método estocástico mais utilizado na filtragem de sinais. Ele se baseia em um conjunto de equações matemáticas que constitui um processo recursivo eficiente de estimação, uma vez que o erro quadrático é minimizado. Através da observação da variável denominada variável de observação, outra variável (não observável) denominada variável de estado pode ser estimada eficientemente.

Para o estudo do filtro de Kalman, utiliza-se a referência (Brown e Hwang, 2012), e alguns conceitos referentes ao tema são apresentados a seguir.

- **Variável de Estado (x):**

- Variável relacionada à dinâmica de um sistema, baseada em um modelo matemático. Relacionada também à predição de dados futuros.
- O valor de x na iteração posterior é o valor na iteração atual com uma correção ϕ_k e corrompido por um sinal w_k de natureza aleatória (sequência branca) e com variância Q , conforme a Eq. (A.1):

$$x_{k+1} = \phi_k x_k + w_k \quad (\text{A.1})$$

- **Variável de Observação (z):**

- Variável relacionada à aquisição de dados de um sensor;
- O valor obtido pelo sensor no tempo k é o valor da variável de estado com uma correção H_k e somado com ruído branco v_k com variância R , de acordo com a Eq. (A.2):

$$z_k = H_k x_k + v_k \quad (\text{A.2})$$

- **Propagação:**

- É a predição do valor das variáveis em iterações seguintes baseada nos valores atuais destas e no modelo matemático do sistema.

- **Atualização:**

- É a correção da predição utilizando os novos dados coletados pelos sensores.

- **Variância do Erro (P):**

- O erro de estimação do filtro de Kalman em uma iteração k é definido como sendo a diferença entre o valor da variável de estado obtido pelo modelo (x_k) e o valor estimado pelo filtro (\hat{x}_k).
- Como o valor do erro flutua em torno de zero durante a execução do filtro (e portanto, tem média nula), a variância do erro é equivalente ao seu erro quadrático. Dessa forma, para se obter as equações do filtro de Kalman, é calculada a derivada de P em relação a K (ganho do filtro) e igualada a zero, de modo a minimizar de forma ótima o valor do erro quadrático.

No tópico a seguir são indicadas as matrizes e vetores que expandem os conceitos acima para problemas em que há mais de uma variável de estado. Além disso, são apresentadas as equações do filtro de Kalman.

A.1 Expansão do Método para Problemas com Mais de Uma Variável de Estado e Equações do Filtro de Kalman

Seguem as matrizes e vetores utilizados nas equações do filtro de Kalman:

- x_k ($nx1$): Vetor de estados do processo no tempo t_k ;
- ϕ_k (nxn): Matriz de transição de estados, relacionando x_k com x_{k+1} ;
- w_k ($nx1$): Vetor cujos elementos são sequências brancas e com matriz de covariância conhecida; Resposta em t_{k+1} devido à presença de ruído branco na entrada do sistema no intervalo (t_k, t_{k+1}) ;
- z_k ($mx1$): Vetor de observações no tempo t_k ;
- H_k (mxn): Matriz que relaciona x_k com z_k ;
- v_k ($mx1$): Vetor cujos elementos são sequências brancas e com matriz de covariância conhecida, sendo que cada elemento está relacionado a uma das variáveis de observação.

Para obter as equações do filtro, adota-se que a correlação entre v_k e w_k é nula;

- P_k (nxn): Matriz de covariância do erro;
- Q_k (nxn): Matriz de covariância de w_k , $Q_k = E[w_k, w_k^T]$;
- R_k ($m \times m$): Matriz de covariância de v_k , $R_k = E[v_k, v_k^T]$;

- K_k ($n \times m$): Ganho do filtro de Kalman;
- I ($n \times n$): Matriz Identidade.

A seguir são apresentadas as equações do filtro de Kalman. Os valores a priori (-) são aqueles que ainda não passaram pelas equações de atualização de sua iteração correspondente, já os valores a posteriori (+) são aqueles que já passaram por esse processo:

- **Equações de Atualização:**

- Ganho do Filtro de Kalman:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (\text{A.3})$$

- Atualização do Estado:

$$\hat{x}_k^+ = \hat{x}_k^- + K_k [z_k - H_k \hat{x}_k^-] \quad (\text{A.4})$$

- Atualização da Matriz de Covariância do Erro:

$$P_k^+ = (I - K_k H_k) P_k^- \quad (\text{A.5})$$

- **Equações de Propagação:**

- Propagação do Estado:

$$\hat{x}_{k+1}^- = \phi_k \hat{x}_k^+ \quad (\text{A.6})$$

- Propagação da Matriz de Covariância do Erro:

$$P_{k+1}^- = \phi_k P_k^+ \phi_k^T + Q_k \quad (\text{A.7})$$

Após terem sido apresentadas as equações do filtro de Kalman, no próximo tópico é abordado como se determinar os parâmetros deste.

A.2 Determinação dos Parâmetros do Filtro de Kalman

Durante o projeto do filtro de Kalman, é desejável ter a representação do processo escrita na forma vetorial. A partir do processo em estudo, é possível reescrevê-lo da forma:

$$\dot{x} = Fx + Gu \quad (\text{A.8})$$

Em que x é o vetor de estados do processo, \dot{x} é sua derivada e u é o vetor de perturbação do sistema, que devido a critérios matemáticos do filtro de Kalman, corresponde a uma entrada de ruído branco unitário.

Alguns dos parâmetros do filtro são determinados de forma mais direta, como o valor da matriz H_k , que relaciona as variáveis de observação e os elementos correspondentes do

vetor de estados x ; e os valores dos elementos da matriz R_k , que quando é assumido que os dados provenientes de cada um dos sensores utilizados não são correlacionados, são determinados pelo cálculo da variância dos dados obtidos por cada um destes.

Por outro lado, o cálculo das matrizes ϕ_k e Q_k depende do modelo do sistema, e tornam-se de difícil obtenção quando os elementos das matrizes F e G não são constantes.

A seguir são apresentados o método analítico e métodos numéricos (relativamente simples e eficazes, mas restritos ao caso de os elementos de F e G serem constantes) para se obter esses parâmetros:

A.2.1 Método Analítico

- **Obtenção de ϕ_k :**

É possível obter o valor de ϕ_k através da equação (A.9), sendo Δt a taxa de amostragem do sistema :

$$\phi_k = \{\mathcal{L}^{-1}[(sI - F)^{-1}]\}_{t=\Delta t} \quad (\text{A.9})$$

- **Obtenção de Q_k :**

Baseando-se no modelo matemático do sistema – Eq. (A.8), a Eq. (A.1) também pode ser escrita na forma:

$$x(t_{k+1}) = \phi(t_{k+1}, t_k)x(t_k) + \int_{t_k}^{t_{k+1}} \phi(t_{k+1}, \tau)G(\tau)u(\tau)d\tau \quad (\text{A.10})$$

Desse modo, conforme pode ser verificado abaixo, é possível obter a matriz Q_k :

$$Q_k = E[w_k, w_k^T]$$

$$Q_k = E \left\{ \left[\int_{t_k}^{t_{k+1}} \phi(t_{k+1}, \xi)G(\xi)u(\xi)d\xi \right] \left[\int_{t_k}^{t_{k+1}} \phi(t_{k+1}, \eta)G(\eta)u(\eta)d\eta \right]^T \right\}$$

$$Q_k = \int_{t_k}^{t_{k+1}} \int_{t_k}^{t_{k+1}} \phi(t_{k+1}, \xi)G(\xi)E[u(\xi)u^T(\eta)]G^T(\eta)\phi^T(t_{k+1}, \eta)d\xi d\eta \quad (\text{A.11})$$

A.2.2 Método Numérico

- **Obtenção de ϕ_k :**

Para se obter o valor de ϕ_k , pode-se partir da Eq. (A.8) e impor $u = 0$ a fim de que a equação descreva a dinâmica natural do sistema. Assim, temos:

$$\dot{x} = Fx \quad (A.12)$$

Verifica-se que a resolução desta equação pode ser escrita da forma:

$$\frac{x(t_2)}{x(t_1)} = \phi(t_2, t_1) = e^{F(t_2 - t_1)} \quad (A.13)$$

A partir da Eq. (A.13), pode-se concluir que:

$$\phi_k = \frac{x_{k+1}}{x_k} = e^{F\Delta t} \quad (A.14)$$

A exponencial matricial que aparece na equação anterior pode ser resolvida pelo comando do MATLAB `expm(FΔt)`, que a expande em série – Eq. (A.15) - e a trunca.

$$e^{F\Delta t} = I + F\Delta t + \frac{(F\Delta t)^2}{2!} + \dots \quad (A.15)$$

- **Obtenção de Q_k :**

- Primeiramente, forme uma matriz $2nx2n$ que se chama A (n é o número de elementos do vetor de estados x).

$$A = \begin{bmatrix} -F & GG^T \\ 0 & F^T \end{bmatrix} \Delta t \quad (A.16)$$

- Faça e^A e chame de B .

$$B = \begin{bmatrix} \dots & \phi_k^{-1}Q_k \\ 0 & \phi_k^T \end{bmatrix} \quad (A.17)$$

- Faça a transposição da parte inferior direita para verificar o valor de ϕ_k .
- Finalmente, Q_k é obtido através da parte superior direita de B , como segue:

$$Q_k = \phi_k x \quad (\text{Parte superior direita de } B) \quad (A.18)$$

B) Estudo e Determinação do Tempo de Amostragem Adotado para o Controle do Veículo de Duas Rodas

O tempo de amostragem T_s adotado para um sistema de controle exerce fundamental influência sobre a resposta da planta. Em alguns casos, quando não se realiza um estudo para a verificação do valor adequado de T_s , o teorema da amostragem de nyquist (a frequência de amostragem f_s deve ser no mínimo 2 vezes superior à maior frequência de operação f_m da planta) pode não estar sendo atendido. Em termos práticos, é desejável que f_s seja no mínimo 10 vezes superior a f_m , frequências as quais podem ser determinadas a partir do método apresentado a seguir:

- Inicialmente, realiza-se a coleta de dados da resposta da planta com a maior frequência de amostragem possível, de forma a garantir que $f_s \gg f_m$. A Fig. A1 apresenta o ângulo de inclinação do veículo, extraído com tempo de amostragem $T_s = 0,002$ s.

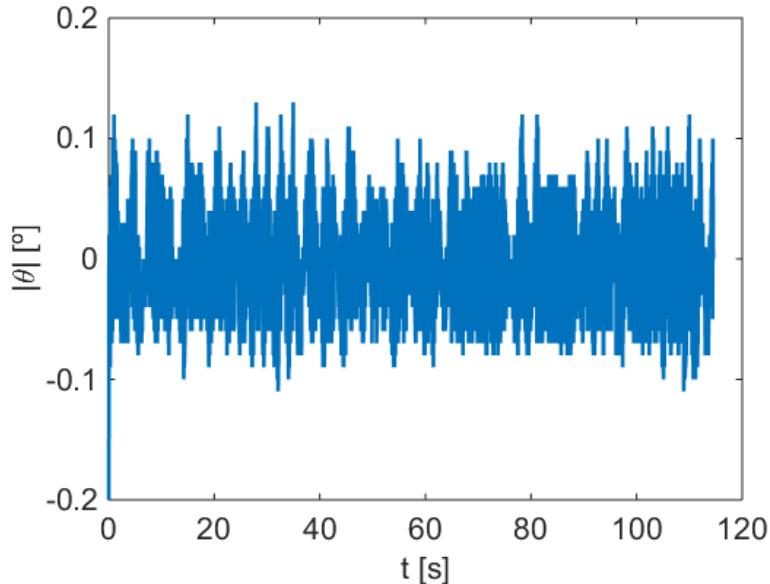


Figura A.1: Ângulo de inclinação extraído com $T_s = 0,002$ s.

- Em sequência, aplica-se a transformada de Fourier sobre o sinal para que se possa analisar o sistema no domínio da frequência. A Fig. A2 é o gráfico da amplitude do ângulo de inclinação do veículo em função da frequência.

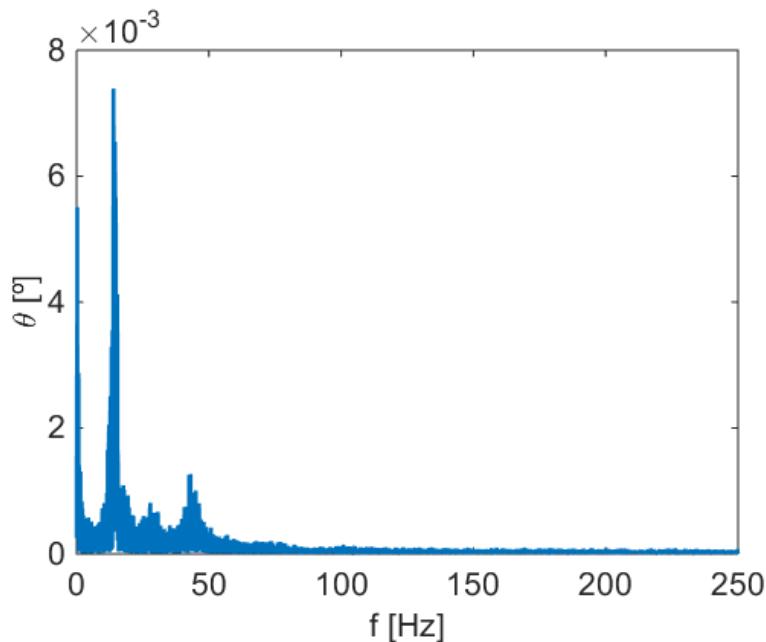


Figura A.2: Amplitude do ângulo de inclinação em função da frequência.

- Através da Fig. A2, é possível extrair que a maior frequência de operação do veículo f_m é de 14 Hz. Ressalta-se que os valores de frequência superiores a 14 Hz são relativos aos ruídos adicionados devido à IMU e à operação do PMW, e por apresentarem amplitudes expressivamente inferiores às observadas até a frequência de 14 Hz, podem ser desprezadas. Assim, é desejável que a frequência de amostragem f_s seja superior a 140 Hz. No intuito de satisfazer a recomendação prática para a frequência de amostragem, mas sem exceder a capacidade de processamento do microcontrolador utilizado, opta-se pela frequência de amostragem $f_s = 200 \text{ Hz}$, equivalente ao tempo de amostragem $T_s = 0,005 \text{ s}$.

C) Identificação do Modelo Dinâmico - MATLAB

- Programa Principal

```

clc
clear all
close all
%%%%%%%%%%%%% dados de treinamento %%%%%%
global u
global ye
dt=0.002;
nd=5000;
dados=load('teste_ident2.txt');
dados1=dados(9000:14000,:); %12000:17000
dados2=dados(7000:12000,:); %7000:12000
%%%%%%%%%%%%% WORKING DATA %%%%%%
u=dados1(:,2);
ye=dados1(:,1);
prop=treinamentoMODELO;
%%%%%%%%%%%%% VALIDATION DATA%%%%%%
u=dados2(:,2);
ye=dados2(:,1);
%%%%%%%%%%%%% %%%%%%
p=11;
N=length(u);
yb=mean(ye);
ycalc=zeros(length(ye),1);
[valor, ycalc]= fobjMODELO(prop');
ycalc=ycalc';

```

```
SQres=sum((ye-ycalc).^2);
SQreg=sum((ycalc-yb).^2);
SQtot=SQres+SQreg;
Porcentagem_ExPLICAVEL=SQreg/SQtot*R^2
MQreg=SQreg/(p-1);
MQres=SQres/(N-p);
Fcalc=MQreg/MQres
Ftab=finv(0.95,p-1,N-p)
prop
tempo=0:0.002:0.002*(nd);
figure
plot(tempo,ycalc,'b')
hold on
plot(tempo,ye,'r')
hold off
xlabel('t [s]', 'FontSize',16)
ylabel('Pitch [Degree]', 'FontSize',16)
set(gca, 'FontSize',16)
grid on
fid = fopen('OUTPUT.txt', 'wt');
fprintf(fid, '%f\n', Porcentagem_ExPLICAVEL);
fprintf(fid, '%f\n', prop);
fclose(fid);
```

- **Evolução Diferencial**

```
function var= treinamentoMODELO
global u
global ye
nd=5000;
nmaxgeracoes=100;
nind=30;
nvar=11;
pcruz=0.95;%Probabilidade de ocorrencia de soma
pvariacao=0;
F=0.4;
ind=zeros(nind,nvar); %individuos
aind=ind; %auxiliar
find=zeros(nind,1); %valor da fobj
afind=find;
var=zeros(nvar,1);
linf(1)=0.075;
lsup(1)=0.078;
linf(2)=1.304;
lsup(2)=1.304;
linf(3)=45*10^(-4);
lsup(3)=55*10^(-4);
linf(4)=0.030;
lsup(4)=0.032;
linf(5)=0.031;
lsup(5)=0.033;
```

```
linf(6)=0.25*10^(-4);
lsup(6)=0.27*10^(-4);
linf(7)=0.001;
lsup(7)=0.005;
linf(8)=0.1;
lsup(8)=0.7;
linf(9)=0.1; %tem q ser menor q o 8
lsup(9)=0.5;
linf(10)=2;
lsup(10)=3;
linf(11)=0.03;
lsup(11)=0.03;
%POPULACAO INICIAL
for i=1:nind
    for j=1:nvar
        ind(i,j)=rand;
    end
end
aind=ind;
%AVALIAÇÃO DOS individuos
for i=1:nind
    for j=1:nvar
        var(j)=(lsup(j)-linf(j))*ind(i,j)+linf(j);
    end
    find(i)=fobjMODELO(var);
end
```

```
afind=find;
contadorgeracoes=0;
flag=1;
flag1=0;
flag2=0;
c1=0;
c2=0;
auxf=inf;
%INICIO DA ED
while (flag)
    contadorgeracoes=contadorgeracoes+1
    vetcontador(contadorgeracoes)=contadorgeracoes;
    for i=1:nind
        aux=rand;
        if aux<=pcruz
            casal=randperm(nind);
            ind2=casal(2);
            ind3=casal(3);
            new=zeros(1,nvar);
```

```
for j=1:nvar
    new(j)=ind(i,j)+F*(ind(ind2,j)-ind(ind3,j));
        if new(j)<0
            new(j)=0;
        end
        if new(j)>1
            new(j)=1;
        end
    end
    for j=1:nvar
        var(j)=(lsup(j)-linf(j))*new(j)+linf(j);
    end
    fnew=fobjMODEL0(var);
    if fnew>find(i)
        afind(i)=fnew;
        aind(i,:)=new;
    end
```

```
if fnew<=find(i)
    aux=rand;
    if aux<pvariacao
        afind(i)=fnew;
        aind(i,:)=new;
    end
end
end
end
find=afind;
ind=aind;
if (contadorgeracoes>=nmaxgeracoes)
    flag=0;
end
if (abs (max (find)-auxf)<0.00001)
    c1=c1+1;
    if (c1>=100)
        flag1=1;
    end
else
    c1=0;
    flag1=0;
end
```

```
if (abs (max (find) -auxf) /max (abs (max (find)) ,10^(-10))<0.001)

    c2=c2+1;

    if  (c2>=100)

        flag2=1;

    end

else

    c2=0;

    flag2=0;

end

if flag1==1 && flag2==1

    flag=0;

end

auxf=max (find);

vetmin (contadorgeracoes)=-1*auxf;

end

find;

[erro_min ind_adapt]=max (find)

erro_min=-1*erro_min;

var=(lsup-linf).*ind(ind_adapt,:)+linf;

[valor,y]= fobjMODELO(var');

valor
```

```
contadorgeracoes

tempo=0:0.002:0.002*(nd);

figure
plot(tempo,y)
hold on
plot(tempo,ye)
hold off
xlabel('t [s]', 'FontSize', 16)
ylabel('Pitch [Degree]', 'FontSize', 16)
set(gca, 'FontSize', 16)
grid on

figure
plot(vetcontador,vetmin, 'LineWidth', 3)
xlabel('Iteração', 'FontSize', 16)
ylabel('EM^2', 'FontSize', 16)
set(gca, 'FontSize', 16)
grid on

end
```

- **Função Objetivo**

```

function [valor,y]= fobjMODELO(var)

tmax=length(u); %numero de dados;
y=zeros(1,tmax);
y(1)=ye(1); %%% SAIDA - TENSAO
y(2)=ye(2);
output=zeros(4,1);
erro=0;

[G,H]=ParModeloR(var);
umin=var(11);

for i=3:tmax-1

    output(3)=ye(i-1);

    if abs(u(i))<umin

        uef=0;

    else

        uef=u(i);

    end

    output=RespModelo(output,uef,G,H);
    y(i)=output(3);
    erro=erro+(y(i)-ye(i))^2;

end

erro=erro/tmax;
valor=-erro;
end

```

- **Resposta da Planta**

```
function new_output=RespModelo(old_output,u,G,H) %%%
entrada e saída: vetores nx1
u=6*u;
u=[0;u;0;u];
new_output=G*old_output+H*u;
end
```

D) Identificação do Modelo Neuro-Fuzzy - MATLAB

- Programa Principal

```

clc

clear all

close all

%%%%% dados de treinamento %%%%%

global u

global ye

global deltaye

global yemaxs

global yemax

global deltayemax

dados=load('Resultados_IDENT.txt');

dados1=dados(1500:3500,:); %1000:3000 %ok

dados2=dados(8960:10960,:); %2000:4000

dt=0.005;

nd=2000;

%%%%% WORKING DATA %%%%%

u=dados1(:,3);

ye=dados1(:,2);

deltaye=zeros(1,length(ye));

deltaye=ye-[0; ye(1:length(ye)-1)]; %%ENTRE 2 E -2

yemaxs=2;

yemax=2;

deltayemax=0.4;

```

```
prop=treinamentoRNA;

%%%%%%%%%%%%% VALIDATION DATA %%%%%%%

u=dados2 (:, 3);
ye=dados2 (:, 2);

figure
plot(t, 6*u, 'LineWidth', 2)
xlabel('t [s]', 'FontSize', 16)
ylabel('u [V]', 'FontSize', 16)
set(gca, 'FontSize', 16)

figure
plot(t, ye, 'LineWidth', 2)
xlabel('t [s]', 'FontSize', 16)
ylabel('\theta [°]', 'FontSize', 16)
set(gca, 'FontSize', 16)

deltaye=zeros(1,length(ye));
deltaye=ye-[0; ye(1:length(ye)-1)]; %%ENTRE 2 E -2
deltaye=ye-[0; ye(1:length(ye)-1)]; %%ENTRE 2 E -2
p=12-6-2;

N=length(u);

yb=mean(ye);
ycalc=zeros(length(ye),1);
[valor, ycalc, X]= fobjRNA(prop');
ycalc=ycalc';
```

```
SQres=sum((ye-ycalc).^2);
SQreg=sum((ycalc-yb).^2);
SQtot=SQres+SQreg;
Porcentagem_Exlicavel=SQreg/SQtot*R^2
MQreg=SQreg/(p-1);
MQres=SQres/(N-p);
Fcalc=MQreg/MQres
Ftab=finv(0.95,p-1,N-p)
%H0: A resposta não é sensível aos parâmetros
%H1: Pelo menos um coeficiente é diferente de zero
%Fcalc >Ftab: rejeita H0
%Fcalc <Ftab: Não rejeita H0
%%%%%CALCULO DO INTERVALO DE CONFIANÇA
%para 95 % de confiança
X=X(50:end,:);
Se=SQres/(N-p);
Sa=Se*diag(inv(X'*X));
t=tinv(0.975,N-p);
prop=prop';
IC(:,1)=prop-t*sqrt(Sa);
IC(:,2)=prop+t*sqrt(Sa);
```

```
prop

IC

tempo=0:0.005:0.005*(nd) ;

length(tempo)

length(ycalc)

length(ye)

figure

plot(tempo, ycalc, 'b', 'LineWidth', 2)

hold on

plot(tempo, ye, 'r', 'LineWidth', 2)

hold off

xlabel('t [s]', 'FontSize', 16)

ylabel('Pitch [Degree]', 'FontSize', 16)

set(gca, 'FontSize', 16)

fid = fopen('OUTPUT.txt', 'wt');

fprintf(fid, '%f\n', Porcentagem_ExPLICAVEL);

fprintf(fid, '%f\n', yemaxs);

fprintf(fid, '%f\n', yemax);

fprintf(fid, '%f\n', deltayemax);

fprintf(fid, '%f\n', prop);

fprintf(fid, '%f\n', sqrt(Sa));

fclose(fid);
```

- **Evolução Diferencial**

```
%Programa análogo ao treinamento MODELO do apêndice B

function var= treinamentoRNA

    global u
    global ye
    global deltaye
    global yemax
    global deltayemax

    (....)

    find(i)=fobjRNA(var);

    (....)

    figure
    plot(vetcontador,vetmin,'LineWidth',3)
    xlabel('Iteration','FontSize',16)
    ylabel('MSE','FontSize',16)
    set(gca,'FontSize',16)
end
```

- **Função Objetivo**

```
function [valor,y,X]= fobjRNA(var)

global u
global ye
global deltaye
global yemaxs
global yemax
global deltayemax

tmax=length(u); %numero de dados;
y=zeros(1,tmax);
X=zeros(tmax-2,12);
y(1)=ye(1); %% SAIDA - TENSÃO
y(2)=ye(2);
erro=0;
for i=3:tmax-1

    pitch=ye(i-1)/yemax;
    if pitch>1
        pitch=1;
    end
    if pitch<-1
        pitch=-1;
    end
    deltapest=deltaye(i-1)/deltayemax;
```

```

if deltapitch>1
    deltapitch=1;
end

if deltapitch<-1
    deltapitch=-1;
end

input=[u(i-1) pitch deltapitch];
[y(i),u3]=plantaneurofuzzy(input,var');
ud=sum(u3);
u3=yemaxs*u3/ud;

X(i-2,:)=[u3(1)*u(i-1)           u3(1)*pitch
u3(1)*deltapitch   u3(1)           u3(2)*u(i-1)           u3(2)*pitch
u3(2)*deltapitch   u3(2)           u3(3)*u(i-1)           u3(3)*pitch
u3(3)*deltapitch   u3(3)           ];

y(i)=y(i)*yemaxs;
erro=erro+(y(i)-ye(i))^2;

end
erro=erro/tmax;
valor=-erro;

%FIM CALCULO FUNCAO OBJETIVO
end

```

- **Modelo Neuro-Fuzzy**

```

function
[output,u3]=plantaneurofuzzy(input,prop)%%%prop é
matriz linha!!!
prop=[prop(1:4);prop(5:8);prop(9:12)];
u1=zeros(3,3);
u2=zeros(3,3); %% 3 entradas com 3 subconjuntos fuzzy
cada
m=[-1 0 1; -1 0 1; -1 0 1]; %%funcao de pert
gaussiana
s=[0.5 0.2 0.5; 0.4 0.4 0.4; 0.4 0.4 0.4];
s2=s.^2;
u3=zeros(3,1); %% Número de Regras Fuzzy=3
u4=zeros(3,1);
%output=0;
%%%primeira etapa: entrada de dados
for i=1:3
    u1(i)=input(i);
end
%%%segunda etapa: obtenção dos graus de pertinência
for i=1:3 %% VARIÁVEL
    for j=1:3 %% SUBCONJUNTO
        u2(i,j)=exp(-(u1(i)-m(i,j))^2/s2(i,j));
    end
end

```

```
%%%terceira etapa: operador "e"  
%u2 (variavel,subconjunto)  
u3(1)=max([u2(2,1)*u2(3,1) u2(1,2)*u2(2,1)]);  
u3(2)=max([u2(1,2)*u2(2,2)*u2(3,2)  
u2(1,3)*u2(2,3)*u2(3,2) u2(1,1)*u2(2,1)*u2(3,2)]);  
u3(3)=max([u2(2,3)*u2(3,3) u2(1,2)*u2(2,3)]);  
%%% quarta etapa: operador "e"  
aux=0;  
for i=1:3  
    aux=aux+1;  
    plano=prop(aux,:);  
    u4(i)=u3(i)*(plano(1)*input(1)+plano(2)*input(2)+plan  
o(3)*input(3)+plano(4));  
end  
%%% quinta etapa: saída do sistema neurofuzzy  
output=sum(u4)/sum(u3); %% pitch  
if output>1  
    output=1;  
end  
if output<-1  
    output=-1;  
end  
end
```

E) Determinação da Incerteza Associada à Resposta da MPU6050 – MATLAB

```

%%% CÁLCULO DA INCERTEZA UTILIZANDO PAQUÍMETRO %%%

%%% Fontes de incerteza: variab, resolucao,
calibracao, profundidade

clc

clear all

close all

valores=load('DATA.txt');

valores=valores(350:length(valores));

valores=valores(randperm(length(valores)));

n=1;

valores=valores(1:length(valores));

niter=1000000;

%%%%%%%%%%%%% GEOMETRIA E RETA %%%%%%%%%%%%%

H=75;

Ucalib=0.0015;

Ucalib=atand(Ucalib/H);

CORR=-(2.6+((100-H)*(-2.6+0.8)/(100-50)));

CORR=CORR/1000;

CORR=atand(CORR/H);

%%%%%%%%%%%%% DETERMINAÇÃO DO ERRO A PARTIR DA POSIÇÃO DE
REFERÊNCIA (aprox 0 graus) %%%%%%%%%%%%%

%%%%% POSICAO A PARTIR DA QUAL O PENDULO VAI OSCILAR

%%% Variável de saída: Erro E

%%% E=Xb-VC

```

```
% 1) DETERMINAÇÃO DA INCERTEZA PADRÃO ASSOCIADA A XB
media=mean(valores);
s=std(valores);

%1.1) VARIÁVEIS DE ENTRADA
%%% xb=f(xb,RES,CAL)

% 1.1.1) VARIABILIDADE DAS LEITURAS
uvar=s/sqrt(1); %% Distribuição normal
uv=media+uvar*randn(1,niter);

%1.1.2) RESOLUÇÃO
res=0.01;
erro=res/2;
ures=erro*2*rand(1,niter)-(erro);

%%% FORMA CORRETA DE RETANGULAR!!!!
% 1.1.3) CALIBRACAO
kv=2;
Uc=5*res;      %VUOLO
uc=(Uc/kv)*randn(1,niter);

%1.2) SAÍDA
uxb=uv+ures+uc;

%2) DETERMINAÇÃO DA INCERTEZA PADRÃO ASSOCIADA A XB
%2.1) VARIÁVEIS DE ENTRADA
%%% VC=f(CAL,CORR)
```

```
%%%%%%%%%%%%% 2.1.1) CALIBRACAO %%%%%%%%%%%%%%
kcalib=2;
ucalib=(Ucalib/kcalib)*randn(1,niter);
%%%%%%%%%%%% 2.1.2) CORREÇÃO %%%%%%%%%%%%%%
CORR=CORR*ones(1,niter);
%%%%%%%%%%%% 2.2) SAÍDA %%%%%%%%%%%%%%
uvc=ucalib+CORR;
%%%%%%%%%%%%% RESULTADO %%%%%%%%%%%%%%
%%% E=Xb-VC
Erro=media-mean(CORR)
resultado=uxb-uvc;
u=std(resultado);
k=2;
U=k*u
%%%%%%%%%%%%% DETERMINACAO DA CONTRIBUIÇÃO %%%%%%%%%%%%%%
uv=std(uv);
ures=std(ures);
uc=std(uc);
ucalib=std(ucalib);
contruv=(uv^2/u^2)*100
contrures=(ures^2/u^2)*100
contruc=(uc^2/u^2)*100
contrucalib=(ucalib^2/u^2)*100
```

```
testecontr=contruv+contrures+contruc+contrucalib  
%%%%%%%%%%%%% Construcao do Histograma %%%%%%  
A=max(resultado)-min(resultado);  
nk=floor(sqrt(niter));  
h=A/nk;  
intervalo=min(resultado);  
acc=min(resultado);  
for i=1:nk+1  
    acc=acc+h;  
    intervalo=[intervalo acc];  
end  
media=[];  
for i=1:length(intervalo)-1  
    media=[media (intervalo(i+1)+intervalo(i))/2];  
end
```

```
for i=1:length(intervalo)-1
    cont=0;
    for j=1:niter
        if resultado(j)>=intervalo(i) &&
        resultado(j)<intervalo(i+1)
            cont=cont+1;
        end
    end
    freq(i)=cont;
end
kurt=kurtosis(resultado)
sk=skewness(resultado)
bar(media,freq);
```

F) Determinação da Incerteza Associada à Resposta do Modelo Neuro-Fuzzy – MATLAB

```

clc
clear all
close all

%%%%%%%%%%%%% dados de treinamento %%%%%%
global yer
global deltayer
global u
global ye
global deltaye
global yefk
global deltayefk
global yemaxs
global yemax
global deltayemax

%%%%%%%%%%%%% DEFINIÇÕES %%%%%%
dt=0.005;

dados=load('Resultados_IDENT.txt');      %%      DADOS      DE
ENTRADA E SAÍDA

dados=dados(4000:6000,:);
u=dados(:,3);
u=u(1:2001);
yer=dados(:,2); %% ENTRE 4 E -4
yer=yer(1:2001);

```

```

deltayer=zeros(length(yer),1);

deltayer=yer-[0; yer(1:length(yer)-1)]; %ENTRE 2 E -2
dados=load('yefk.txt'); %% DADOS DE ENTRADA E SAÍDA
yefk=dados(:,1); %% ENTRE 4 E -4
deltayefk=zeros(length(yefk),1);
deltayefk=yefk-[0; yefk(1:length(yefk)-1)];%ENTRE2E-2
dados=load('OUTPUT.txt'); %% DADOS DE INCERTEZA
Porcentagem_Exlicavel=dados(1);
yemaxs=dados(2);
yemax=dados(3);
deltayemax=dados(4);
prop=zeros(12,1);
prop=dados(5:5+length(prop)-1);
ui=zeros(12+2,1);
ui=dados(5+length(prop):5+length(prop)+length(ui)-3);
ui(end+1)=0.0479;
ui(end+1)=0.0479*sqrt(2);
for i=1:length(ui)-2
    if prop(i)==0
        ui(i)=0;
    end
end
%
```

```
niter=10000;
saida=zeros(10,niter);
for i=1:niter
    i
    ye=ui(end-1)*randn(length(yefk),1);
    ye=yefk+ye;
    deltaye=ui(end)*randn(length(deltayefk),1);
    deltaye=deltayefk+deltaye;
    propaux=ui(1:end-2).*randn(length(ui)-2,1);
    propaux=propaux+prop;
    [valor, ycalc, X]= fobjRNAr (propaux);
    for j=1:10
        saida(j,i)=ycalc(floor(j*0.1*length(ycalc)));
    end
end
for j=1:10
    ui_parcial(j)=std(saida(j,:));
end
[utotal j]=max(ui_parcial);
resultado=saida(j,:);
media_resultado=mean(resultado);
utotal
U=2*utotal
```

```
%%%%%% Construcao do Histograma %%%%%%
A=max(resultado)-min(resultado);
nk=floor(sqrt(niter));
h=A/nk;
intervalo=min(resultado);
acc=min(resultado);
for i=1:nk+1
    acc=acc+h;
    intervalo=[intervalo acc];
end
media=[];
for i=1:length(intervalo)-1
    media=[media (intervalo(i+1)+intervalo(i))/2];
end
for i=1:length(intervalo)-1
    cont=0;
    for j=1:niter
        if resultado(j)>=intervalo(i) && resultado(j)<intervalo(i+1)
            cont=cont+1;
        end
    end
    freq(i)=cont;
end
```

```
kurt=kurtosis(resultado)
sk=skewness(resultado)
figure
bar(media,freq);
%%%%%%%%%%%%% GRAFICO %%%%%%
ye=yer;
deltaye=deltayer;
[valor,ycalc,X]= fobjRNAr(prop);
figure
plot(ycalc,'y')
hold on
plot(ycalc+U,'r')
plot(ycalc-U,'g')
plot(yer,'b')
```

G) Obtenção dos Parâmetros do Controlador DENFNF – MATLAB

- **Programa Principal**

```
clc
clear all
close all
%%%%%%%%%%%%% PERTURBAÇÃO %%%%%%
global perturbacao
dt=0.005;
t=0:dt:10;
perturbacao=zeros(2,length(t)); %%%
sd_sensor=0.1;%0.005;
for i=floor(0.05*length(t)):floor(0.8*length(t))
    perturbacao(1,i)=sd_sensor*randn;
end
sd_estado=0.05;%0.001;
for i=floor(0.05*length(t)):floor(0.8*length(t))
    perturbacao(2,i)=sd_estado*randn;
end
perturbacao(1,1)=1;
perturbacao(1,floor(0.3*length(t)))=-1.5;
perturbacao(1,floor(0.5*length(t)))=1;
var=treinamentoRNA;
[erro,uvet,yevet]=CalcErro(var);
```

```
figure
plot(t,uvet,'LineWidth',2)
xlabel('Iteration','FontSize',16)
ylim([-0.16 0.16])
ylabel('u_{N,k}','FontSize',16)
set(gca,'FontSize',16)

figure
plot(t,yevet,'LineWidth',2)
xlabel('Iteration','FontSize',16)
ylim([-2.5 2.5])
ylabel('Pitch [°]','FontSize',16)
set(gca,'FontSize',16)

fid = fopen('OUTPUT.txt','wt');
fprintf(fid,'%f\n',var);
%fprintf(fid,'%f %f\n',uvet,yevet);
fprintf(fid,'%f\n',uvet);
fprintf(fid,'%f\n',yevet);
fclose(fid);
```

- **Evolução Diferencial**

```
%Programa análogo ao treinamento MODELO do apêndice B

function var= treinamentoRNA
global perturbacao
nmaxgeracoes=20;
nind=20;
nvar=12;
pcruz=0.95;%Probabilidade de ocorrencia de soma
vetorial
pvariacao=0;
F=0.4;
ind=zeros(nind,nvar); %individuos
aind=ind; %auxiliar
find=zeros(nind, 1); %valor da fobj para cada
individuo
(...)
find(i)=-1*CalcErro(var');
(...)
figure
plot(vetcontador,vetmin,'LineWidth', 3)
xlabel('Iteração', 'FontSize', 16)
ylabel('EM^2', 'FontSize', 16)
set(gca, 'FontSize', 16)
grid on
```

- **Função Objetivo**

```
function [erro,uvet,yevet]=CalcErro(prop)
yemaxs=2;
yemax=2;
deltayemax=0.4;
dt=0.005;
u=0;
ua=0;
ye=0; %%%%%% DISTURBIO INICIAL DO ANGULO
deltaye=0;
t=0:0.005:10;
uvet=[];
yevet=[];
deltayevet=[];
for i=1:length(t) %i=1:10 %i=1:length(t)
    auxye=ye/1; %% % CONTROLADOR
    auxdeltaye=(deltaye)/1;
    if auxye>1
        auxye=1;
    end
    if auxye<-1
        auxye=-1;
    end
```

```
if auxye<-1
    auxye=-1;
end
if auxdeltaye>1
    auxdeltaye=1;
end
if auxdeltaye<-1
    auxdeltaye=-1;
end
X=plantaneurofuzzy([u,auxye,auxdeltaye],0);
X=X*yemaxs;
yea=ye;
ye=X;
deltaye=ye-yea;

ye=ye+perturbacao(1,i);
deltaye=deltaye+perturbacao(2,i);
end
n=length(yevet);
setvet=zeros(1,length(t));
erro=2*(sum((yevet-setvet).^2)/n)+20*(sum((uvet-
setvet).^2)/n)+0.5*(sum((deltayevet-setvet).^2)/n);
end
```

- Controlador Neuro-Fuzzy

```
function output=neurofuzzy2(input,prop)
prop=[prop(1:4);prop(5:8);prop(9:12)];
u1=zeros(3,1);
u2=zeros(3,3); %% 3 entradas com 3 subconjuntos fuzzy
m=[-1 0 1; -1 0 1;-1 0 1]; %%f. pert. gaussiana
s=ones(2,3)*0.4;
s=[s;0.8 0.2 0.8];
s2=s.^2;
u3=zeros(3,1); %% Número de Regras Fuzzy=3
u4=zeros(3,1);
output=0;
%%%primeira etapa: entrada de dados
for i=1:3
    u1(i)=input(i);
end
%%%segunda etapa: obtenção dos graus de pertinência
for i=1:3
    for j=1:3
        u2(i,j)=exp(-(u1(i)-m(i,j))^2/s2(i,j));
    end
end
```

```
%%%terceira etapa: operador "e"
%u3(1) - associado a neg
%u3(2) - associado a LOW
%u3(3) - associado a pos
u3(1)=max([u2(1,1) u2(2,1) u2(1,2)*u2(2,2)*u2(3,1)]);
u3(2)=u2(1,2)*u2(2,2)*u2(3,2);
u3(3)=max([u2(1,3) u2(2,3) u2(1,2)*u2(2,2)*u2(3,3)]);
aux=0;
for i=1:3
    aux=aux+1;
    plano=prop(aux,:);
    u4(i)=u3(i)*(plano(1)*input(1)+plano(2)*input(2)+plano(3)*input(3)+plano(4));
end
%%% quinta etapa: saída do sistema neurofuzzy
output=sum(u4)/sum(u3);
if output>1
    output=1;
end
if output<-1
    output=-1;
end
end
```

- **Modelo Neuro-Fuzzy**

```

function [output,u3]=plantaneurofuzzy(input,prop)

prop=[0.000000    0.667233    0.100294    0    0.000000
0.993516    0.081839    0    0.000000    0.667233
0.100294    0];

prop=[prop(1:4);prop(5:8);prop(9:12)];

u1=zeros(3,3);

u2=zeros(3,3); %% 3 entradas com 3 subconjuntos fuzzy

m=[-1 0 1; -1 0 1; -1 0 1]; %%f.pert. gaussiana

s=[0.5 0.2 0.5; 0.4 0.4 0.4; 0.4 0.4 0.4];

s2=s.^2;

u3=zeros(3,1); %% Número de Regras Fuzzy=3

u4=zeros(3,1);

%output=0;

%%%primeira etapa: entrada de dados

for i=1:3

    u1(i)=input(i);

end

%%%segunda etapa: obtenção dos graus de pertinência

for i=1:3 %% VARIÁVEL

    for j=1:3 %% SUBCONJUNTO

        u2(i,j)=exp(-(u1(i)-m(i,j))^2/s2(i,j));

    end

end

```

```

function [output,u3]=plantaneurofuzzy(input,prop)

prop=[0.000000    0.667233    0.100294    0    0.000000
0.993516    0.081839    0    0.000000    0.667233
0.100294    0];

prop=[prop(1:4);prop(5:8);prop(9:12)];

u1=zeros(3,3);

u2=zeros(3,3); %% 3 entradas com 3 subconjuntos fuzzy

m=[-1 0 1; -1 0 1; -1 0 1]; %%f.pert. gaussiana

s=[0.5 0.2 0.5; 0.4 0.4 0.4; 0.4 0.4 0.4];

s2=s.^2;

u3=zeros(3,1); %% Número de Regras Fuzzy=3

u4=zeros(3,1);

%output=0;

%%%primeira etapa: entrada de dados

for i=1:3

    u1(i)=input(i);

end

%%%segunda etapa: obtenção dos graus de pertinência

for i=1:3 %% VARIÁVEL

    for j=1:3 %% SUBCONJUNTO

        u2(i,j)=exp(-(u1(i)-m(i,j))^2/s2(i,j));

    end

end

```

H) Controlador DENFNF – Arduino

```
#include<Wire.h>
#include<Kalman.h>
#include<math.h>
#define M1_IN1 4
#define M1_IN2 5
#define M2_IN1 4
#define M2_IN2 5
#define M_PWM 3
#define T amostragem 5e-3
#define PI_T 3.1415
#define PWM_NVL 4096*6/8
Kalman kalmanX; // Create the Kalman instances
float kalmanY;
double accX, accY, accZ;
double gyroX;
double kalAngleX, kalAngleY; // Kalman filter
double omega;
double tempo = 0;
uint32_t N_Ciclo=0;
uint32_t timer;
uint32_t timer1;
uint8_t i2cData[15]; // Buffer for I2C data
```

```
if (u1[i] > 1) {  
    u1[i] = 1;  
}  
if (u1[i] < -1) {  
    u1[i] = -1;  
}  
}  
//%%segunda etapa: obtenção dos graus de pertinência  
for (int i = 0; i < 3; i++) {  
    for (int j = 0; j < 3; j++) {  
        u2[i][j] = exp(-pow(u1[i] - m[i][j], 2) /  
s2[i][j]);  
    }  
}  
//terceira etapa: operador "e"  
//u3(1) - associado a neg  
//u3(2) - associado a LOW  
//u3(3) - associado a pos  
//Segue a base de regras do CONTROLE2.fis  
u3[0] = fmax(fmax(u2[0][0], u2[1][0]), u2[0][1] *  
u2[1][1] * u2[2][0]);  
u3[1] = u2[0][1] * u2[1][1] * u2[2][1];  
u3[2] = fmax(fmax(u2[0][2], u2[1][2]), u2[0][1] *  
u2[1][1] * u2[2][2]);
```

```
for (int i = 0; i < 3; i++) {  
    u4[i] = u3[i] * (prop[i][0] * input[0] +  
    prop[i][1] * input[1] + prop[i][2] * input[2]);  
}  
  
//quinta etapa: saída do sistema neurofuzzy  
uot = (u4[0] + u4[1] + u4[2]) / (u3[0] + u3[1] +  
u3[2])/5;  
uot = 0.2*input[2] + uot;  
if (uot > 0.9) {  
    uot = 0.9;  
}  
if (uot < -0.9) {  
    uot = -0.9;  
}  
////////// Acionamento dos Motores ///////////  
int x;  
x = abs(uot) * PWM_NVL;  
if (x > PWM_NVL) x = PWM_NVL;  
if (uot > 0) {  
    analogWrite(M1_IN1, 0);  
    analogWrite(M1_IN2, 4096);  
    analogWrite(M_PWM, x);  
}
```

```
else if (uot < 0) {  
    analogWrite(M1_IN1, 4069);  
    analogWrite(M1_IN2, 0);  
    analogWrite(M_PWM, x);  
}  
else {  
    analogWrite(M1_IN1, 4096);  
    analogWrite(M1_IN2, 4096);  
    analogWrite(M_PWM, 0);  
}  
}  
void setup() {  
    Serial.begin(57600);  
    pinMode(M_PWM, OUTPUT);  
    pinMode(M1_IN1, OUTPUT);  
    pinMode(M1_IN2, OUTPUT);  
    pinMode(M2_IN1, OUTPUT);  
    pinMode(M2_IN2, OUTPUT);  
    pinMode(LED_BUILTIN, OUTPUT);  
    Wire.begin();  
#if ARDUINO >= 157  
    Wire.setClock(100000UL); // I2C 100kHz
```

```
#else
    TWBR = ((F_CPU / 100000UL) - 16) / 2; // Set I2C
frequency to 100kHz , Standard-mode
#endif

i2cData[0] = 7;
i2cData[1] = 0x00;
i2cData[2] = 0x00;
i2cData[3] = 0x00;
while (i2cWrite(0x19, i2cData, 4, false));
while (i2cWrite(0x6B, 0x01, true));
while (i2cRead(0x3B, i2cData, 6));
accX = (int16_t)((i2cData[0] << 8) | i2cData[1]);
accY = (int16_t)((i2cData[2] << 8) | i2cData[3]);
accZ = (int16_t)((i2cData[4] << 8) | i2cData[5]);
#endif RESTRICT_PITCH // Eq. 25 and 26
double roll = atan2(accY, accZ) * RAD_TO_DEG;
double pitch = atan(-accX / sqrt(accY * accY + accZ
* accZ)) * RAD_TO_DEG;
#endif // Eq. 28 and 29
double roll = atan(accY / sqrt(accX * accX + accZ
* accZ)) * RAD_TO_DEG;
double pitch = atan2(-accX, accZ) * RAD_TO_DEG;
```

```
#endif

    kalmanX.setAngle(roll); // Set starting angle
    kalmanY = pitch;
    timer = micros();
    timer1 = micros();
}

void loop() {
    /* Update all the values */
    while ((i2cRead(0x3B, i2cData, 14))) {
    }

    accX = (int16_t)((i2cData[0] << 8) | i2cData[1]);
    accY = (int16_t)((i2cData[2] << 8) | i2cData[3]);
    accZ = (int16_t)((i2cData[4] << 8) | i2cData[5]);
    gyroX = (int16_t)((i2cData[8] << 8) | i2cData[9]);

#ifdef RESTRICT_PITCH // Eq. 25 and 26
    double roll = atan2(accY, accZ) * RAD_TO_DEG;
    double pitch = atan(-accX / sqrt(accY * accY + accZ
* accZ)) * RAD_TO_DEG;
#else // Eq. 28 and 29
    double roll = atan(accY / sqrt(accX * accX + accZ
* accZ)) * RAD_TO_DEG;
    double pitch = atan2(-accX, accZ) * RAD_TO_DEG;
#endif
```

I) Método OnMoDE – Arduino

```
#include <Wire.h>
#include <Kalman.h>
#include <math.h>
#define M1_IN1 4
#define M1_IN2 5
#define M2_IN1 4
#define M2_IN2 5
#define M_PWM 3
#define T amostragem 5e-3
#define PI_T 3.1415
#define PWM_NVL 4096*6/8
Kalman kalmanX; // Create the Kalman instances
float kalmanY;
double accX, accY, accZ;
double gyroX;
double kalAngleX, kalAngleY; // Calculated angle
using a Kalman filter
double omega;
double tempo = 0;
uint32_t N_Ciclo;
uint32_t timer;
uint32_t timer1;
uint8_t i2cData[15]; // Buffer for I2C data
```

```
#define nind 3
#define nusos 10
#define nvar 9
////////// VARIÁVEIS CONTROLE //////////
double input[3] = {0, 0, 0};
double prop[3][3] = {0.0808, 0.0422, 0.0426, 0.1554,
0.2286, 0.0705, 0.0808, 0.0422, 0.0426};
double u1[3] = {0, 0, 0};
double u2[3][3] = {0, 0, 0, 0, 0, 0, 0, 0, 0};
double m[3][3] = { -1, 0, 1, -1, 0, 1, -1, 0, 1};
double s2[3][3] = {0.16, 0.16, 0.16, 0.16, 0.16,
0.16, 0.64, 0.04, 0.64};
double u3[3] = {0, 0, 0};
double u4[3] = {0, 0, 0};
double uot;
float teta[2] = {0, 0};
int x;
double theta;
//// VARIÁVEIS OTIMIZAÇÃO //////////
int casal[nind];
double param0[nind][nvar];
double param1[nind][nvar];
double err = 0;
```

```
double erro0[nind];
double erro1[nind];
double deltayevetotim[nusos];
int conti = 1;
int contii;
int contu = 0;
int flag = 0;
double linf[nvar] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
Double lsup[nvar]=
{0.2,0.2,0.1,0.4,0.4,0.2,0.2,0.2,0.1};
int flagmut = 0;
double probmut[2] = {0.05, 0.01};
double serro0;
double serrol;
double aleatorio;
void randperm() {
    for (int i = 0 ; i < nind; i++) {
volta:
    casal[i] = random(0, nind);
    for (int j = 0; j < i; j++) {
        if (casal[i] == casal[j]) {
            goto volta;
        }
    }
}
```

```

void otim01() {
    if (contu != 0) {
        err = err + 1 * (teta[0] * teta[0]) + 20 * (uot * uot) + 0.5 * ((teta[0] - teta[1]) * (teta[0] - teta[1]));
        if (contu == nusos) {
            erro0[conti - 1] = err;
            erro1[conti - 1] = erro0[conti - 1];
            err = 0;
        }
    }
    if (contu == nusos && conti != nind) {
        conti = conti + 1;
        contu = 0;
    }
    else if (contu == nusos && conti == nind) {
        conti = 1;
        contu = 0;
        flag = 1;
    }
    if (flag == 0) {
        contu = contu + 1;
        if (contu == 1) {

```

```
for (int i = 0; i < nvar; i++) {  
    if (i == 6 || i == 7 || i == 8) {  
        param0[conti - 1][i] = param0[conti - 1][i - 6];  
        param1[conti - 1][i] = param0[conti - 1][i];  
    }  
    else {  
        param0[conti - 1][i] = random(1000) / 1000.0;  
        param0[conti - 1][i] = (lsup[i] - linf[i]) *  
        param0[conti - 1][i] + linf[i];  
        param1[conti - 1][i] = param0[conti - 1][i];  
    }  
}  
for (int i = 0; i < 3; i++) {  
    for (int j = 0; j < 3; j++) {  
        prop[i][j] = param0[conti - 1][i * 3 + j];  
    }  
}  
control();  
}  
}
```

```
void otim02() {  
    if (contu != 0) {  
        err = err + 2 * (teta[0] * teta[0]) + 5 * (uot * uot)  
        + 1 * ((teta[0] - teta[1]) * (teta[0] - teta[1]));  
        if (contu == nusos) {  
            if (err < erro0[conti - 1]) {  
                errol[conti - 1] = err;  
            }  
            else {  
                errol[conti - 1] = erro0[conti - 1];  
                for (int i = 0; i < nvar; i++) {  
                    param1[conti - 1][i] = param0[conti - 1][i];  
                }  
            }  
            err = 0;  
            if (conti == nind) {  
                serro0 = 0;  
                serrol = 0;  
                for (int i = 0; i < nind; i++) {  
                    serro0 = serro0 + erro0[i];  
                    serrol = serrol + errol[i];  
                }  
            }  
        }  
    }  
}
```

```
if (fabs(serrol - serro0) / serro0 < 0.01) {  
    flagmut = 1;  
}  
else {  
    flagmut = 0;  
}  
memcpy(erro0, errol, sizeof(errol));  
memcpy(param0, param1, sizeof(param1));  
}  
}  
}  
if (contu == nusos && conti != nind) {  
    conti = conti + 1;  
    contu = 0;  
}  
else if (contu == nusos && conti == nind) {  
    conti = 1;  
    contu = 0;  
    randperm();  
}  
contu = contu + 1;
```

```
if (contu == 1) {  
    aleatorio = random(1000) / 1000.0;  
    if (aleatorio < probmut[flagmut]) {  
        for (int i = 0; i < nvar; i++) {  
            if (i == 6 || i == 7 || i == 8) {  
                param1[conti - 1][i] = param1[conti - 1][i - 6];  
            }  
            else {  
                param1[conti - 1][i] = random(1000) / 1000.0;  
                param1[conti - 1][i] = (lsup[i] - linf[i])  
                * param1[conti - 1][i] + linf[i];  
            }  
        }  
    }  
    else {  
        contii = conti + 1;  
        if (contii == nind + 1) {  
            contii = 1;  
        }  
        aleatorio = random(1000) / 1000.0;  
        for (int i = 0; i < nvar; i++) {  
            param1[conti - 1][i] = param0[conti - 1][i] +  
            aleatorio * (param0[casal[conti - 1]][i] -  
            param0[casal[contii - 1]][i]);  
        }  
    }  
}
```

```
if (param1[conti - 1][i] < linf[i]) {
    param1[conti - 1][i] = linf[i];
}
if (param1[conti - 1][i] > lsup[i]) {
    param1[conti - 1][i] = lsup[i];
} } }

for (int i = 0; i < 3; i++) { //numero de
subconjuntos fuzzy na saída
    for (int j = 0; j < 3; j++) { //num de
variaveis de entrada
        prop[i][j] = param1[conti - 1][i * 3 + j];
    }
}
control();
}

void control()
{
////////////////// CONTROLE //////////////////
//%%primeira etapa: entrada de dados
input[0]=teta[0]/1;
input[1]=(teta[0]-teta[1])/1;
input[2]=uot;
```

```
for (int i = 0; i < 2; i++) {  
    u1[i] = input[i];  
    if (u1[i]>1) {  
        u1[i]=1;  
    }  
    if (u1[i]<-1) {  
        u1[i]=-1;  
    }  
}  
//segunda etapa: obtenção dos graus de pertinência  
for (int i = 0; i < 3; i++) {  
    for (int j = 0; j < 3; j++) {  
        u2[i][j] = exp(-pow(u1[i] - m[i][j], 2) / s2[i][j]);  
    }  
}  
//terceira etapa: operador "e"  
u3[0]=fmax(fmax(u2[0][0],u2[1][0]),u2[0][1]*u2[1][1]*  
u2[2][0]);  
  
u3[1] = u2[0][1] * u2[1][1] * u2[2][1];  
  
u3[2]=fmax(fmax(u2[0][2],u2[1][2]),u2[0][1]*u2[1][1]*  
u2[2][2]);
```

```
for (int i = 0; i < 3; i++) {  
    u4[i] = u3[i] * (prop[i][0] * input[0] +  
    prop[i][1] * input[1] + prop[i][2] * input[2]);  
}  
//quinta etapa: saída do sistema neurofuzzy  
uot=(u4[0]+u4[1]+u4[2]) / (u3[0]+u3[1]+u3[2]);  
uot=1*input[2]+uot;  
if (uot>1){  
    uot=1;}  
if (uot<-1){  
    uot=-1;}  
x = abs(uot) * PWM_NVL; // Acionamento dos Motores  
if (x > PWM_NVL) x = PWM_NVL;  
if (uot > 0) {  
    analogWrite(M1_IN1, 0);  
    analogWrite(M1_IN2, 4096);  
    analogWrite(M_PWM, x);  
}  
else if (uot < 0) {  
    analogWrite(M1_IN1, 4069);  
    analogWrite(M1_IN2, 0);  
    analogWrite(M_PWM, x);  
}
```

```
else {  
    analogWrite(M1_IN1, 4096);  
    analogWrite(M1_IN2, 4096);  
    analogWrite(M_PWM, 0);  
}  
}  
  
void setup() {  
    Serial.begin(57600);  
    pinMode(M_PWM, OUTPUT);  
    pinMode(M1_IN1, OUTPUT);  
    pinMode(M1_IN2, OUTPUT);  
    pinMode(M2_IN1, OUTPUT);  
    pinMode(M2_IN2, OUTPUT);  
    pinMode(LED_BUILTIN, OUTPUT);  
    Wire.begin();  
  
#if ARDUINO >= 157  
    Wire.setClock(100000UL);  
#else  
    TWBR = ((F_CPU / 100000UL) - 16) / 2;  
#endif  
  
    i2cData[0] = 7;  
    i2cData[1] = 0x00;  
    i2cData[2] = 0x00;  
    i2cData[3] = 0x00;
```

```
while (i2cWrite(0x19, i2cData, 4, false)); // Write to all four registers at once

while (i2cWrite(0x6B, 0x01, true));

while (i2cRead(0x3B, i2cData, 6));

accX = (int16_t)((i2cData[0] << 8) | i2cData[1]);
accY = (int16_t)((i2cData[2] << 8) | i2cData[3]);
accZ = (int16_t)((i2cData[4] << 8) | i2cData[5]);

#ifndef RESTRICT_PITCH // Eq. 25 and 26

    double roll = atan2(accY, accZ) * RAD_TO_DEG;
    double pitch = atan(-accX / sqrt(accY * accY + accZ * accZ)) * RAD_TO_DEG;

#else // Eq. 28 and 29

    double roll = atan(accY / sqrt(accX * accX + accZ * accZ)) * RAD_TO_DEG;
    double pitch = atan2(-accX, accZ) * RAD_TO_DEG;

#endif

    kalmanX.setAngle(roll); // Set starting angle
    kalmanY = pitch;
    timer = micros();
    timer1 = micros();
}
```

```
void loop() {  
    /* Update all the values */  
    while ((i2cRead(0x3B, i2cData, 14))) {  
    }  
    accX = (int16_t)((i2cData[0] << 8) | i2cData[1]);  
    accY = (int16_t)((i2cData[2] << 8) | i2cData[3]);  
    accZ = (int16_t)((i2cData[4] << 8) | i2cData[5]);  
    gyroX = (int16_t)((i2cData[8] << 8) | i2cData[9]);  
#ifdef RESTRICT_PITCH // Eq. 25 and 26  
    double roll = atan2(accY, accZ) * RAD_TO_DEG;  
    double pitch = atan(-accX / sqrt(accY * accY + accZ  
* accZ)) * RAD_TO_DEG;  
#else // Eq. 28 and 29  
    double roll = atan(accY / sqrt(accX * accX + accZ  
* accZ)) * RAD_TO_DEG;  
    double pitch = atan2(-accX, accZ) * RAD_TO_DEG;  
#endif  
    omega = (gyroX / 131.0); // Convert to deg/s  
    roll = roll;  
    double dt = (double)(micros() - timer) / 1000000;  
// Calculate delta time  
    timer = micros();
```

```
if ((pitch < -90 && kalAngleY > 90) || (pitch > 90 && kalAngleY < -90)) {  
    kalmanY = pitch;  
}  
else  
    kalAngleY = pitch;  
  
if (abs(kalAngleY) > 90) omega = -omega;  
omega = omega + 4.1744-0.1722;  
kalAngleX = kalmanX.getAngle(roll, omega, dt);  
  
float cft = 0.75;  
  
theta = (1-cft)*(0.9 * (theta + omega * dt) + 0.1 *  
roll) + cft*kalAngleX ;  
  
digitalWrite(LED_BUILTIN, HIGH );  
teta[1] = teta[0];  
teta[0] = theta;  
  
tempo = (double) (micros() - timer1) / 1000000;  
if (tempo > (double) N_Ciclo * T amostragem) {  
    N_Ciclo = N_Ciclo + 1;  
    if (flag == 0) {  
        otim01();  
    }  
    if (flag == 1) {  
        otim02();  
    } } }
```