

Airton Silva Monteiro

**Projeto e controle de Pêndulo Invertido sobre  
Rodas baseado no modelo *fuzzy* Takagi-Sugeno**

Manaus - AM

2025

Airton Silva Monteiro

## **Projeto e controle de Pêndulo Invertido sobre Rodas baseado no modelo *fuzzy* Takagi-Sugeno**

Trabalho de Conclusão de Curso apresentado à banca avaliadora do curso de Engenharia de Controle e Automação da Escola Superior de Tecnologia da Universidade do Estado do Amazonas como pré-requisito para obtenção do título de Engenheiro de Controle e Automação.

Universidade do Estado do Amazonas – UEA

Escola Superior de Tecnologia – EST

Engenharia de Controle e Automação

Orientador: Rodrigo Farias Araújo

Coorientador: Almir Kimura Junior

Manaus - AM

2025

## Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).  
**Sistema Integrado de Bibliotecas da Universidade do Estado do Amazonas.**

M772p

Monteiro, Airton

Projeto e controle de Pêndulo Invertido sobre Rodas baseado no modelo fuzzy Takagi-Sugeno / Airton Monteiro . Manaus : [s.n], 2025.

78 f.: color.; 21,0 cm.

TCC - Graduação em Engenharia de Controle e Automação-  
Universidade do Estado do Amazonas, Manaus, 2025.

Inclui Bibliografia.

Orientador: Farias Araújo, Rodrigo.

Coorientador: Kimura Junior, Almir.

1. Sistema não linear. 2. pêndulo invertido sobre rodas. 3. Controle PDC. 4. Modelos fuzzy Takagi-Sugeno. I. Farias Araújo, Rodrigo (Orient.) II . Kimura Junior, Almir (Coorient.) III. Universidade do Estado do Amazonas. IV. Título

CDU(1997)681.5

Airton Silva Monteiro

## **Projeto e controle de Pêndulo Invertido sobre Rodas baseado no modelo *fuzzy* Takagi-Sugeno**

Trabalho de Conclusão de Curso apresentado à banca avaliadora do curso de Engenharia de Controle e Automação da Escola Superior de Tecnologia da Universidade do Estado do Amazonas como pré-requisito para obtenção do título de Engenheiro de Controle e Automação.

Trabalho aprovado. Manaus - AM, 13 de Junho de 2025;

*Luiz Alberto Queiroz Cordovil Júnior*

**Prof. Dr. Luiz Alberto Queiroz  
Cordovil Júnior**

Sidia/Samsung e Universidade do Estado do  
Amazonas (UEA)

*Vitor Fernando de Souza Gadelha*

**Prof. Esp. Vitor Fernando de Souza  
Gadelha**

Instituto Federal do Amazonas (IFAM)

*Israel Mazaira Morales*

**Prof. Dr. Israel Mazaira Morales**  
Universidade Estadual do Amazonas (UEA)

Manaus - AM  
2025

# Agradecimentos

A Deus e à Nossa Senhora Aparecida. À minha família, que sempre me apoiou ao longo desta jornada. Em especial à minha mãe, que se esforçou para transmitir a mim e ao meu irmão o valor da educação; ao meu pai, que foi minha primeira referência como engenheiro, mesmo sem sê-lo formalmente, e me inspirou a desenvolver a capacidade de buscar soluções para os problemas; e ao meu irmão, que sempre será para mim um exemplo de determinação.

À minha namorada, Brenda, que me acompanha há 5 anos, tanto nos momentos de angústia e luta quanto nas vitórias alcançadas. Aos bons amigos que fiz durante esse período na faculdade, (caso leiam esse trabalho, saberão quem são), com quem coletei diversos momentos de alegria, tornando esse período mais fácil de ser vivido.

Aos meus orientadores, Prof. Rodrigo Farias Araújo e Prof. Almir Kimura Junior, pois foram suas aulas que me interessaram e impulsionaram a buscar conhecer mais sobre a área de sistemas de controle, além de me orientarem durante o desenvolvimento deste trabalho.

Ao Laboratório de Fabricação Digital do OCEAN, que foi minha segunda casa, e a todos que, de alguma forma, contribuíram para a realização deste trabalho.

Por fim, à Universidade do Estado do Amazonas por ter sido um lugar de muito aprendizado e um espaço de desenvolvimento profissional.

*Não importa quanto a vida possa ser ruim, sempre existe algo que você pode fazer, e triunfar. Enquanto há vida, há esperança. (Hawking, Stephen)*

# Resumo

A implementação em tempo real de sistemas de controle frequentemente envolve incertezas associadas à dinâmica complexa e à natureza não linear do sistema, o que dificulta sua análise analítica para o projeto de controladores. Neste contexto, este trabalho aborda o desenvolvimento de um pêndulo invertido sobre rodas, utilizado como plataforma experimental para aplicação de técnicas de controle. A modelagem do sistema é realizada por meio de modelos *fuzzy* Takagi-Sugeno, permitindo representar sua dinâmica não linear a partir da combinação de modelos lineares locais. Para estabilização do sistema, é empregado um controlador baseado na técnica de *Parallel Distributed Compensation* (PDC). O desenvolvimento inclui tanto a construção da planta física quanto a implementação do controlador digital em um sistema embarcado, responsável pela aquisição dos sinais, processamento das leis de controle e acionamento dos atuadores. Os resultados obtidos indicam que o controlador baseado na técnica PDC foi capaz de estabilizar o pêndulo invertido sobre duas rodas na posição de equilíbrio vertical, além de realizar rastreamento de referência com desempenho satisfatório em condições práticas. O sistema respondeu de forma adequada frente a pequenas perturbações, embora tenham sido observadas limitações relacionadas à capacidade dos atuadores em situações de deslocamento de maior amplitude. De modo geral, o trabalho evidencia a viabilidade da aplicação de controladores baseados em modelos *fuzzy* Takagi-Sugeno em sistemas reais, considerando os desafios inerentes à implementação prática.

**Palavras-chave:** Sistema não linear, pêndulo invertido sobre rodas, Controle PDC, Modelos *fuzzy* Takagi-Sugeno.

# Abstract

Real-time implementation of control systems often involves uncertainties associated with the system's complex dynamics and nonlinear nature, which makes its analytical analysis challenging for controller design. In this context, this work addresses the development of a two-wheeled inverted pendulum, used as an experimental platform for the application of control techniques. The system modeling is carried out using the Takagi-Sugeno methodology, which allows representing its nonlinear dynamics through the combination of local linear submodels. For system stabilization, a controller based on the *Parallel Distributed Compensation* (PDC) technique is employed. The development includes both the construction of the physical plant and the digital implementation of the controller on an embedded system, responsible for signal acquisition, control law processing, and actuator driving. The results indicate that the controller based on the PDC fuzzy technique was able to stabilize the two-wheeled inverted pendulum in the upright position, as well as perform reference tracking with satisfactory performance under practical conditions. The system responded adequately to small disturbances, although some limitations related to actuator capacity were observed in situations involving larger displacement demands. Overall, the work demonstrates the feasibility of applying controllers based on Takagi-Sugeno models to real systems, considering the challenges inherent to practical implementation.

**Keywords:** Nonlinear system, two-wheeled inverted pendulum, digital controller, Takagi-Sugeno fuzzy models, PDC.

# Lista de ilustrações

Figura 1 – Aplicações práticas do pêndulo invertido sobre duas rodas. . . . .	11
Figura 2 – Diagrama de corpo livre para o pêndulo invertido sobre o carro. . . . .	15
Figura 3 – Exemplos de funções de pertinência parametrizadas: (a) triangular( $x; 20, 60, 80$ ); (b) trapezoidal( $x; 10, 20, 60, 95$ ); (c) gaussiana( $x; 50, 20$ ); (d) $sino(x; 20, 4, 50)$ . . . . .	25
Figura 4 – Exemplo de Sistema Fuzzy Genérico. . . . .	27
Figura 5 – Funções de pertinência associadas às duas regras fuzzy do sistema pêndulo invertido. . . . .	31
Figura 6 – Diagrama de Blocos de um Servomecanismo de tempo contínuo. . . . .	33
Figura 7 – Diagrama de Blocos de um sistemas de controle amostrado . . . . .	34
Figura 8 – Sinal gerado pelo segurador de ordem zero. . . . .	34
Figura 9 – Desenho CAD das bases do pêndulo. . . . .	37
Figura 10 – Visualização de posicionamento das hastes. . . . .	37
Figura 11 – Peças para suporte e fixação dos dispositivos utilizados. . . . .	38
Figura 12 – Estrutura montada via <i>software</i> CAD e estrutura real do sistema. . . . .	39
Figura 13 – Configuração dos pinos na ESP32-DevKitC. . . . .	40
Figura 14 – Módulo MPU6050. . . . .	41
Figura 15 – Módulo MPU6050. . . . .	42
Figura 16 – Motor de Passo NEMA 17 Modelo: 42HS40-1704-13A. . . . .	43
Figura 17 – Driver Motor de Passo DRV8825. . . . .	43
Figura 18 – CNC Shield V3. . . . .	44
Figura 19 – Etapas do projeto e fabricação da PCI. . . . .	45
Figura 20 – Diagrama de Blocos do Sistema Proposto . . . . .	46
Figura 21 – Variáveis das equações de movimento para um robô de duas rodas. . . . .	48
Figura 22 – Gráfico das funções de pertinência considerando $\theta = 0, -30^\circ, 30^\circ$ . . . . .	51
Figura 23 – Localização do centro de gravidade em relação ao eixo das rodas em $l = 62, 90\text{mm}$ . . . . .	57
Figura 24 – Resposta do ângulo $\theta$ à aplicação de distúrbios sob ação do controlador LQR. . . . .	59
Figura 25 – Resposta da posição linear $x$ à aplicação de distúrbios com ação do controlador LQR. . . . .	59
Figura 26 – Sinal de controle (Torque) com aplicação de distúrbio no sistema. . . . .	60
Figura 27 – Resposta do ângulo de inclinação $\theta$ sob distúrbio, com ação do controla- dor PDC (3 regras <i>fuzzy</i> ). . . . .	62
Figura 28 – Resposta da posição linear $x$ à aplicação do distúrbio com controlador PDC (3 regras <i>fuzzy</i> ). . . . .	62

Figura 29 – Evolução do sinal de controle em torque sob distúrbio com controlador PDC (3 regras <i>fuzzy</i> ). . . . .	63
Figura 30 – Gráfico das funções de pertinência considerando $\theta = -45^\circ, -30^\circ, 0, 30^\circ, 45^\circ$ . . . . .	65
Figura 31 – Resposta do ângulo de inclinação $\theta$ à aplicação do distúrbio com controlador PDC (5 regras <i>fuzzy</i> ). . . . .	65
Figura 32 – Resposta da posição linear $x$ à aplicação do distúrbio com controlador PDC (5 regras <i>fuzzy</i> ). . . . .	66
Figura 33 – Evolução do sinal de control em torque sob distúrbio com controlador PDC (5 regras <i>fuzzy</i> ). . . . .	66
Figura 34 – Resposta da posição linear $x$ a um degrau de 10 cm, sob ação do controlador LQR. . . . .	67
Figura 35 – Evolução do ângulo de inclinação $\theta$ durante o rastreamento de posição, com controlador LQR. . . . .	67
Figura 36 – Resposta da posição linear $x$ a um degrau de 10 cm, sob ação do controlador <i>PDC</i> (3 regras <i>fuzzy</i> ) . . . . .	68
Figura 37 – Evolução do ângulo de inclinação $\theta$ durante o rastreamento de posição, sob ação do controlador <i>PDC</i> (3 regras <i>fuzzy</i> ). . . . .	68
Figura 38 – Resposta da posição linear $x$ a um degrau de 10 cm, sob ação do controlador <i>PDC</i> com 5 regras <i>fuzzy</i> . . . . .	69
Figura 39 – Evolução do ângulo de inclinação $\theta$ durante o rastreamento de posição, sob ação do controlador <i>PDC</i> com 5 regras <i>fuzzy</i> . . . . .	69
Figura 40 – Resposta da posição linear $x$ a um degrau de 15 cm, sob ação do controlador PDC com 5 regras <i>fuzzy</i> . . . . .	70

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
<b>1.1</b>	<b>Trabalhos Relacionados</b>	<b>12</b>
<b>1.2</b>	<b>Objetivos</b>	<b>13</b>
1.2.1	Objetivo geral	13
1.2.2	Objetivos específicos	13
<b>1.3</b>	<b>Estrutura do trabalho</b>	<b>13</b>
<b>2</b>	<b>REFERÊNCIAL TEÓRICO</b>	<b>14</b>
<b>2.1</b>	<b>Pêndulo Invertido de duas rodas</b>	<b>14</b>
2.1.1	Modelagem Matemática	14
2.1.2	Linearização e Modelo em Espaço de Estados	17
<b>2.2</b>	<b>Modelagem fuzzy</b>	<b>21</b>
2.2.1	Modelo fuzzy Takagi-Sugeno	29
2.2.2	Compensação Paralela Distribuida	32
<b>2.3</b>	<b>Sistemas de controle amostrado</b>	<b>33</b>
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>36</b>
<b>3.1</b>	<b>Projeto do pêndulo invertido sobre rodas</b>	<b>36</b>
3.1.1	Estrutura Mecânica	36
3.1.2	Sistema Computacional	38
3.1.3	Sensor IMU	40
3.1.4	Motor de passo	42
3.1.4.1	Driver de Controle	42
3.1.5	Placa de circuito impresso	43
<b>3.2</b>	<b>Sistema de Controle</b>	<b>44</b>
3.2.1	Aquisição de Medidas dos Estados	46
3.2.1.1	Filtro Complementar	46
3.2.1.2	Odometria	47
3.2.2	Projeto de controlador LQR	49
3.2.3	Projeto de controlador Fuzzy via PDC	50
<b>3.3</b>	<b>Sobre a implementação do sinal de controle</b>	<b>55</b>
<b>4</b>	<b>RESULTADOS</b>	<b>56</b>
<b>4.1</b>	<b>Parâmetros físicos</b>	<b>56</b>
<b>4.2</b>	<b>Estabilidade em torno da origem</b>	<b>58</b>
4.2.1	Controle LQR	58

4.2.2	Modelo com 3 Funções de Pertinência . . . . .	60
4.2.3	Modelo com 5 Funções de Pertinência . . . . .	62
4.3	<b>Rastreamento de uma entrada de referência . . . . .</b>	<b>65</b>
5	<b>CONCLUSÕES . . . . .</b>	<b>71</b>
5.1	<b>Trabalhos Futuros . . . . .</b>	<b>72</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>74</b>

# 1 Introdução

Dentro da área de sistemas de controle, o pêndulo invertido é amplamente estudado devido à sua característica não linear e por sua instabilidade natural, características que o tornam uma referência clássica para experimentação e validação de estratégias de controle. Segundo [Hazem e Bingül \(2023\)](#), esse tipo de sistema pode ser dividido em duas categorias principais, de acordo com sua estrutura e quantidade de atuadores: modelos com atuador único e com múltiplos atuadores. Na segunda categoria, destaca-se o pêndulo invertido sobre duas rodas, também conhecido como pêndulo invertido móvel ou robô de autoequilíbrio, cuja estrutura é composta por duas rodas laterais que se locomovem independentemente, permitindo estabilizar um corpo superior ao manter seu centro de massa na posição de equilíbrio sobre o eixo das rodas ([JIMENEZ; L.; RUGE, 2020](#)). Esta arquitetura é empregada em diferentes aplicações práticas, como no transporte de carga, podemos citar o robô *evoBot* fabricado pela *Fraunhofer IML*, além de soluções voltadas à mobilidade urbana por meio de veículos como o *Segway*. Estes dispositivos podem ser vistos na Figura 1.

Figura 1 – Aplicações práticas do pêndulo invertido sobre duas rodas.



(a) Robô *evoBot* da Fraunhofer IML



(b) Veículo *Ninebot S2*

Fonte: [FraunhoferIML \(2023\)](#); [Segway \(2023\)](#)

A implementação de sistemas de controle em aplicações reais, como o pêndulo invertido sobre duas rodas, envolve desafios adicionais em relação ao ambiente teórico. Incertezas na modelagem, limitações dos atuadores, ruídos nos sensores e efeitos não modelados impactam diretamente a estabilidade e o desempenho do sistema. Embora a modelagem linear em torno do ponto de equilíbrio seja uma abordagem comum, ela nem sempre captura de forma satisfatória o comportamento dinâmico do sistema, ([OGATA, 2010](#)).

## 1.1 Trabalhos Relacionados

Na literatura, são encontrados diversos estudos que tratam da implementação em tempo real do pêndulo invertido sobre duas rodas, os quais apresentam variações tanto na modelagem dinâmica quanto na escolha dos componentes físicos e nas estratégias de controle adotadas. A modelagem completa do sistema considera três graus de liberdade: a inclinação do pêndulo em relação ao eixo das rodas, o ângulo de orientação do chassi no plano e o movimento translacional do conjunto (GRASSER et al., 2002; CHAVES et al., 2020). Uma abordagem frequentemente adotada na literatura consiste em simplificar a dinâmica do sistema, desconsiderando o grau de liberdade associado à rotação no plano, de modo que o modelo represente apenas o movimento de translação e a inclinação do pêndulo (JIMENEZ; L.; RUGE, 2020). A partir dessa simplificação, é comum que o modelo do pêndulo invertido sobre duas rodas seja aproximado ao modelo clássico do pêndulo invertido sobre um carro, amplamente utilizado no ensino e pesquisa em sistemas de controle (IACOB, 2020; PUTRA; WAHAB; TAMBA, 2025).

As variações metodológicas também se refletem nas estratégias de controle aplicadas ao pêndulo invertido sobre rodas. Abordagens clássicas, como os controladores proporcional-integral-derivativo (PID) e regulador quadrático linear (LQR), são amplamente utilizadas devido à simplicidade de implementação e à fundamentação baseada na linearização do sistema. Contudo, essas técnicas apresentam limitações fora da região de operação linearizada. Nesse contexto, o controle baseado em modelos *fuzzy* surge como alternativa, permitindo representar o sistema e o controlador por meio de regras dependentes da dinâmica. Essa abordagem pode ser empregada na forma de controladores *fuzzy* linguísticos, muitas vezes como extensão dos PID (MAI et al., 2018), ou por meio de modelos Takagi-Sugeno, que combinam submodelos lineares locais associados a diferentes regiões operacionais (WANG; TANAKA; GRIFFIN, 1995). Nessa última, destaca-se a técnica de Compensação Distribuída Paralela (*Parallel Distributed Compensation* - PDC), amplamente utilizada para a síntese de leis de controle locais (GONZALEZ-CARDENAS et al., 2024).

Neste trabalho, busca-se representar a dinâmica do pêndulo invertido sobre duas rodas por meio de um modelo *fuzzy* Takagi-Sugeno baseado no sistema do pêndulo invertido sobre o carro, de modo a capturar as características não lineares do sistema a partir da combinação de modelos lineares locais. A partir dessa modelagem, propõe-se a aplicação da técnica de controle PDC, que permite a síntese de leis de controle associadas a cada submodelo, possibilitando a estabilização do sistema em diferentes condições operacionais. A abordagem adotada visa explorar a capacidade dessa metodologia em lidar com as variações dinâmicas presentes na operação do sistema em ambiente real.

## 1.2 Objetivos

### 1.2.1 Objetivo geral

Desenvolver um protótipo do pêndulo invertido sobre duas rodas e construir um modelo matemático capaz de representar sua dinâmica, tanto na região próxima ao equilíbrio quanto em diferentes condições de operação, a fim de projetar um controlador que permita a estabilização do sistema em posição de equilíbrio vertical.

### 1.2.2 Objetivos específicos

1. Projetar e modelar os aspectos mecânicos do sistema via *software* de modelagem 3D e realizar a fabricação das peças;
2. Dimensionar e projetar o funcionamento elétrico do sistema, realizando testes em *protoboard* e com programação;
3. Projetar e fabricar a placa de circuito impresso (PCI);
4. Integrar as partes mecânicas com as partes elétricas que foram desenvolvidas.
5. Obter um modelo matemático que consiga descrever a dinâmica do pêndulo;
6. Obter um modelo *fuzzy* Takagi-Sugeno e a partir deste modelo, projetar um controlador PDC;
7. Implementar metodologias para obtenção de medidas dos estados que compõem o sistema;
8. Implementar o controlador via algoritmo de programação em uma placa de desenvolvimento microcontroladora;
9. Realizar testes de funcionamento do sistema com ação do controlador;
10. Comparar o desempenho do sistema, com o desempenho simulado;
11. Comparar os resultados da pesquisa com outras pesquisas relacionadas.

## 1.3 Estrutura do trabalho

O trabalho está estruturado conforme os seguintes capítulos. O [Capítulo 2](#) abordará os aspectos teóricos relacionados à planta de controle escolhida, a utilização da lógica *fuzzy* em sistemas de controle e os aspectos relacionados à implementação de controle em sistemas reais. O [Capítulo 3](#) apresenta as metodologias utilizadas no desenvolvimento do trabalho. O [Capítulo 4](#) apresenta a discussão sobre os resultados obtidos através do controlador proposto. Por fim, o [Capítulo 5](#) apresenta as conclusões e sugestões de trabalhos futuros.

## 2 Referencial Teórico

Este capítulo apresenta os principais conceitos relacionados ao desenvolvimento deste trabalho. Inicialmente, são abordados os fundamentos da modelagem dinâmica do sistema, incluindo o diagrama de corpo livre e as leis físicas utilizadas na formulação das equações de movimento. Logo após, é feita uma breve introdução à modelagem de sistemas de controle por meio da lógica nebulosa, com ênfase no modelo de Takagi-Sugeno. Por fim, são discutidos os conceitos fundamentais envolvidos na implementação de controladores digitais em sistemas embarcados.

### 2.1 Pêndulo Invertido de duas rodas

#### 2.1.1 Modelagem Matemática

O pêndulo invertido sobre duas rodas pode ser descrito como um sistema mecânico composto por dois subsistemas: um corpo intermediário, que apresenta comportamento dinâmico semelhante ao de um pêndulo, e uma base móvel, cuja dinâmica se assemelha à de um carro (JIMENEZ; L.; RUGE, 2020). Essa configuração permite aproximar seu comportamento ao modelo de pêndulo invertido sobre um carrinho, conforme discutido por Jacob (2020).

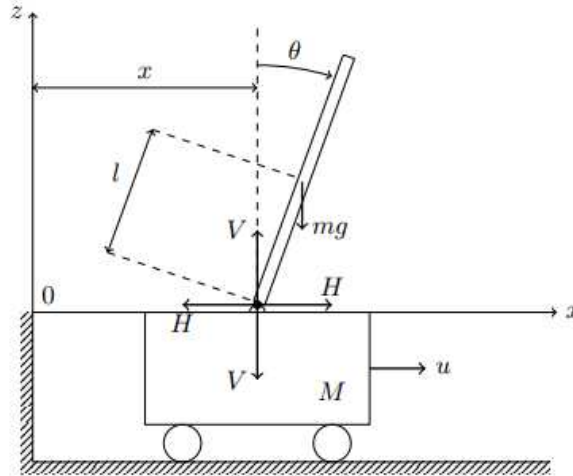
O objetivo neste trabalho é conseguir o equilíbrio vertical do pêndulo e realizar movimentos horizontais sobre o eixo  $x$ . Portanto, as seguintes ponderações foram feitas:

1. As rodas se movimentam sem deslizamento, são rígidas e possuem mesma massa e dimensão. Isto permite assumir que há contato puro entre as rodas e a superfície, eliminando efeitos de escorregamento e simplificando a relação entre a rotação das rodas e o deslocamento linear do sistema.
2. A massa do corpo intermediário está uniformemente distribuída ao longo de sua estrutura, permitindo considerar que seu centro de massa se localiza ao longo do eixo central vertical do corpo. Isto permite ter o momento de inércia definido em relação ao ponto de rotação sobre o eixo das rodas.

Para o desenvolvimento das equações de dinâmica do sistema, toma-se como base o diagrama de corpo livre apresentado na Figura 2. O sistema é composto por um pêndulo rígido de massa  $m$  e comprimento  $l$ , acoplado a um carrinho de massa  $M$  que se move sobre uma superfície horizontal sem atrito. O pêndulo está sujeito à ação da gravidade e conectado ao carrinho por uma articulação sem atrito, permitindo a rotação em torno

de seu ponto de fixação. O carrinho, por sua vez, é controlado por uma força de entrada  $u$ , aplicada horizontalmente. A coordenada  $x$  representa o deslocamento horizontal do carrinho, enquanto  $\theta$  indica o ângulo de inclinação do pêndulo em relação à vertical.

Figura 2 – Diagrama de corpo livre para o pêndulo invertido sobre o carro.



Fonte: [Iacob \(2020, p.2\)](#)

Adota-se um sistema de coordenadas no qual a direção frontal do corpo corresponde ao sentido positivo do eixo  $x$ , orientado horizontalmente em relação ao solo. A direção vertical para cima é definida como o sentido positivo do eixo  $z$ , conforme indicado no diagrama de corpo livre. O eixo  $y$ , perpendicular ao plano  $xz$ , representa a direção transversal ao movimento, coincidindo com o eixo de rotação das rodas. Neste trabalho, não há variação angular em torno do eixo  $y$ , sendo este mantido apenas como referencial auxiliar para aplicação da regra da mão direita na definição do plano  $xz$ , conforme a convenção descrita em ([IACOB, 2020](#)). Com base nesse referencial, o centro de gravidade do pêndulo, denominado  $CDG$ , apresenta as seguintes coordenadas:  $x$ ,  $z$  e  $y$ .

$$X_{CDG} = x + l \sin \theta \quad (2.1)$$

$$Y_{CDG} = 0 \quad (2.2)$$

$$Z_{CDG} = l \cos \theta \quad (2.3)$$

Aplica-se a segunda Lei de Newton para determinar o somatório de forças atuantes na direção horizontal ( $x$ ) do carrinho e o somatório de momentos exercidos no pêndulo em torno do seu ponto de articulação:

$$M\ddot{x} = u - H \quad (2.4)$$

$$I\ddot{\theta} = Vl \sin \theta - Hl \cos \theta \quad (2.5)$$

Na Equação (2.4),  $\ddot{x}$  representa a aceleração do carrinho,  $H$  corresponde à componente horizontal da força de reação exercida pelo pêndulo sobre o carrinho e possui sentido contrário a  $u$ . Já na Equação (2.5),  $I$  é o momento de inércia do pêndulo em relação ao ponto de articulação,  $\ddot{\theta}$  é a aceleração angular do pêndulo,  $V$  representa a componente vertical da força de reação no ponto de conexão com o carrinho. Os termos  $Vl \sin \theta$  e  $Hl \cos \theta$  correspondem, respectivamente, aos torques gerados pelas componentes vertical e horizontal das forças de reação, considerados em relação ao centro de massa do pêndulo e levando em conta sua inclinação angular  $\theta$ .

$$H = m\ddot{X}_{CDG} \quad (2.6)$$

$$V = m\ddot{Z}_{CDG} + mg \quad (2.7)$$

A partir das expressões das coordenadas do centro de gravidade do pêndulo, é possível obter suas acelerações ao calcular as derivadas segundas de  $X_{CDG}$  e  $Z_{CDG}$ . As componentes da aceleração nas direções horizontal e vertical são expressas por:

$$\ddot{X}_{CDG} = \ddot{x} + l(\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta) \quad (2.8)$$

$$\ddot{Z}_{CDG} = -l(\ddot{\theta} \sin \theta + \dot{\theta}^2 \cos \theta) \quad (2.9)$$

Substituindo as Equações (2.8) e (2.9) nas Equações (2.6) e (2.7), obtêm-se as forças de reação  $H$  e  $V$  no ponto de articulação:

$$H = m\ddot{x} + ml(\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta) \quad (2.10)$$

$$V = -ml(\ddot{\theta} \sin \theta + \dot{\theta}^2 \cos \theta) + mg \quad (2.11)$$

Essas expressões para  $H$  e  $V$  podem então ser substituídas nas equações que descrevem a dinâmica rotacional do pêndulo e a movimentação horizontal do carrinho, dadas respectivamente pelas Equações (2.5) e (2.4).

$$I\ddot{\theta} = [-ml(\ddot{\theta} \sin \theta + \dot{\theta}^2 \cos \theta) + mg] l \sin \theta - [m\ddot{x} + ml(\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta)] l \cos \theta \quad (2.12)$$

$$M\ddot{x} = u - [m\ddot{x} + ml(\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta)] \quad (2.13)$$

Após algumas manipulações algébricas, podemos obter as expressões que relacionam a aceleração angular do pêndulo e a aceleração linear do carrinho:

$$\ddot{\theta}(I + ml^2) = -ml \cos \theta \ddot{x} + mgl \sin \theta \quad (2.14)$$

$$\ddot{x}(M + m) = ml\dot{\theta}^2 \sin \theta - ml\ddot{\theta} \cos \theta + u \quad (2.15)$$

Isolando a variável  $\ddot{\theta}$  na Equação (2.14) e substituindo seu valor na Equação (2.15), obtém-se uma expressão que descreve a dinâmica da aceleração angular do pêndulo invertido sobre o carro.

$$\ddot{\theta} = \frac{-m^2 l^2 \dot{\theta}^2 \sin \theta \cos \theta + mgl(M + m) \sin \theta - ml \cos \theta u}{(I + ml^2)(M + m) - m^2 l^2 \cos^2 \theta} \quad (2.16)$$

Da mesma forma, podemos isolar o valor de  $\ddot{x}$  na Equação (2.15) e substituí-lo em (2.14) e obter a equação que define a aceleração linear do carrinho em função das variáveis do sistema:

$$\ddot{x} = \frac{ml(I + ml^2)\dot{\theta}^2 \sin \theta - m^2 l^2 g \cos \theta \sin \theta + u(I + ml^2)}{(I + ml^2)(M + m) - m^2 l^2 \cos^2 \theta} \quad (2.17)$$

As Equações (2.16) e (2.17) representam a dinâmica não linear do pêndulo invertido sobre o carro.

### 2.1.2 Linearização e Modelo em Espaço de Estados

A maioria dos sistemas físicos apresentam comportamentos não lineares, múltiplas variáveis de estado e, frequentemente, relações complexas entre suas entradas e saídas. O pêndulo invertido sobre rodas é um exemplo clássico desse tipo de sistema, cuja dinâmica é caracterizada por não linearidades, devido à interação entre a inclinação do pêndulo e o movimento translacional do corpo.

Conforme descrito por Ogata (2010), para analisar sistemas dessa natureza, é essencial reduzir a complexidade de suas expressões matemáticas e recorrer a computadores. Sob esse ponto de vista, a abordagem por espaço de estados se torna mais apropriada. De maneira geral, um sistema pode ser representado em espaço de estados, a partir das suas equações de dinâmica, na seguinte forma:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad (2.18)$$

Nessa representação:

- $x(t)$  é o vetor de estados do sistema;

- $\dot{x}(t)$  representa a derivada dos estados em relação ao tempo;
- $u(t)$  é o vetor de entradas;
- $y(t)$  é o vetor de saídas;

As matrizes  $A$ ,  $B$ ,  $C$  e  $D$  definem as relações internas do sistema, sendo que:

- $A$  descreve a dinâmica do sistema, isto é, como os estados evoluem ao longo do tempo;
- $B$  indica como as entradas influenciam essa dinâmica;
- $C$  define como os estados afetam a saída;
- $D$  representa o efeito direto das entradas sobre a saída, sem passar pela dinâmica do sistema.

Mas se tratando de equações de dinâmica não lineares, a representação em espaço de estados não segue esse formato devido ao fato de que as derivadas dos estados têm uma relação não linear com o próprio estado ou entradas de controle, (FRANKLIN et al., 2010). Uma forma de lidar com as não linearidades de sistemas em sua modelagem é considerar que para sinais pequenos, o sistema pode ser aproximado de um sistema linear. Essa redução de um sistema não linear para linear é denominada de linearização. Uma das técnicas de linearização conhecidas dentro da área de controle, baseia-se na expansão em série de Taylor das equações diferenciais que descrevem a dinâmica do sistema. Tal procedimento permite representar o sistema por um modelo linear cuja validade é restrita a regiões próximas a um ponto de operação definido, (KHALIL, 2002). Um sistema não linear é descrito por

$$\dot{x} = f(x(t), u(t)) \quad (2.19)$$

em que  $x(t) \in \mathbb{R}^n$  representa o vetor de estados,  $u(t) \in \mathbb{R}^m$  simboliza o vetor de entradas de controle, e  $f : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^n$  é uma função continuamente diferenciável e descreve a dinâmica do sistema ao longo do tempo. Suponha que este sistema possua um ponto de equilíbrio  $(x^*, u^*)$ , tal que  $f(x^*, u^*) = 0$ . Com o intuito de descrever o comportamento do sistema em torno deste ponto, definem-se variáveis de perturbação em relação aos valores de  $x$  e  $u$  como:

$$\delta x = x - x^*, \quad \delta u = u - u^*$$

$$\delta \dot{x} + \dot{x}^* \approx f(x^*, u^*) + \left. \frac{\partial f}{\partial x} \right|_{(x^*, u^*)} \delta x + \left. \frac{\partial f}{\partial u} \right|_{(x^*, u^*)} \delta u + \text{H.O.T}$$

Assumimos que a análise de interesse se restrinja a uma vizinhança suficientemente pequena de  $(x, u)$ , de forma que os termos de ordem superior (H.O.T.) sejam desprezados, portanto:

$$\delta \dot{x} = \left. \frac{\partial f}{\partial x} \right|_{(x^*, u^*)} \delta x + \left. \frac{\partial f}{\partial u} \right|_{(x^*, u^*)} \delta u$$

O sistema linearizado pode ser representado por:

$$\delta \dot{x} = A \delta x + B \delta u$$

em que as matrizes  $A$  e  $B$  são descritas por:

$$A = \left. \frac{\partial f}{\partial x} \right|_{(x^*, u^*)}, \quad B = \left. \frac{\partial g}{\partial u} \right|_{(x^*, u^*)}$$

As matrizes  $A$  e  $B$  são compostas pelas derivadas parciais da função que descreve a dinâmica do sistema em relação aos estados e às entradas, respectivamente, avaliadas no ponto de equilíbrio. Essa formulação permite obter um modelo linear que aproxima o comportamento do sistema nas proximidades do ponto de operação considerado.

Aplicando esse conceito ao pêndulo invertido sobre rodas, parte-se das Equações (2.16) e (2.17), que representam sua dinâmica não linear. O procedimento de linearização envolve a expansão em série de Taylor em torno do ponto de equilíbrio do sistema. Para facilitar esse desenvolvimento, as equações são reorganizadas por meio do agrupamento de constantes, o que evidencia os termos não lineares presentes na dinâmica. Assim, definem-se as seguintes constantes:

$$\begin{aligned} c_1 &= m^2 l^2, \\ c_2 &= mgl(M + m), \\ c_3 &= ml, \\ c_4 &= ml(I + ml^2), \\ c_5 &= m^2 l^2 g, \\ c_6 &= (I + ml^2), \\ d_1 &= (I + ml^2)(M + m), \\ d_2 &= m^2 l^2. \end{aligned}$$

A partir dessas constantes, as equações diferenciais não lineares que descrevem a aceleração angular do pêndulo e a aceleração linear do carro podem ser reescritas de forma mais compacta da seguinte maneira:

$$\ddot{\theta} = \frac{-c_1 \dot{\theta}^2 \sin \theta \cos \theta + c_2 \sin \theta - c_3 \cos \theta u}{d_1 - d_2 \cos^2 \theta} \quad (2.20)$$

$$\ddot{x} = \frac{c_4 \dot{\theta}^2 \sin \theta - c_5 \cos \theta \sin \theta + c_6 u}{d_1 - d_2 \cos^2 \theta} \quad (2.21)$$

As equações diferenciais não lineares expressas em (2.20) e (2.21) podem ser reorganizadas em termos das variáveis de estado, definidas por:

$$\theta = x_1, \dot{\theta} = x_2, x = x_3, \dot{x} = x_4.$$

Com base nessas variáveis, o modelo dinâmico é reescrito como um sistema de quatro equações diferenciais de primeira ordem.

$$\dot{x}_1 = x_2 \quad (2.22)$$

$$\dot{x}_2 = \frac{-c_1 x_2^2 \sin x_1 \cos x_1 + c_2 \sin x_1 - c_3 \cos x_1 u}{d_1 - d_2 \cos^2 x_1} \quad (2.23)$$

$$\dot{x}_3 = x_4 \quad (2.24)$$

$$\dot{x}_4 = \frac{c_4 x_2^2 \sin x_1 - c_5 \cos x_1 \sin x_1 + c_6 u}{d_1 - d_2 \cos^2 x_1} \quad (2.25)$$

A Equação (2.23) mostra que a velocidade angular do pêndulo depende diretamente do ângulo de inclinação. De forma análoga, a Equação (2.25) evidencia que a velocidade linear do carro está diretamente relacionada à sua posição. A partir dessa formulação, aplica-se o processo de linearização apresentado anteriormente, com o objetivo de obter uma representação linear aproximada do sistema nas proximidades do ponto de equilíbrio, que corresponde ao pêndulo na posição vertical e ao carro em repouso, ou seja,

$$x^* = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad u^* = 0.$$

A linearização é realizada calculando-se as derivadas parciais das equações diferenciais em relação às variáveis de estado e à entrada, avaliadas no ponto de equilíbrio. Resultando nas matrizes  $A$  e  $B$  do modelo linear em espaço de estados:

$$A = \begin{bmatrix} \frac{\partial \dot{x}_1}{\partial x_1} & \frac{\partial \dot{x}_1}{\partial x_2} & \frac{\partial \dot{x}_1}{\partial x_3} & \frac{\partial \dot{x}_1}{\partial x_4} \\ \frac{\partial \dot{x}_2}{\partial x_1} & \frac{\partial \dot{x}_2}{\partial x_2} & \frac{\partial \dot{x}_2}{\partial x_3} & \frac{\partial \dot{x}_2}{\partial x_4} \\ \frac{\partial \dot{x}_3}{\partial x_1} & \frac{\partial \dot{x}_3}{\partial x_2} & \frac{\partial \dot{x}_3}{\partial x_3} & \frac{\partial \dot{x}_3}{\partial x_4} \\ \frac{\partial \dot{x}_4}{\partial x_1} & \frac{\partial \dot{x}_4}{\partial x_2} & \frac{\partial \dot{x}_4}{\partial x_3} & \frac{\partial \dot{x}_4}{\partial x_4} \end{bmatrix}, \quad B = \begin{bmatrix} \frac{\partial \dot{x}_1}{\partial u} \\ \frac{\partial \dot{x}_2}{\partial u} \\ \frac{\partial \dot{x}_3}{\partial u} \\ \frac{\partial \dot{x}_4}{\partial u} \end{bmatrix} \quad (2.26)$$

A partir dessas matrizes (2.26), pode-se então obter a representação do pêndulo invertido sobre o carro em espaço de estados.

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ A_{21} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ A_{41} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ B_{21} \\ 0 \\ B_{41} \end{bmatrix} u \\ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \end{aligned} \quad (2.27)$$

em que:

$$\begin{aligned} A_{21} &= \frac{(M+m)mg l}{\Delta} \\ A_{41} &= -\frac{m^2 l^2 g}{\Delta} \\ B_{21} &= -\frac{ml}{\Delta} \\ B_{41} &= \frac{I + ml^2}{\Delta} \\ \Delta &= I(M+m) + Mml^2 \end{aligned}$$

## 2.2 Modelagem fuzzy

Quando lidamos com sistemas reais, notamos que seu comportamento pode diferir do que se é analisado teoricamente. Quanto mais complexo o sistema for, mais inexata ou imprecisa é a informação que utilizamos para caracterizá-lo (ROSS, 2009). Como consequência, a aplicação direta de técnicas de controle tradicionais, que dependem de modelos precisos, torna-se limitada ou até inviável em alguns casos.

Uma alternativa para descrever a dinâmica de sistemas complexos é a utilização da modelagem *fuzzy*. Trata-se de uma técnica baseada na lógica *fuzzy* (ou lógica nebulosa), a qual permite lidar com incertezas e não linearidades por meio da aproximação de comportamentos dinâmicos complexos utilizando descrições linguísticas e qualitativas. Em contraste com modelos matemáticos precisos que exigem equações diferenciais completas, a modelagem fuzzy estrutura o conhecimento do sistema com base em regras heurísticas do tipo “SE-ENTÃO”, associando regiões do espaço de estado a comportamentos específicos do sistema.

Cada regra fuzzy é composta por duas partes fundamentais: o antecedente, que representa a condição da regra, e o conseqüente, que expressa a ação correspondente. Por

exemplo, no caso do pêndulo invertido, pode-se representar o comportamento do sistema com regras simples, por exemplo:

$R_1$  : “SE inclinação é positiva, ENTÃO o torque aplicado deve ser negativo”.

Nesse exemplo, tanto no antecedente da regra (SE) quanto no conseqüente (ENTÃO) são utilizadas variáveis linguísticas para descrever qualitativamente os estados e ações do sistema. Uma variável linguística é uma variável cujo valor não é numérico, mas sim uma expressão verbal, como “positivo”, “negativo”, “próximo de zero”, “alto”, “baixo”, entre outros. Cada variável linguística é definida sobre um universo de discurso, que corresponde ao conjunto de todos os valores reais possíveis e relevantes que essa variável pode assumir dentro do contexto do sistema.

Cada valor verbal atribuído à variável linguística é denominado termo linguístico. Por exemplo, se considerarmos a variável linguística “velocidade angular”, os termos linguísticos poderiam ser “negativa”, “nula” ou “positiva”. Cada um desses termos é representado matematicamente por uma função de pertinência, a qual define, para cada valor numérico do universo de discurso, o grau com que esse valor pertence ao termo linguístico em questão.

Para ilustrar esse conceito, considere  $X$  como o universo de discurso da variável linguística “velocidade angular”, ou seja, o conjunto de todos os valores reais relevantes que essa variável pode assumir durante o funcionamento do pêndulo invertido. Suponha agora que  $A \subseteq X$  represente o subconjunto dos valores considerados “positivos” da velocidade angular. Na teoria de conjuntos clássicos, o pertencimento de um elemento  $x \in X$  ao conjunto  $A$  é determinado por uma função indicadora  $\mu_A(x)$ , que assume apenas dois valores:

$$\mu_A(x) = \begin{cases} 1, & \text{se } x \in A \\ 0, & \text{se } x \notin A \end{cases}$$

Ou seja, o pertencimento de um elemento ao conjunto é binário: ele pertence totalmente ou não pertence. A teoria de conjuntos *fuzzy* apresenta uma extensão da teoria clássica de conjuntos. Zadeh (1965), propõe o pertencimento de um elemento  $x$  descrito por uma função de pertinência  $\mu_A(x)$ , definida como:

$$\mu_A : X \rightarrow [0, 1]$$

Essa função atribui a cada elemento  $x \in X$  um grau de pertencimento a um conjunto fuzzy  $A$ . Ao invés de uma decisão rígida,  $\mu_A(x)$  pode assumir qualquer valor real

entre 0 e 1, refletindo o quanto aquele elemento pertence ao conjunto. Formalmente, o conjunto *fuzzy*  $A$  é definido por:

$$A = \{(x, \mu_A(x) | x \in X\}$$

Por exemplo, considere o universo de discurso da velocidade angular do pêndulo invertido  $X$  e o subconjunto (conjunto *fuzzy*)  $A$  da velocidade angular considerada “alta” para o sistema. Um valor como  $x = 10$  rad/s pode ter  $\mu_A(x) = 0,7$ , indicando pertencimento relativamente alto ao conjunto  $A$ , enquanto uma velocidade angular de  $x = 1$  rad/s pode apresentar  $\mu_A(x) = 0,1$ , refletindo um pertencimento baixo ao conjunto de velocidades “alta”. Isso mostra que, quanto mais distante do centro do conceito, menor é o grau de pertencimento. Assim como a teoria de conjuntos clássicos, existem operações matemáticas que estabelecem relações entre conjuntos *fuzzy* definidos (JANG; SUN, 1996). Considere dois conjuntos *fuzzy*  $A$  e  $B$  no universo de discurso  $X$ :

1. Inclusão ou subconjunto:

Um conjunto fuzzy  $A$  está contido em  $B$ , se e somente se  $\mu_A \leq \mu_B$  para qualquer valor de  $x$ :

$$A \subseteq B \iff \mu_A \leq \mu_B$$

2. União

A união de dois conjuntos *fuzzy*  $A$  e  $B$  resulta é um conjunto *fuzzy*  $C$ , expressado por  $C = A \cup B$ , cuja função de pertinência de  $A$  e  $B$  são relacionadas por:

$$\mu_C(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x)$$

A operação max especifica que o grau de pertencimento  $x$  a união de  $A$  e  $B$  é o maior grau de pertencimento de  $x$  a qualquer um dos conjuntos.

3. Interseção:

A interseção de dois conjuntos *fuzzy*  $A$  e  $B$  resulta em um conjunto  $C$ , expressado por  $C = A \cap B$ , cuja função de pertinência relaciona os conjuntos  $A$  e  $B$  por:

$$\mu_C(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x)$$

De forma análoga a operação de união, a operação min é utilizada para especificar o menor grau de pertinência de  $x$  como medida do quanto um elemento pertence simultaneamente a  $A$  e  $B$ .

## 4. Complemento:

Dado o conjunto *fuzzy*  $A$ , seu complemento expressado por  $\bar{A}$  é definido como:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

Em geral, as funções de pertinência são definidas de modo a refletir a natureza gradual da transição entre diferentes regiões do universo de discurso. A forma dessas funções pode ser definida seguindo formulações matemáticas.

- Funções triangulares são definidas utilizando três parâmetros  $(a, b, c)$ , cada parâmetro descreve a coordenada  $x$  dos três vértices do triângulo:

$$\mu_{triangular}(x; a, b, c) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ \frac{c-x}{c-b}, & b \leq x \leq c. \\ 0, & c \leq x \end{cases}$$

- Funções trapezoidais são definidas utilizando quatro parâmetros  $(a, b, c, d)$  com  $(a < b \leq c < d)$  e estes determinam as coordenadas  $x$  dos quatro cantos de um trapézio. Formalmente é definida por:

$$\mu_{trapezoidal}(x; a, b, c, d) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d. \\ 0, & d \leq x \end{cases}$$

Devido a suas formulações simples, tanto funções de pertinência triangulares e trapezoidais são bastante usadas em implementações de sistemas reais, mas é possível notar que tais funções são lineares por partes, o que pode levar a alterações bruscas no processamento de informações. Existem outras funções que são suaves porém não-lineares, como por exemplo:

- Funções Gaussianas, são especificadas por dois parâmetros  $(c, \sigma)$ , onde  $c$  representa o centro da função e  $\sigma$  representa a largura, são expressas por:

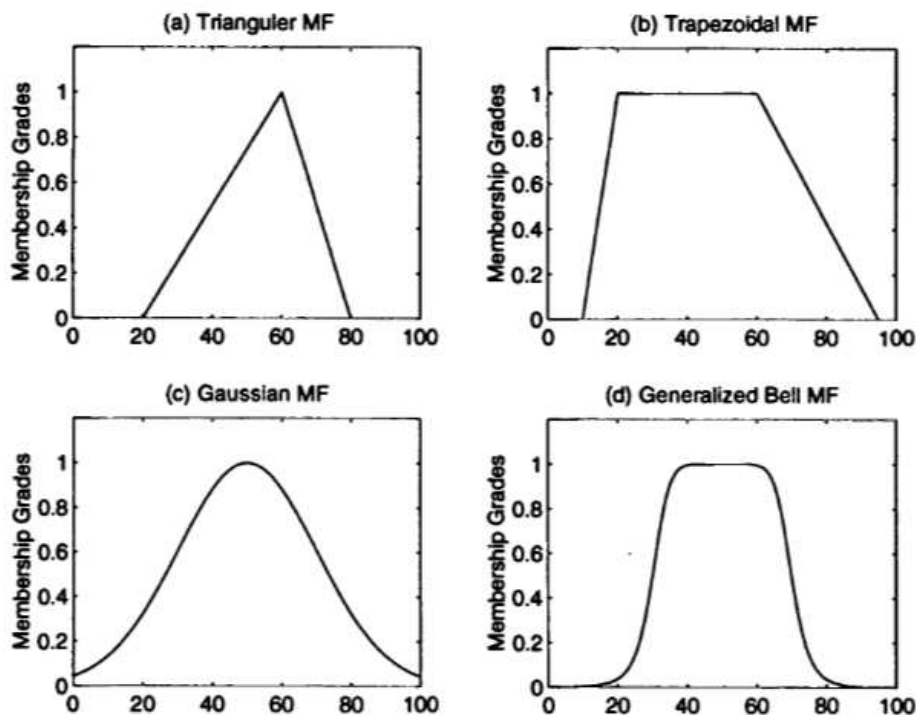
$$\mu_{gaussiana}(x, c, \sigma) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2}$$

- Funções “sinos”, são definidas por três parâmetros  $(a, b, c)$  em que o parâmetro  $b$  é normalmente positivo, se  $b$  for negativo a função parecerá um sino invertido, com abertura para cima.  $c$  e  $a$  variam o centro e largura da função e  $b$  é usado para controlar as inclinações nos pontos de cruzamentos.

A [Figura 3](#) ilustra graficamente os principais tipos de funções de pertinência discutidos, considerando exemplos unidimensionais. Cada uma delas é parametrizada para demonstrar suas diferentes formas e comportamentos. Embora os exemplos apresentados sejam para uma única variável, essas funções podem ser estendidas para domínios multidimensionais, mantendo a lógica subjacente.

[Babuška \(2012\)](#) explica que a definição de qual função utilizar pode ser baseada em conhecimento prévio do desenvolvedor do modelo, baseada em experimentação e ajustada empiricamente, e baseada em dados, construídas ou adaptadas a partir de dados de entrada-saída do sistema. Para sistemas de controle em tempo real, [Moness, Mahmoud e Hussein \(2022\)](#) citam que quanto menor o número de funções de pertinência, mais rápida será a velocidade de computação exigida e quanto mais estreita for o formato da função de pertinência, mais abruptas são as mudanças no sinal de controle.

Figura 3 – Exemplos de funções de pertinência parametrizadas: (a)  $\text{triangular}(x; 20, 60, 80)$ ; (b)  $\text{trapezoidal}(x; 10, 20, 60, 95)$ ; (c)  $\text{gaussiana}(x; 50, 20)$ ; (d)  $\text{sino}(x; 20, 4, 50)$ .



Fonte: [Jang e Sun \(1996, p.26\)](#)

Os conceitos de conjuntos *fuzzy* e da lógica nebulosa descritos até o momento, podem ser utilizados para modelar sistemas de diversas formas. [Babuška \(2012\)](#) cita que

os sistemas *fuzzy* podem ser desenvolvidos através de regras *fuzzy*, através de regressões lineares *fuzzy* ou modelos *fuzzy* utilizando estruturas celulares. Neste trabalho, o foco está em utilizar o modelo *fuzzy* baseado em regras SE-ENTÃO.

Considera-se um sistema com  $m$  variáveis de entrada, denotadas por  $x_1, x_2, \dots, x_m$  e uma variável de saída  $\gamma$ . As entradas podem representar estados ou grandezas relacionadas ao sistema e a saída  $\gamma$  representa uma resposta do sistema *fuzzy*. Dependendo da forma como o consequente da regra é definido, diferentes modelos *fuzzy* podem ser estabelecidos. De acordo com (SUGENO, 1999), destacam-se três principais categorias:

- Modelo *Fuzzy Linguístico*, ((ZADEH, 1973); (MAMDANI, 1977)); em que a regra é associada a termos linguísticos como:

$$\begin{aligned} \text{Regra } R_i : \text{ SE } x_1 \text{ é } M_1^i, x_2 \text{ é } M_2^i, \dots, x_m \text{ é } M_m^i, \\ \text{ ENTÃO } \gamma \text{ é } H^i, \quad i = 1, 2, \dots, n \end{aligned}$$

em que  $x_m$  representam as variáveis de entradas,  $\gamma$  representa a saída,  $H^i$  e  $M_j^i$ ,  $j = 1, 2, \dots, m$ , são conjuntos *fuzzy* e  $n$  representa o número de regras.

- Modelo Takagi-Sugeno, (TAKAGI; SUGENO, 1985), em que o consequente é uma função das variáveis antecedentes:

$$\begin{aligned} \text{Regra } R_i : \text{ SE } x_1 \text{ é } M_1^i, x_2 \text{ é } M_2^i, \dots, x_m \text{ é } M_m^i, \\ \text{ ENTÃO } \gamma \text{ é } f^i(x_1, x_2, \dots, x_m), \quad i = 1, 2, \dots, n \end{aligned}$$

A função  $f^i$  normalmente é representada por uma função linear:

$$f^i(x_1, x_2, \dots, x_m) = b^i + a_1^i x_1 + a_2^i x_2 + \dots + a_m^i x_m$$

Os coeficientes  $b^i, a_j^i$  com  $j = 1, \dots, m$  são termos constantes.

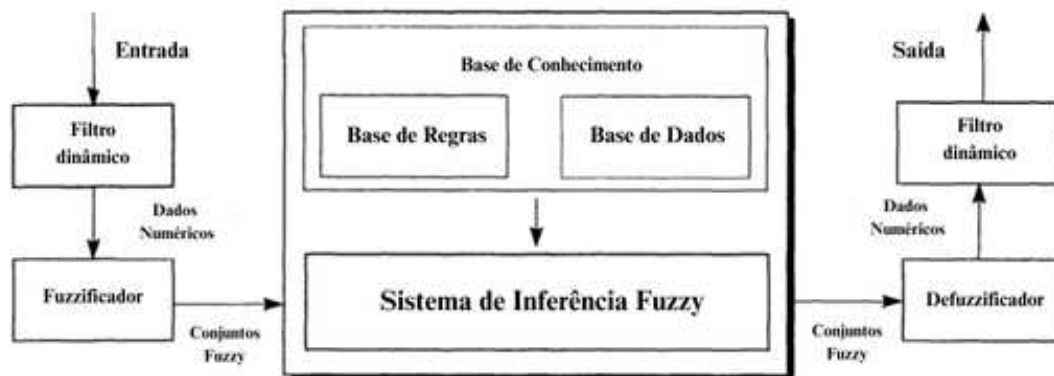
- Modelo *Fuzzy Singleton* (SUGENO, 1999). Nesse tipo de sistema, as regras são desenvolvidas da seguinte forma:

$$\begin{aligned} \text{Regra } R_i : \text{ SE } x_1 \text{ é } M_1^i, x_2 \text{ é } M_2^i, \dots, x_m \text{ é } M_m^i, \\ \text{ ENTÃO } \gamma \text{ é } b^i, \quad i = 1, 2, \dots, n \end{aligned}$$

em que  $b^i$  é um *singleton*, isso significa que o consequente da regra desse sistema é um valor numérico fixo.

Até este momento, foram apresentados os principais elementos que compõem um sistema *fuzzy*: variáveis linguísticas, funções de pertinência e estruturas de regras SE-ENTÃO. No entanto, para que um sistema fuzzy possa ser efetivamente utilizado em aplicações práticas de controle ou tomada de decisão, é necessário compreender como essas regras são processadas em tempo real a partir de entradas numéricas. Esse processo é realizado por meio de três etapas: fuzzificação, inferência e defuzzificação. A ilustração de como essas etapas são interligadas pode ser vista na [Figura 4](#).

Figura 4 – Exemplo de Sistema Fuzzy Genérico.



Fonte: (Adaptado de [Babuška \(2012\)](#), p.11)

A etapa de fuzzificação consiste na conversão dos valores numéricos provenientes do sistema, como medições dos estados, em valores *fuzzy*. Esses valores correspondem ao grau de pertencimento da variável medida aos conjuntos *fuzzy* que foram definidos. Considere uma entrada escalar  $x \in X$ , pertencente ao universo de discurso  $X$ . Sejam  $A_1, A_2, \dots, A_k$  os conjuntos fuzzy definidos sobre  $X$ , com respectivas funções de pertinência  $\mu_{A_j}(x)$ , para  $j = 1, 2, \dots, k$ . A fuzzificação consiste em calcular,

$$\mu_{A_j}(x), \quad \text{para } j = 1, 2, \dots, k$$

ou seja, o valor de  $x$  passa a ser representado por um vetor de graus de pertencimento aos conjuntos *fuzzy*, como por exemplo:

$$\mu_x = [\mu_{A_1}(x) \quad \mu_{A_2}(x) \quad \dots \quad \mu_{A_k}(x)]$$

Após a fuzzificação, ocorre o processo de inferência *fuzzy*, responsável por ativar as regras “SE-ENTÃO” com base nos graus de pertencimento calculados nas entradas. Considere um conjunto de  $n$  regras *fuzzy* do tipo:

$$R_i : \text{SE } x_1 \text{ é } A_1^i \text{ E } x_2 \text{ é } A_2^i \text{ ENTÃO } \gamma \text{ é } B^i, \quad i = 1, 2, \dots, n$$

A ativação (ou grau de disparo) da  $i$ -ésima regra depende dos graus de pertinência das entradas aos termos do antecedente. Para com operadores do tipo “E”, a ativação pode ser feita utilizando operações de mínimo. O Operador mínimo (Mamdani) determina que o grau de ativação da regra é dado pelo menor valor entre os graus de pertinência das variáveis de entrada. Essa operação representa a interseção *fuzzy* dos antecedentes, conforme discutido anteriormente.

$$\alpha_i = \min\{\mu_{A_1^i}(x_1), \mu_{A_2^i}(x_2), \dots, \mu_{A_m^i}(x_m)\}$$

O valor de  $\alpha_i$  é usado para compor a saída fuzzy correspondente ao conseqüente da regra, ponderando o conjunto *fuzzy*  $B^i$  (no modelo Mamdani), ou o valor numérico/função (no modelo Sugeno). A última etapa é a defuzzificação, responsável por converter o resultado *fuzzy* obtido no processo de inferência em uma saída numérica precisa, que pode ser aplicada em um sistema físico. Nos sistemas Mamdani, o resultado da inferência é um conjunto *fuzzy* agregado, e a defuzzificação consiste em extrair um valor escalar representativo. Um dos métodos muito utilizados em sistemas do tipo Mamdani é o Centro de Gravidade (centróide). Esse método é expresso pela equação a seguir.

$$x^* = \frac{\int_X x \mu_B(x) dx}{\int_X \mu_B(x) dx}$$

Com base numa função de pertinência  $\mu_B(x)$  agregada a saída *fuzzy* após a aplicação de todas as regras, o numerador  $\int_X x \mu_B(x) dx$  calcula o quanto a distribuição da pertinência está deslocada em relação à origem e o denominador  $\int_X \mu_B(x) dx$  calcula a “massa total” dessa distribuição que corresponde a área sobre a curva de  $\mu_B(x)$ .

Em modelos onde o conseqüente da regra não é expresso por um conjunto *fuzzy* (Modelo Takagi-Sugeno ou *Singleton*), o método da Média Ponderada pode ser utilizado. Trata-se de uma abordagem em que cada função de pertinência da saída é representada por um valor característico (geralmente o centróide da função), e esse valor é ponderado pelo grau máximo de ativação da respectiva regra, conforme:

$$x^* = \frac{\sum \mu_B(\bar{x}) \cdot \bar{x}}{\sum \mu_B(\bar{x})}$$

em que  $\mu_B(\bar{x})$  representa o grau máximo de ativação da função de pertinência de saída e  $\bar{x}$  é o centróide da função de pertinência considerada. Este método é o mais comum de ser utilizado em aplicações devido à sua eficiência computacional (ROSS, 2009), porém é restrito a funções de pertinência simétricas na saída.

Em resumo, o funcionamento de um sistema *fuzzy* baseia-se na conversão de informações exatas em representações linguísticas (fuzzificação), na ativação de regras *fuzzy* segundo o grau de pertinência das entradas (inferência) e na geração de uma saída precisa por meio da defuzzificação. Essas etapas viabilizam o uso da lógica *fuzzy* em contextos de controle automático, especialmente em sistemas com complexidades matemáticas.

Em [Nguyen et al. \(2019\)](#), os autores citam que os três tipos de sistemas *fuzzy* são capazes de lidar com funções não lineares através de aproximações generalizadas; no entanto, o modelo T-S, quando comparado aos outros dois modelos, é capaz de reduzir o número de regras *fuzzy* estabelecidas, permitindo melhor capacidade computacional no contexto de controle em tempo real, especialmente para sistemas mais complexos. Neste trabalho, o objetivo é projetar um controlador *fuzzy* a partir de um modelo Takagi-Sugeno devido à sua capacidade de representar com certa precisão o comportamento dinâmico de sistemas não lineares através de poucas regras *fuzzy*. A abordagem adotada será explorada com mais detalhes na próxima subseção.

### 2.2.1 Modelo fuzzy Takagi-Sugeno

O modelo *fuzzy* Takagi-Sugeno foi proposto por [Takagi e Sugeno \(1985\)](#) e é descrito por regras “SE-ENTÃO” que representam relações de entrada e saída de sistemas não lineares. O principal aspecto desse modelo é expressar dinâmicas locais para cada implicação *fuzzy* através de um modelo linear e unir cada modelo obtido com o intuito de descrever o sistema geral. Portanto, podemos dizer que, dada uma planta não linear de tempo contínuo, podemos descrever seu comportamento através da seguinte regra.

$$R_i : \mathbf{SE} \ z_1(t) \text{ é } M_{i1} \text{ e } \dots \text{ e } z_p(t) \text{ é } M_{ip},$$

$$\mathbf{ENTÃO} \ \begin{cases} \dot{x}(t) = A_i x(t) + B_i u(t), \\ y(t) = C_i x(t), \end{cases} \quad i = 1, 2, \dots, r.$$

Na estrutura mostrada,  $M_{ij}$  é um conjunto *fuzzy* e  $r$  é o número de regras para cada modelo;  $x(t) \in \mathbb{R}^n$  é o vetor de estados,  $u(t) \in \mathbb{R}^m$  é o vetor de entrada de controle,  $y(t) \in \mathbb{R}^q$  é o vetor de saídas,  $A_i \in \mathbb{R}^{n \times n}$ ,  $B_i \in \mathbb{R}^{n \times m}$ ,  $C_i \in \mathbb{R}^{q \times n}$ ;  $z_1(t), \dots, z_p(t)$  representam as variáveis premissa e podem ser funções das variáveis de estado, de perturbações externas, do tempo ou de uma combinação desses fatores ([TANAKA; WANG, 2004](#)).

Na literatura, existem duas metodologias que podem ser aplicadas para a construção desse modelo, são elas: identificação (modelagem *fuzzy*) utilizando dados de entrada e saída do sistema e derivar o modelo através de um sistema de equações não lineares. Uma das abordagens para obtenção do modelo *fuzzy* T-S (Takagi-Sugeno) através de um modelo dinâmico de um sistema é a utilização de aproximações locais em espaços de partição *fuzzy*.

A premissa dessa abordagem é que, embora a dinâmica do sistema seja globalmente não linear, ela pode ser bem representada por modelos lineares em torno de diferentes pontos de operação. Esses pontos são distribuídos ao longo da variável premissa e permitem a criação de modelos lineares locais que, quando combinados, representam o comportamento da planta em toda a faixa de operação.

A interpolação (ou combinação) consiste em ponderar dinamicamente os modelos locais com base em pesos obtidos a partir das funções de pertinência. Esses pesos variam continuamente de acordo com a variável de premissa, resultando em uma transição entre modelos. Se consideramos  $z(t)$  como a variável premissa de determinado sistema não linear e que representa um dos estados deste sistema. Considerando diferentes pontos de operação deste estado, podemos utilizar a técnica de linearização descrita na [Subseção 2.1.2](#) de forma a obter os modelos lineares locais de tempo contínuo na seguinte forma.

$$\dot{x}(t) = A_i x(t) + B_i u(t)$$

em que  $A_i$  e  $B_i$  são matrizes obtidas a partir de diferentes pontos de operação de  $z(t)$ . Cada modelo obtido será então associado a uma regra fuzzy:

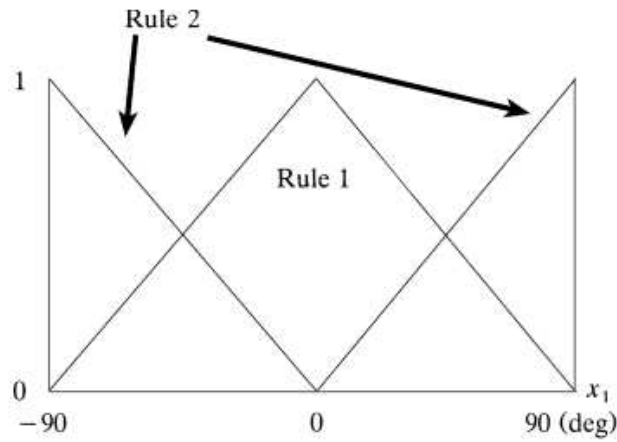
$$R_i : \text{SE } z_j(t) \text{ é } M_{ij}, \text{ ENTÃO } \dot{x}(t) = A_i x(t) + B_i u(t)$$

Para cada conjunto *fuzzy*  $M_{ij}$  que compõe a premissa da  $i$ -ésima regra (associada à  $j$ -ésima variável premissa), define-se uma função de pertinência  $\mu_{M_{ij}}$ . A etapa de fuzzificação em modelos T-S ocorre quando a função quantifica o grau em que a variável premissa  $z_j(t)$  pertence a cada região *fuzzy* definida no espaço de entrada. Cada função de pertinência mapeia a variável premissa no intervalo  $[0, 1]$ , permitindo que múltiplas regras sejam ativadas simultaneamente em função do valor de  $z(t)$ . Para ilustrar esse processo, a [Figura 5](#) mostra um exemplo das funções de pertinência triangulares utilizadas na modelagem *fuzzy* de um sistema pêndulo invertido.

Na figura, observa-se como a transição entre os modelos locais ocorre conforme a variável de premissa se desloca no espaço de entrada. No caso do sistema pêndulo invertido, esse espaço corresponde à inclinação do pêndulo na região de  $-90^\circ$  a  $90^\circ$ . A etapa de inferência consiste na determinação do grau de ativação de cada regra fuzzy, denotado por  $w_i(z(t))$ . Esse valor expressa o quanto a regra é relevante para o estado atual do sistema e é obtido pela combinação dos graus de pertinência de cada variável premissa associada a essa regra. O grau de ativação  $w_i(z(t))$  é calculado por:

$$w_i(z(t)) = \prod_{j=1}^p \mu_{M_{ij}}(z_j(t))$$

Figura 5 – Funções de pertinência associadas às duas regras fuzzy do sistema pêndulo invertido.



Fonte: Tanaka e Wang (2004, p.25)

onde  $p$  representa o número de variáveis premissa envolvidas na  $i$ -ésima regra. O peso de ativação da  $i$ -ésima regra ( $w_i(z(t))$ ) é calculado multiplicando os graus de pertinência de cada variável premissa  $z_j(t)$  a cada um dos conjuntos *fuzzy* ( $M_{ij}$ ) que compõem a premissa dessa regra. Se a  $i$ -ésima regra tiver partes na sua premissa (envolvendo  $p$  variáveis de premissa), então  $p$  graus de pertinência serão multiplicados para obter o peso de ativação dessa regra. Essa multiplicação garante que a regra só terá peso de ativação significativo se todas as suas condições de premissa forem satisfeitas em um certo grau.

Na etapa de inferência é feita a normalização dos pesos obtidos para cada regra. Essa normalização é representada por  $h_i(z(t))$ . Os pesos normalizados indicam a influência relativa de cada modelo linear local na composição do modelo Takagi-Sugeno global, em função do valor atual da variável premissa. A normalização é realizada dividindo o peso de ativação de cada regra pela soma de todos os pesos de ativação, conforme a expressão:

$$h_i(z(t)) = \frac{w_i(z(t))}{\sum_{j=1}^r w_j(z(t))}, \quad i = 1, 2, \dots, r$$

Conforme descrito em Nguyen et al. (2019), as funções de pertinência normalizadas atendem à propriedade de soma convexa:

$$0 \leq h_i(z(t)) \leq 1, \quad \sum_{i=1}^r h_i(z(t)) = 1$$

A interpolação entre os submodelos locais permite que o modelo T-S *fuzzy* seja descrito conforme.

$$\dot{x}(t) = \sum_{i=1}^r h_i(z(t))(A_i x(t) + B_i u(t)) \quad (2.28)$$

## 2.2.2 Compensação Paralela Distribuída

O PDC (*Parallel Distributed Compensation*) é uma técnica de controle baseada em um modelo *fuzzy* Takagi-Sugeno. A formulação desta técnica foi discutida em (SUGENO; KANG, 1986), no qual os autores sugerem a associação de cada regra de um sistema *fuzzy* T-S a uma regra de controle. As etapas de projeto de controle e análise de estabilidade de sistemas a partir da ação de controle foram apresentadas em (WANG; TANAKA; GRIFFIN, 1995) e (NIEMANN et al., 1999), respectivamente. De forma geral, as regras de controle são descritas da seguinte forma:

$$\begin{aligned} R_i : \text{SE } z_1(t) \text{ é } M_{i1} \text{ e } \dots z_p(t) \text{ é } M_{ip}, \\ \text{ENTÃO } u(t) = -F_i x(t), \quad i = 1, 2, \dots, r. \end{aligned}$$

Observa-se que o controlador projetado depende diretamente do número  $r$  de regras estabelecidas para o modelo *fuzzy* Takagi-Sugeno e compartilha os mesmos conjuntos *fuzzy* nas suas premissas (NIEMANN et al., 1999). No consequente de cada regra,  $F_i$  é uma matriz de realimentação de estados dos subsistemas lineares, obtidos por meio da linearização em distintos pontos de operação (TANAKA; WANG, 2004). O controlador após o processo de inferência e defuzzificação é representado por:

$$u(t) = -\frac{\sum_{i=1}^r w_i(z(t))F_i x(t)}{\sum_{i=1}^r w_i(z(t))} = -\sum_{i=1}^r h_i(z(t))F_i x(t) \quad (2.29)$$

De forma análoga ao modelo *fuzzy* Takagi-Sugeno, o peso de ativação  $w_i(z(t))$  de cada regra de controle é determinado pela saída da variável de premissa  $z(t)$ . A função  $h_i(t)$  desempenha o papel de normalizar esses pesos de ativação, assegurando que sua soma total seja unitária. Ao substituir (2.29) em (2.28), obtemos a dinâmica do sistema em malha fechada.

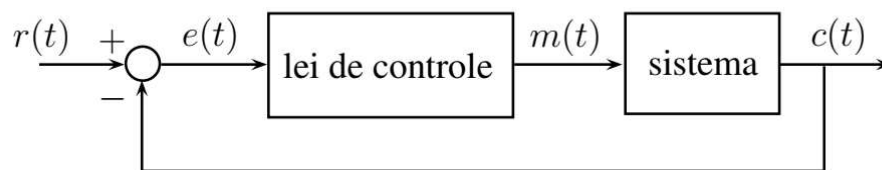
$$\begin{aligned} \dot{x}(t) &= \sum_{i=1}^r h_i(z(t))(A_i x(t) + B_i(-\sum_{j=1}^r h_j(z(t))F_j x(t))) \\ \dot{x}(t) &= \sum_{i=1}^r \sum_{j=1}^r h_i(z(t))h_j(z(t))\{A_i - B_i F_j\}x(t) \end{aligned}$$

O principal desafio do projeto de controle *fuzzy* PDC é determinar os ganhos de realimentação locais  $F_i$ . A obtenção desses ganhos pode ser realizada utilizando técnicas clássicas para projeto de controladores, como: alocação de polos ou LQR. Embora o sistema seja modelado localmente por submodelos, a estabilidade do sistema como um todo depende de que os ganhos locais garantam condições de estabilidade globais, (TANAKA; WANG, 2004).

## 2.3 Sistemas de controle amostrado

Com a evolução dos computadores digitais, sistemas de controle que anteriormente eram implementados utilizando dispositivos eletrônicos analógicos passaram a utilizar microprocessadores/microcontroladores que possuem tecnologia de *hardware* necessária para processamento de informações do sistema. Para ilustrar isso, podemos considerar o que é descrito por Aguirre (2023). Considere o diagrama de blocos de um sistema com realimentação negativa mostrado na Figura 6.

Figura 6 – Diagrama de Blocos de um Servomecanismo de tempo contínuo.



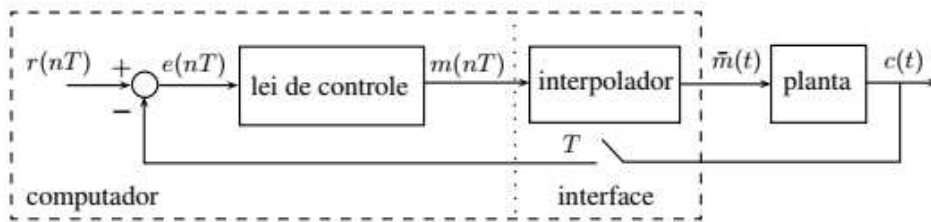
Fonte: (Aguirre (2023), p.13)

O funcionamento deste sistema baseia-se na contínua comparação da saída, ou variável controlada, denominada por  $c(t)$ , com a referência  $r(t)$ . Essa diferença entre a referência e o valor da saída expressa o erro  $e(t)$  e existe para todo e qualquer valor de  $t$ . A partir deste sinal de erro, a lei de controle determina  $m(t)$ , que representa o sinal ou ação de controle. No entanto, quando os sistemas de controle são implementados com dispositivos digitais, essa dinâmica contínua é substituída por uma lógica que só admite ter informações do sinal de erro em instantes regularmente espaçados. Como exemplo, podemos citar a leitura de posição de um avião por um radar, a temperatura em um forno industrial lida por um termopar, a velocidade de um veículo através de encoders, entre outros.

A forma mais comum de gerar sinais que são discretos no tempo é por meio da amostragem da variável que existe continuamente ao longo do tempo (AGUIRRE, 2023). O valor de saída do sistema só pode ser comparado com a referência a cada período de amostragem, indicado por  $T_s$ , gerando uma sequência possivelmente infinita, mas contável dos valores de erro  $e(T_s)$ ,  $e(2T_s)$ ,  $e(3T_s)$  e assim por diante. O sistema de controle, nesse caso, passa a ser representado pelo diagrama apresentado na Figura 7. Note que não apenas a lei de controle deve ser capaz de determinar o sinal a partir da sequência de valores de erro, como também essa ação é uma sequência de valores  $m(nT_s)$ ,  $n = 1, 2, \dots$ . O sistema de controle, portanto, passa a funcionar com uma parte que opera com sequências discretas, ao passo que outra parte é de tempo contínuo.

É importante afirmar que a informação dos sistemas físicos controlados é, por natureza, analógica, representando grandezas como temperatura, velocidade, pressão,

Figura 7 – Diagrama de Blocos de um sistemas de controle amostrado

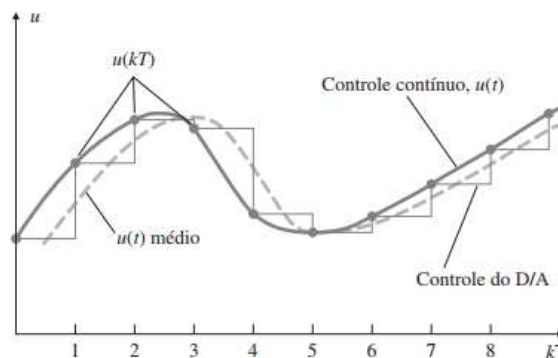


Fonte: (Aguirre (2023), p.14)

tensão e corrente elétrica. Somente após passar por um processo de conversão analógica-digital (AD) essa informação assume forma digital. Na prática, o conversor AD realiza dois processos: primeiro, amostra a variável física em instantes discretos no tempo e, em seguida, quantiza sua amplitude para um conjunto finito de valores, codificados em binário. A quantidade de níveis que o conversor pode representar está diretamente relacionada ao número de bits; por exemplo, um conversor de 10 bits permite até  $2^{10} = 1024$  níveis distintos. Esse processo de discretização está indicado na Figura 7 pelo símbolo de chave no bloco de interface.

O sinal de controle calculado com base na sequência de valores discretos do erro, precisa ser convertido novamente para um sinal contínuo antes de ser aplicado ao sistema físico. Esse processo é conhecido como interpolação, cuja implementação prática é feita pelo conversor digital-analógico (DA), representado na Figura 7 pelo bloco interpolador. O conversor DA mais utilizado em implementações é o segurador de ordem zero (ZOH - *Zero Order Holder*). Nesse tipo de conversor, o valor binário é convertido em analógico e esse valor durante o período de amostragem. Esse processo pode ser visualizado na Figura 8, em que o sinal de controle discreto é representado por  $u(kT_s)$ , com  $k = 1, 2, \dots$  e  $u(t)$  é o sinal convertido.

Figura 8 – Sinal gerado pelo segurador de ordem zero.



Fonte: (Franklin, Powell e Workman (2022), p.14)

Para que o controlador digital possa operar corretamente, é necessário que o modelo dinâmico do sistema a ser analisado seja também convertido para uma forma compatível com o domínio discreto. Existem diferentes métodos matemáticos capazes de discretizar sistemas de tempo contínuo, podemos citar: o método de aproximações por integrais numéricas, o método de Tustin, o método de Euler e suas variantes.

Para um modelo representado em espaço de estados, conforme a Equação (2.18), o método de discretização de Euler para frente pode ser aplicado. Esse método pode ser definido da seguinte forma, considere a definição da derivada de uma função no tempo contínuo:

$$\dot{x}(t) = \frac{dx(t)}{dt}$$

No contexto da discretização, essa derivada pode ser aproximada por uma diferença entre dois instantes de tempo consecutivos, separados por um intervalo igual ao período de amostragem  $T_s$  (FRANKLIN; POWELL; WORKMAN, 2022), tal que:

$$\dot{x}(k) \approx \frac{x((k+1)T_s) - x(kT_s)}{T_s}.$$

Substituindo essa relação na Equação (2.18), chegamos à seguinte representação do modelo em espaço de estados discreto.

$$\begin{aligned} x(k+1) &= (I + AT_s)x(k) + BT_s u(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{2.30}$$

A representação expressa pela Equação (2.30) pode ser simplificada, considerando que a discretização resulte em matrizes de domínio de tempo discreto, expressas por  $A_d$ ,  $B_d$ ,  $C_d$  e  $D_d$ . Dessa forma, a representação pode ser reescrita.

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k) \\ y(k) &= C_d x(k) + D_d u(k) \end{aligned}$$

O método de Euler para frente é uma técnica de discretização de primeira ordem, cuja simplicidade algébrica torna sua implementação direta. Contudo, por ser uma aproximação linear da dinâmica contínua, sua precisão está condicionada à escolha de um período de amostragem suficientemente pequeno, de forma a garantir que o erro introduzido pela discretização permaneça aceitável para o projeto de controle.

## 3 Materiais e Métodos

Neste capítulo, são apresentados os materiais, as ferramentas e os procedimentos metodológicos adotados para o desenvolvimento do pêndulo invertido sobre rodas. Abordando o projeto mecânico assistido por computador (*CAD*) do sistema, a escolha e funcionamento dos sensores IMU, o atuador empregado (motor de passo), e a integração dos dispositivos eletrônicos utilizados em um circuito impresso. Além disso, são descritas as metodologias para aquisição dos estados e o desenvolvimento dos controladores baseados nas técnicas LQR e PDC *fuzzy*.

### 3.1 Projeto do pêndulo invertido sobre rodas

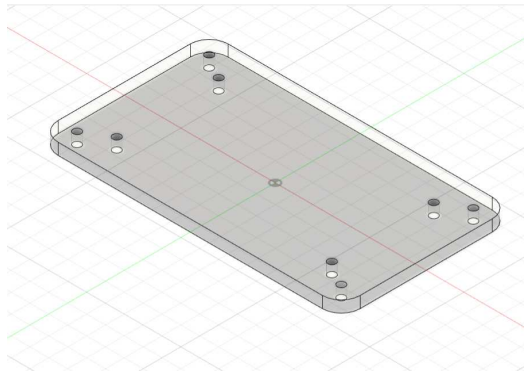
Observa-se na literatura que, desde a concepção dos primeiros modelos de pêndulos invertidos sobre duas rodas (GRASSER et al., 2002), diferentes configurações estruturais foram propostas, assim como diferentes métodos de controle foram implementados. De maneira geral, este tipo de sistema é composto por duas rodas motorizadas acopladas lateralmente a um corpo intermediário, onde são montados os componentes eletrônicos responsáveis pela aquisição de dados e processamento do sinal de controle. As rodas, acionadas de forma independente, devem ser capazes de gerar torque suficiente para estabilizar o sistema em torno de seu eixo de equilíbrio. Para isso, o sistema requer sensores capazes de medir o ângulo de inclinação, o deslocamento linear e suas respectivas velocidades.

#### 3.1.1 Estrutura Mecânica

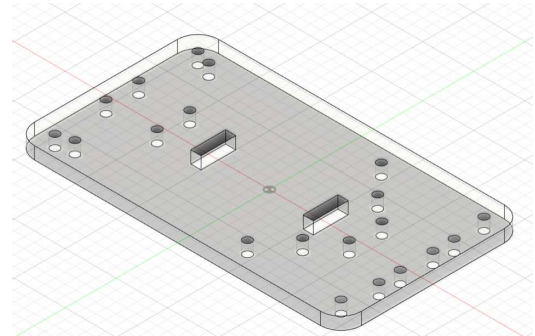
O design da estrutura mecânica foi desenvolvido utilizando técnicas de fabricação digital, como impressão 3D e corte a laser, visando a produção de peças de forma ágil, precisa e de baixo custo (VOLPATO, 2007). O modelo tridimensional da estrutura foi elaborado no software Fusion 360, por meio de ferramentas de desenho assistido por computador (*CAD*). A estrutura é composta por duas bases principais com espessura de 6 mm, representadas nas Figuras 9a e 9b. Em cada base estão acopladas duas peças em acrílico, modeladas conforme os esboços apresentados nas Figuras 9c e 9d, fabricadas com espessura de 3 mm.

Essas bases são interligadas por quatro hastes verticais com 85 mm de altura, exibidas na Figura 10. Altura suficiente para que, entre as bases, seja possível comportar a bateria e a placa de circuito impresso com sensores, drivers dos motores e placa de desenvolvimento já fixadas. Foram modelados suportes para a fixação dos atuadores

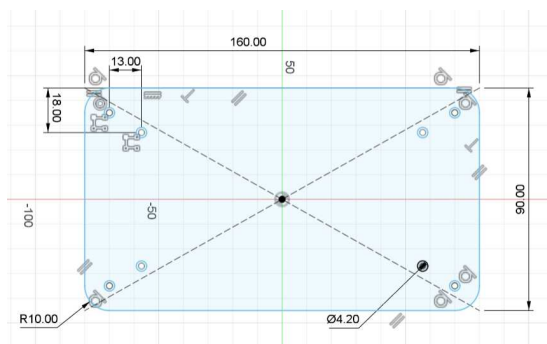
Figura 9 – Desenho CAD das bases do pêndulo.



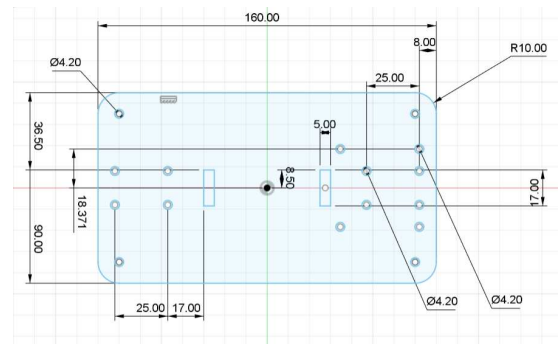
(a) Base superior em perspectiva.



(b) Base inferior em perspectiva.



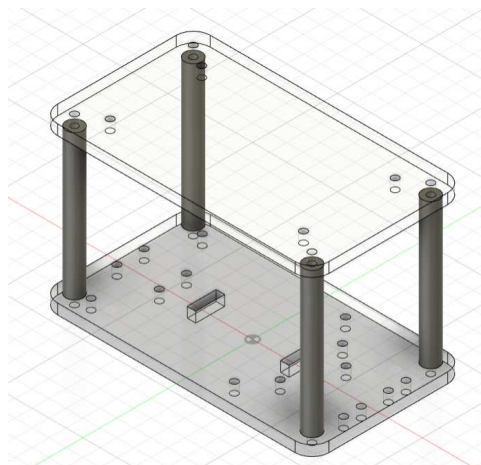
(c) Esboço da base superior.



(d) Esboço da base inferior.

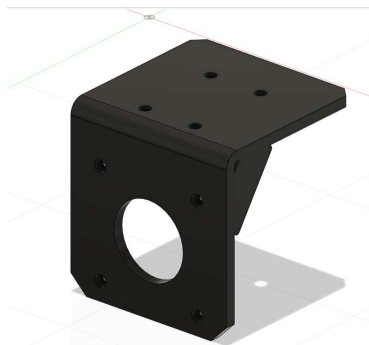
(motores), um compartimento para alojamento do sensor inercial, e também dois suportes para sustentação da PCI. Essas estruturas podem ser vistas na [Figura 11](#).

Figura 10 – Visualização de posicionamento das hastes.

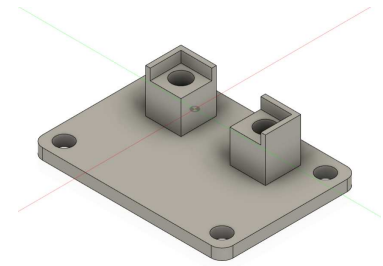


O sensor de inclinação deve ser posicionado o mais próximo possível do centro de massa do pêndulo. Como o corpo intermediário é tratado como um corpo rígido, a altura foi definida pela distância entre a superfície superior da base superior e o eixo das rodas. Assim, considerou-se que o centro de massa estaria, aproximadamente, na metade dessa altura. Além disso, adotou-se a premissa de que o corpo intermediário é composto

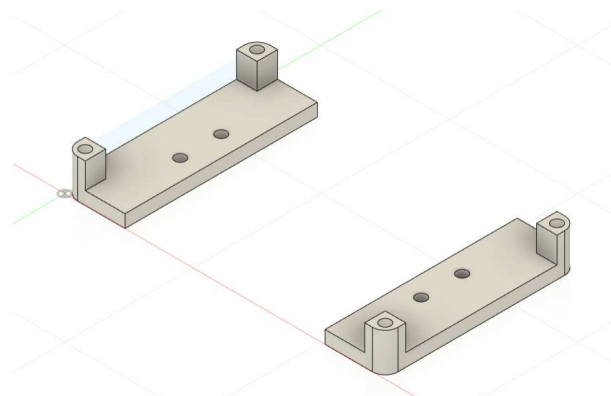
Figura 11 – Peças para suporte e fixação dos dispositivos utilizados.



(a) Suporte para Motor de Passo.



(b) Alojamento para fixação do sensor IMU.



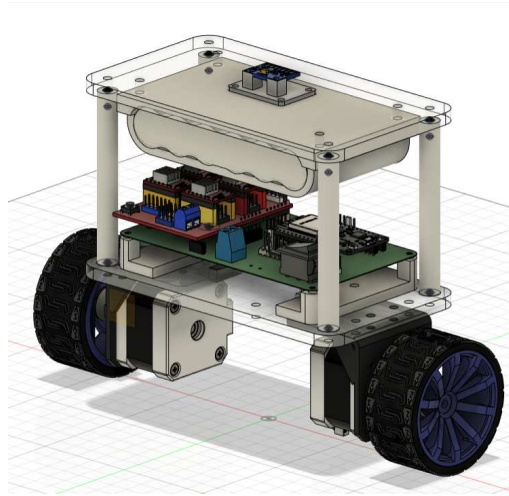
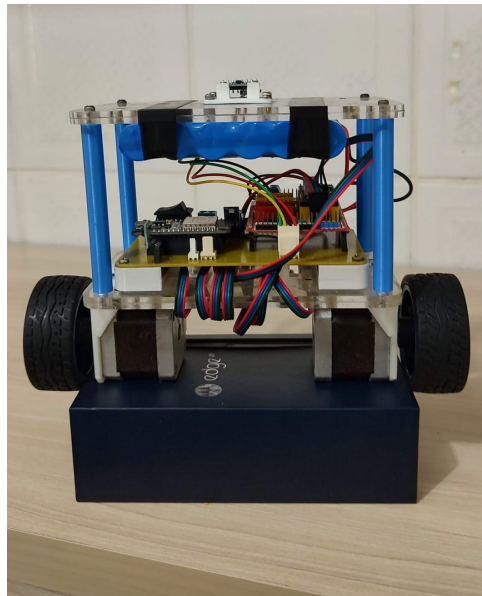
(c) Suporte para PCI.

apenas pelos elementos localizados acima do eixo das rodas, desconsiderando metade da massa dos motores. Com isso, o centro de massa desloca-se ligeiramente para cima, ficando próximo à posição da bateria. Por essa razão, o sensor de inclinação foi instalado na parte superior da estrutura. A montagem de todas as peças modeladas via *CAD* permite uma pré-visualização de como ficará o pêndulo, conforme apresentada na [Figura 12a](#)

Em relação ao processo de fabricação das peças, optou-se pela construção das bases em acrílico via usinagem a laser pela disponibilidade de material, resistência e rapidez no processo de fabricação. Para o restante das peças, como: compartimento do sensor de inclinação e da bateria, suporte para a PCI; assim como as hastes verticais; foram fabricadas em PLA via impressão 3D. A escolha das rodas levou em conta a necessidade de garantir atrito suficiente com a superfície, de forma a evitar deslizamentos durante a operação. O sistema montado após a fabricação das peças pode ser visto na [Figura 12b](#).

### 3.1.2 Sistema Computacional

O elemento central do módulo de controle é a placa de desenvolvimento ESP32 DevKitC V1 baseada no microcontrolador ESP32-WROOM-32, produzido pelo fabricante Espressif. É um dispositivo de baixo custo, mas que oferece boa capacidade de processamento, incorporando dois microprocessadores Xtensa LX6 de 32 bits, que operam a até

Figura 12 – Estrutura montada via *software* CAD e estrutura real do sistema.(a) Montagem da estrutura via *software* CAD.

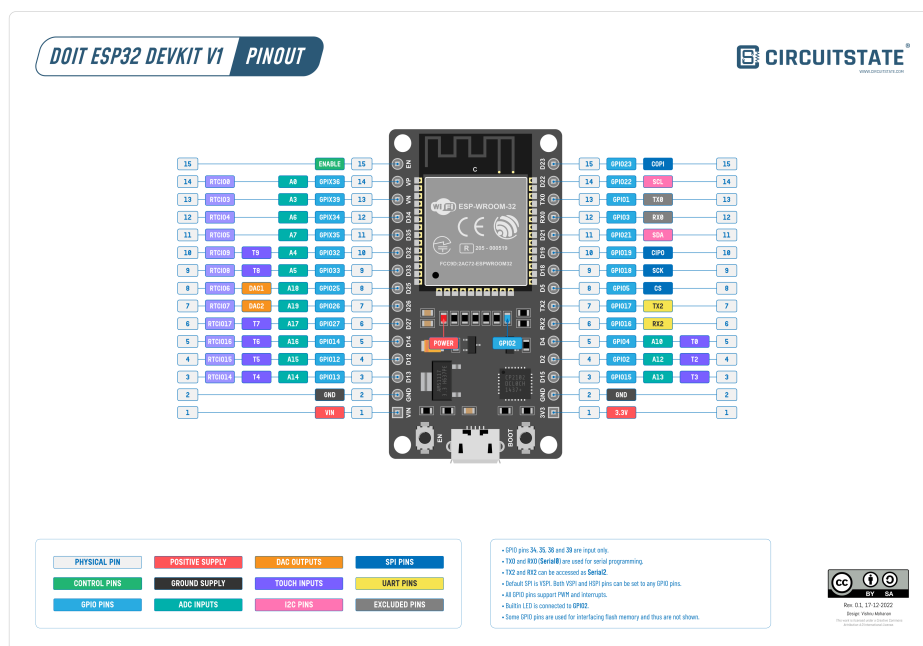
(b) Estrutura real do pêndulo invertido sobre rodas.

160 MHz, além de contar com 520 KB de RAM interna e suporte para memória externa SPI (ESPRESSIF, 2025). Entre os componentes periféricos essenciais, temos o conversor USB-to-UART, que permite a comunicação e programação do dispositivo via USB ao converter o protocolo, e os botões EN e Boot, responsáveis pela reinicialização e ativação do modo de gravação do chip.

Entre suas principais funcionalidades estão múltiplas interfaces de comunicação, como barramento I2C, para a leitura de sensores, e saídas PWM, que permitem o controle de atuadores, como motores. Além disso, sua arquitetura inclui um conjunto de temporizadores, periféricos de captura e capacidade de lidar com até 32 interrupções. O esquema dos pinos do ESP32-DevKitC V1 é apresentado na Figura 13. O ESP32 DevKitC opera com tensões de entrada de até 5 V, como um regulador de tensão interno de 3,3 V para o *chip*.

A programação do ESP32 foi desenvolvida utilizando a extensão PlatformIO integrada ao ambiente de desenvolvimento *Visual Studio Code* (VSCode). A arquitetura *Dual Core* do chip ESP32 permite a execução paralela de tarefas em seus dois núcleos. Com o auxílio do sistema operacional de tempo real FreeRTOS, nativamente embarcado no ESP32, foi possível dedicar um dos núcleos exclusivamente para a execução do algoritmo de controle. Enquanto isso, o segundo núcleo ficou responsável pela execução de tarefas secundárias, como a aquisição de dados dos sensores e atualização dos parâmetros no sistema de visualização via *Serial Monitor*.

Figura 13 – Configuração dos pinos na ESP32-DevKitC.



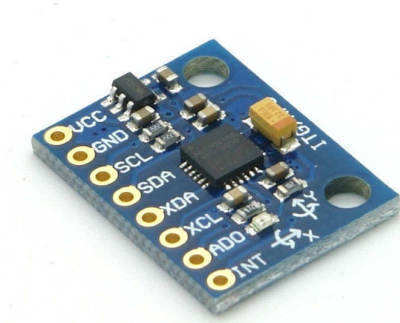
Fonte:(CIRCUITSTATE (2022))

### 3.1.3 Sensor IMU

A Unidade de Medição Inercial (IMU - *Inertial Measurement Unit*) é um dispositivo eletrônico que mede acelerações, taxas de rotação e, em alguns casos, o campo magnético terrestre (GUI; TANG; MUKHOPADHYAY, 2015). No contexto deste trabalho, será o sensor utilizado para a obtenção da inclinação do pêndulo e de sua velocidade angular. Isso é possível por meio da integração de um acelerômetro triaxial, um giroscópio triaxial e, opcionalmente, um magnetômetro triaxial, permitindo assim determinar a orientação de um objeto no espaço.

O sensor escolhido para esta aplicação é o MPU6050, fabricado pela *InvenSense*. O dispositivo é exibido na Figura 14. Esse módulo é amplamente adotado em aplicações que demandam baixo consumo de energia e baixo custo, tornando-o uma escolha popular para projetos acadêmicos e de pesquisa em sistemas de controle, (JIMENEZ; L.; RUGE, 2020).

Figura 14 – Módulo MPU6050.



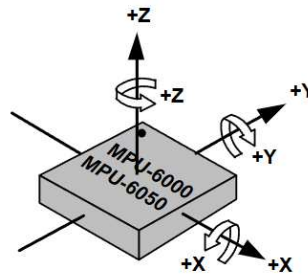
Fonte: [EletronicWings \(2023\)](#).

O funcionamento do MPU6050 baseia-se na tecnologia MEMS (*Micro-Electro-Mechanical Systems*) para seus elementos sensores. No giroscópio, as medições de velocidade angular são obtidas com base no efeito Coriolis. Massas internas são mantidas em vibração constante e, quando o sensor sofre uma rotação em torno de qualquer um dos eixos, surge uma força de Coriolis perpendicular ao movimento vibratório. Essa força causa um pequeno desvio nas massas, que é detectado por meio de variações capacitivas diferenciais, proporcionais à taxa angular. O acelerômetro, por sua vez, opera pela deflexão de uma massa de prova suspensa. Quando o sensor é submetido a uma aceleração, essa massa se desloca, alterando a capacitância entre ela e eletrodos fixos. Essa variação de capacitância também é diferencial e proporcional à aceleração aplicada.

O MPU6050 opera com uma tensão de alimentação principal (VCC) na faixa de 2,375 V a 3,3 V. A comunicação primária com microcontroladores é realizada através do protocolo serial I<sup>2</sup>C (*Inter-Integrated Circuit*). O sensor suporta o modo rápido (Fast Mode) do I<sup>2</sup>C, com uma frequência de clock de até 400kHz. A interface I<sup>2</sup>C requer apenas dois fios: SDA (Serial Data) para a linha de dados e SCL (Serial Clock) para a linha de clock. As leituras do giroscópio são expressas em graus por segundo °/s e podem ser configuradas em diferentes faixas de sensibilidade:  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$  e  $\pm 2000$  °/s, permitindo ajustar a resolução conforme a aplicação. O acelerômetro realiza medições de aceleração nos três eixos (*X*, *Y* e *Z*) conforme [Figura 15](#) com faixas de  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$  e  $\pm 16g$ . As medições são expressas em unidades de gravidade (*g*), sendo que, em repouso sobre uma superfície plana, registra-se aproximadamente 0*g* nos eixos *X* e *Y*, e +1*g* no eixo *Z*, devido à gravidade.

Com base nos vetores de gravidade, é possível estimar o ângulo do vetor resultante dos eixos vertical e horizontal do acelerômetro utilizando trigonometria, ([MAI et al., 2018](#)). Considerando o modelo dinâmico descrito na [Subseção 2.1.1](#), o ângulo de interesse para o sistema é o que rotaciona em relação ao eixo *Y*. Tendo isso em vista, só será descrito o cálculo para a obtenção desse ângulo. Portanto, sejam  $A_x$ ,  $A_y$  e  $A_z$  as componentes de

Figura 15 – Módulo MPU6050.



Fonte: (InvenSense (2013), p.21).

aceleração (expressas em unidades de  $g$ ), o ângulo de inclinação, expresso por  $\theta$  pode ser obtido segundo a relação a seguir.

$$\theta = \arctan \left( \frac{-A_x}{\sqrt{A_y^2 + A_z^2}} \right)$$

### 3.1.4 Motor de passo

Segundo Hadi (2024), a solução mais utilizada para atuadores no sistema do pêndulo invertido sobre rodas é a utilização de motores CC (Corrente Contínua) com armadura devido às características de torque-velocidade possíveis. Neste trabalho, optou-se pela utilização do Motor de Passo devido a duas vantagens em relação ao motor CC. A primeira é a de não ser necessário adicionar uma armadura, pois motores de passo conseguem entregar alto torque. A segunda é a propriedade de rotação por passos discretos, que possibilitam controle de posição e velocidade de forma precisa em malha aberta (IACOB, 2020).

O modelo utilizado foi o NEMA 17, exibido na Figura 16. Esse modelo opera com dois enrolamentos e normalmente apresenta um ângulo de passo de  $1,8^\circ$  por passo. A tensão de alimentação varia de 12 V até 24 V gerando torque de saída que varia de 30 N/cm e 60 N/cm. Seu controle de rotação, seja no sentido horário ou anti-horário, é efetuado através da polarização sequencial correta dos dois enrolamentos e em condições ideais, sua velocidade máxima de rotação pode chegar a 1000 rpm, (JIMENEZ; L.; RUGE, 2020).

#### 3.1.4.1 Driver de Controle

Foram utilizados dois motores NEMA 17, e cada motor foi controlado individualmente utilizando um driver DRV8825 (Figura 17), que possui um chip fabricado pela Texas Instruments. Este driver é projetado para motores de passo bipolares, oferecendo controle

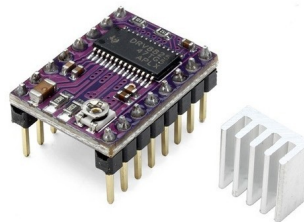
Figura 16 – Motor de Passo NEMA 17 Modelo: 42HS40-1704-13A.



Fonte: ([MAKERHERO, 2025c](#)).

de corrente e suporte a *microstepping*. O controle por *microstepping* permite que motores de passo alcancem maior precisão e suavidade de movimento, especialmente em velocidades pequenas. O DRV8825 permite resolução de até 1/32 por passo. Internamente, o DRV8825 conta com dois drivers em ponte H formados por MOSFETs de canal N, responsáveis por acionar diretamente os enrolamentos do motor. O driver suporta correntes de até 2,5 A por fase, desde que se utilize um sistema adequado de dissipação térmica. Além disso, seu controle é realizado através de sinais digitais simples, sendo um sinal para o avanço de passos (*STEP*) e outro para definição do sentido de rotação (*DIR*), o que simplifica a interface com microcontroladores ([Texas Instruments, 2014](#)).

Figura 17 – Driver Motor de Passo DRV8825.



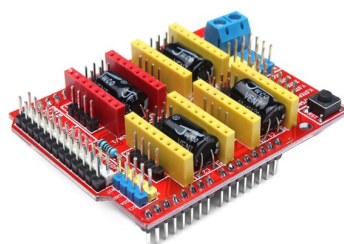
Fonte: ([MAKERHERO, 2025b](#)).

### 3.1.5 Placa de circuito impresso

O funcionamento elétrico do sistema é composto por dois motores de passo e dois *driver's* para controle, o sistema computacional: ESP32 e um sensor IMU: MPU6050. A partir desses dispositivos foi projetada uma placa de circuito impresso utilizando o *software Altium Designer* da *AutoDesk*. Tendo em vista a simplificação do projeto do circuito, optou-se pela utilização do *CNC Shield V3* para Arduino ([Figura 18](#)). Esse *shield*

é bastante utilizado em controle de máquinas CNC's, pois oferece uma forma prática de controlar até 4 motores de passo. A resolução de passo dos motores pode ser feita na própria placa, o que facilita a execução de testes práticos para diferentes resoluções.

Figura 18 – CNC Shield V3.



Fonte: (MAKERHERO, 2025a).

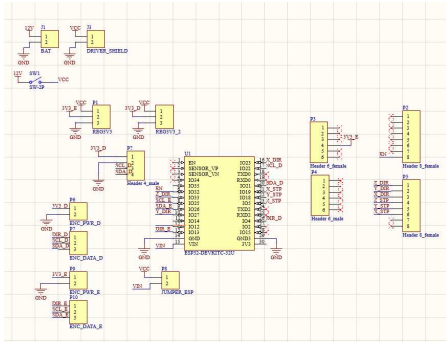
O *shield* é compatível com a placa de desenvolvimento Arduino UNO R3, a placa de circuito impresso teve que ser projetada de forma a adaptar a estrutura dos pinos e estabelecer a conexão com o ESP32. O esquemático elétrico desenvolvido é apresentado na [Figura 19a](#). O circuito é alimentado com uma bateria de 12 V e 5000 mAh. A tensão de alimentação do *shield* é 12 V e um pino lógico de 3,3 V que conectaria a porta correspondente no Arduino UNO R3. O sensor MPU6050 também é alimentado com 3,3 V e para regular a tensão da bateria nesses dispositivos, foram utilizados dois reguladores de tensão AMS1117.

A partir do esquemático, as trilhas de conexão foram configuradas para uma placa fenolite de única face conforme a [Figura 19b](#) e os arquivos de fabricação foram gerados. A PCI foi usinada utilizando a *Router CNC Monofab* chegando ao resultado da [Figura 19c](#). Após a usinagem, os barramentos de pinos para conexão dos componentes foram soldados, chegando à PCI apresentada na [Figura 19d](#).

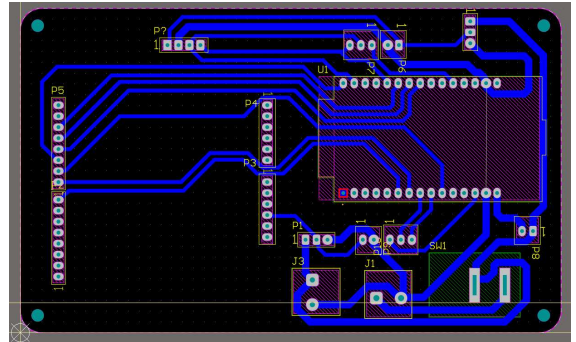
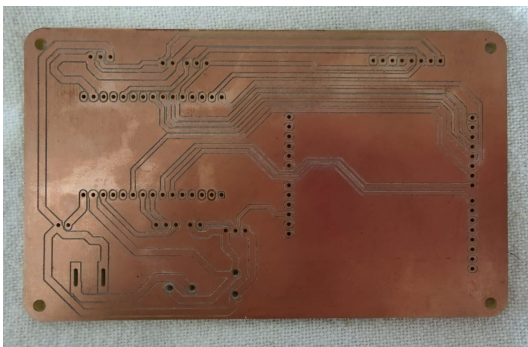
## 3.2 Sistema de Controle

Conforme estabelecido no [Capítulo 2](#), o problema de controle do pêndulo invertido sobre rodas tem como objetivo garantir a estabilização da sua estrutura na posição vertical, bem como o controle da sua movimentação no plano. A realização desses objetivos é viabilizada por meio de uma arquitetura de controle em malha fechada, representada na [Figura 20](#). Esta seção descreve, de forma estruturada, o processo de síntese do sistema de controle desenvolvido. Inicialmente, são apresentados os métodos empregados na aquisição das variáveis de estado, os quais são fundamentais para a realização de controle por realimentação de estados.

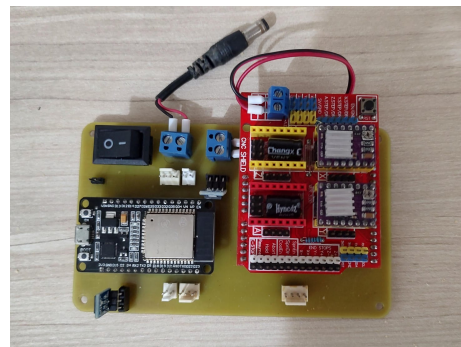
Figura 19 – Etapas do projeto e fabricação da PCI.



(a) Esquema elétrico da PCI.

(b) Roteamento de trilhas no *bottom layer*.

(c) PCI após usinagem em placa de fenolite.

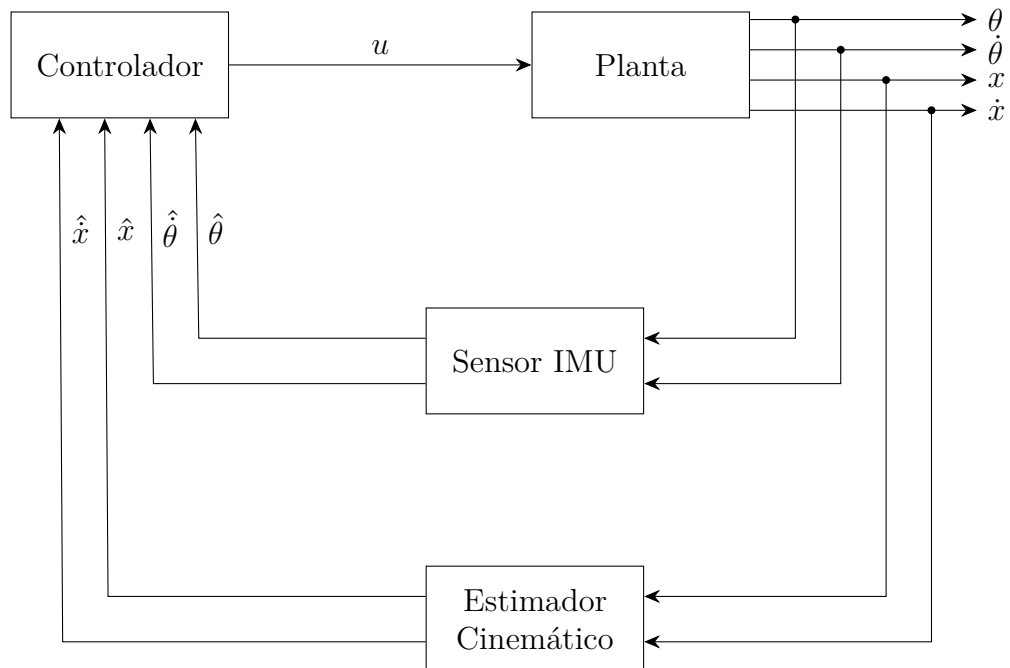


(d) PCI finalizada com componentes montados.

Nesse contexto, aborda-se a implementação de um filtro complementar, utilizado para a estimação do ângulo de inclinação e sua derivada, a partir dos sinais provenientes do sensor inercial MPU6050, bem como a estratégia de odometria, aplicada para a estimação da posição e velocidade linear, utilizando as informações dos motores de passo. Na sequência, é apresentado o projeto de um controlador Linear Quadrático (LQR), empregado como referência de desempenho para fins comparativos. Por fim, desenvolve-se a síntese do controlador Fuzzy do tipo PDC (Parallel Distributed Compensation), fundamentado no modelo Takagi-Sugeno da planta, que representa a principal estratégia adotada neste trabalho para lidar com as não linearidades inerentes ao sistema, bem como são descritos os procedimentos de implementação do sinal de controle no sistema embarcado.

Na Figura 20, os sinais  $\hat{x}$ ,  $\hat{\dot{x}}$ ,  $\hat{\theta}$  e  $\hat{\dot{\theta}}$  correspondem às estimativas da velocidade linear, posição linear, ângulo de inclinação e velocidade angular do pêndulo, respectivamente. O bloco denominado estimador cinemático implementa as equações de odometria, utilizadas para a obtenção dos estados associados ao deslocamento e à velocidade linear do sistema, cuja descrição detalhada será apresentada nas próximas subseções.

Figura 20 – Diagrama de Blocos do Sistema Proposto



### 3.2.1 Aquisição de Medidas dos Estados

#### 3.2.1.1 Filtro Complementar

A eficácia de controladores por realimentação de estados depende da qualidade das medições. Sensores como o MPU6050, abordado na [Subseção 3.1.3](#), estão sujeitos a ruídos e viés, o que compromete a precisão das estimativas. Para mitigar esses efeitos e garantir estimativas mais robustas dos estados, torna-se necessário o emprego de técnicas de fusão sensorial.

Conforme descrito por [Mai et al. \(2018\)](#), acelerômetros são sensíveis não apenas à gravidade, mas também às acelerações lineares impostas pela dinâmica do pêndulo. Assim, durante movimentos acelerados, as leituras de ângulo baseadas no acelerômetro tornam-se imprecisas e ruidosas, estabilizando-se apenas quando o sistema retorna ao equilíbrio. A estimativa de ângulo a partir do giroscópio é obtida pela integração da velocidade angular. Essa abordagem oferece medições precisas no curto prazo e é imune a forças externas, como acelerações lineares. No entanto, a integração sucessiva acumula pequenos erros, levando ao fenômeno conhecido como viés (*drift*). Com isso, o ângulo estimado se afasta progressivamente do valor real ao longo do tempo, não retornando ao zero mesmo quando o sistema retorna à sua posição original.

Diante dessas limitações individuais dos sensores, o filtro complementar surge como uma solução de simples implementação para estimar uma melhor medida do ângulo de inclinação do pêndulo ([JAYAKODY; SUCHARITHARATHNA, 2019](#)). Esse filtro consiste

na combinação ponderada entre duas fontes de dados com características de frequência complementares. O acelerômetro, embora sensível a ruídos de alta frequência e vibrações, fornece uma estimativa estática confiável da inclinação, sendo mais preciso em baixas frequências. Por isso, sua saída é processada por um filtro passa-baixa, que atenua os componentes de alta frequência indesejados, preservando as tendências lentas e estáveis do sinal.

Já o giroscópio fornece uma medida precisa da velocidade angular e é mais confiável em altas frequências, pois não é afetado por acelerações externas. No entanto, sua integração ao longo do tempo leva a um acúmulo de erro (deriva). Para minimizar essa limitação, sua saída é tratada com um filtro passa-alta. Dessa forma, ao combinar os sinais filtrados, o filtro complementar aproveita as vantagens de ambos os sensores, gerando uma estimativa mais estável e precisa do ângulo de inclinação  $\hat{\theta}$ . A equação do filtro complementar em tempo discreto é dada por:

$$\hat{\theta}(k) = \alpha(\hat{\theta}(k-1) + \theta_{\text{gyro}}(k)\Delta_t) + (1 - \alpha)\theta_{\text{acc}}(k)$$

em que  $\alpha$  é o coeficiente de fusão restrito à  $0 < \alpha < 1$  e que determina o quanto a medida do acelerômetro e giroscópio influencia na medida final e  $\Delta_t$  é o intervalo de tempo entre cada amostra da medida.

### 3.2.1.2 Odometria

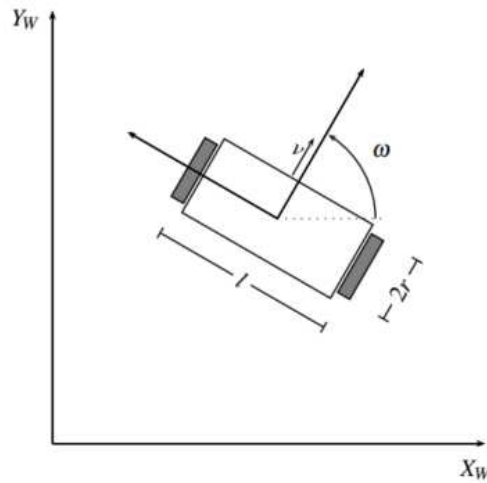
A partir das características do motor de passo descritas na [Subseção 3.1.4](#), é possível obter as medições de posição e velocidade angular de cada roda. Cada passo dado pelo motor corresponde a um incremento angular fixo, definido pela resolução do motor. No caso do modelo NEMA 17 é de  $1,8^\circ$  por passo. Com a utilização do módulo DRV8825, a resolução do motor de passo utilizado pode chegar a  $1/32$  de passo, portanto para cada passo, o motor se movimentará  $0,05625^\circ$ . Isso significa que para cada revolução dada, serão executados 6400 passos. Portanto, a partir de uma referência inicial, a posição do motor pode ser obtida através de:

$$\theta_{\text{motor}} = \text{PASSOS} \cdot \frac{2\pi}{\text{PASSOS POR REV}} \text{rad}$$

A velocidade angular, por sua vez, é determinada pela taxa de variação do número de passos em relação ao tempo, considerando que os comandos correspondam efetivamente ao deslocamento físico do eixo do motor, sem perdas de passo.

$$\omega_{\text{motor}} = \frac{\Delta \text{PASSOS}}{\Delta t} \cdot \frac{2\pi}{\text{PASSOS POR REV}} \text{rad/s}$$

Figura 21 – Variáveis das equações de movimento para um robô de duas rodas.



Fonte: (Velazquez, Cruz e Garcia (2016), p.5).

A partir das medidas de posição e velocidade angular em cada roda possível, é possível estimar a posição e deslocamento do pêndulo, utilizando odometria, (VELAZQUEZ; CRUZ; GARCIA, 2016). A odometria é um processo de estimação do estado posicional de um sistema móvel, que se baseia na integração dos deslocamentos medidos localmente pelos atuadores ou sensores associados às rodas.

Baseado na Figura 21 e assumindo que não há deslizamento entre as rodas e o chão. O valor da velocidade linear em  $m/s$  pode ser obtido por:

$$\hat{x} = \frac{r}{2}(w_r + w_l)$$

onde  $r$  representa o raio da roda em metros e  $w_d$  e  $w_e$  expressam a velocidade angular no motor direito e esquerdo, respectivamente. Considerando que a velocidade linear de um corpo é a derivada de sua posição, ao integrar a velocidade, obtemos a seguinte relação para a posição:

$$\hat{x} = \frac{r}{2}(\theta_d + \theta_e)$$

Observe que a posição depende da medida de ângulo gerado pela rotação do motor, cuja informação já obtemos. Agora que já estabelecemos os métodos de obtenção das medidas dos estados, podemos avançar para o projeto dos controladores por realimentação desses estados.

### 3.2.2 Projeto de controlador LQR

Segundo [Ogata \(2010\)](#), o controlador LQR (*Linear Quadratic Regulator*) é baseado na minimização de um índice de desempenho quadrático. Essa abordagem busca otimizar o comprometimento entre o esforço de controle e a resposta do sistema, penalizando tanto as variações dos estados quanto a magnitude do sinal de controle. Para um sistema linear na forma de espaço de estados, representado pela Equação (2.18), é possível determinar uma matriz  $K$  com ganhos ótimos de forma que a seguinte função de custo  $J$  seja minimizada.

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt$$

onde  $x$  é o vetor de estados,  $u$  é o vetor de sinais de controle,  $Q$  é uma matriz simétrica semidefinida positiva  $n \times n$  em que  $n$  é igual ao número de estados, e  $R$  é uma matriz simétrica definida  $m \times m$ , no qual  $m$  é o número de entradas de controle ([IACOB, 2020](#)). Para que as matrizes  $Q$  e  $R$  sejam semidefinidas e definidas positivas, ambas são estabelecidas como matrizes diagonais com entradas positivas. Por exemplo,

$$Q = \begin{bmatrix} q_{11} & 0 & \cdots & 0 \\ 0 & q_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_{nn} \end{bmatrix} \quad R = \begin{bmatrix} r_{11} & 0 & \cdots & 0 \\ 0 & r_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{mm} \end{bmatrix}.$$

Os valores de  $q_{ii}$  e  $r_{jj}$  são escolhidos considerando o desempenho do sistema em malha fechada. A solução para o problema de otimização definido é dada por um controlador por realimentação de estados com ganhos definidos por:

$$K = R^{-1} B^T S$$

onde  $S$  é a solução para a Equação Algébrica de Riccati:

$$A^T S + S A - S B R^{-1} B^T S + Q = 0 \quad (3.1)$$

Para implementação em sistemas digitais, o projeto do controlador LQR deve ser realizado a partir do modelo discretizado do sistema. Portanto, seja um sistema em espaço de estados discreto:

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k) \\ y(k) &= C_d x(k) + D_d u(k) \end{aligned}$$

em que  $A_d$ ,  $B_d$ ,  $C_d$  e  $D_d$  são matrizes obtidas a partir da discretização, considerando um período de amostragem  $T_s$ , de acordo com o que foi descrito na [Seção 2.3](#). A função de custo  $J$  será definida como:

$$J = \sum_{k=0}^{\infty} (x(k)^T Q x(k) + u(k)^T R u(k))$$

A matriz de ganho  $K$  será calculada por:

$$K = (B_d^T S B_d + R)^{-1} (B_d^T S A_d)$$

onde  $S$  soluciona a Equação (3.1) em tempo discreto:

$$S = A_d^T S A_d - A_d^T S B_d (B_d^T S B_d + R)^{-1} B_d^T S A_d + Q$$

### 3.2.3 Projeto de controlador Fuzzy via PDC

O projeto do controlador *fuzzy* baseado na técnica de PDC, descrita na [Seção 2.2](#), baseia-se na representação do sistema como um modelo Takagi-Sugeno. Esse modelo é obtido a partir da linearização da dinâmica não linear do pêndulo invertido sobre rodas em diferentes regiões operacionais, definidas pela variável premissa. Neste projeto, a variável premissa é definida como o ângulo de inclinação do dispositivo  $\theta$ . A escolha dessa variável é justificada pela forte influência na dinâmica do sistema.

Para cada submodelo linear, é realizada uma expansão em série de Taylor das equações diferenciais do sistema, apresentadas na [Subseção 2.1.2](#), em torno de um ponto de operação específico. O ponto de operação para cada submodelo é caracterizado por um valor fixo de  $\theta$ , assumindo que, neste instante, as demais variáveis de estado são nulas. Os pontos de operação definidos foram em  $\theta = 0^\circ$ ,  $\theta = 30^\circ$  e  $\theta = -30^\circ$ . Em radianos,  $30^\circ$  corresponde a  $\frac{\pi}{6}$ . Dessa forma, os pontos de operação considerados são descritos por:

$$\begin{aligned} x^* &= [0, 0, 0, 0]^T, \text{ para } \theta = 0^\circ \\ x^* &= \left[ \frac{\pi}{6}, 0, 0, 0 \right]^T, \text{ para } \theta = 30^\circ \\ x^* &= \left[ -\frac{\pi}{6}, 0, 0, 0 \right]^T, \text{ para } \theta = -30^\circ \end{aligned}$$

e para cada região escolhida  $u^* = 0$ . Embora os estados  $\dot{\theta}$ ,  $x$  e  $\dot{x}$  sejam assumidos como zero, é importante destacar que as inclinações em  $\pm 30^\circ$  não correspondem a um ponto de equilíbrio estático. Esses pontos representam, na prática, regiões operacionais nas quais se deseja que o modelo linear local seja válido, permitindo capturar os efeitos não lineares associados a inclinações mais acentuadas. Portanto, essa linearização não visa modelar o comportamento de equilíbrio, mas sim aproximar a dinâmica do sistema em torno dessas condições específicas de operação.

Assim, as regras do modelo *fuzzy* são formuladas com base nos submodelos discretizados utilizando o método de Euler:

$$R_1: \text{SE } \theta \text{ for } -30^\circ, \text{ ENTÃO } \begin{cases} x(k+1) = A_{d_1}x(k) + B_{d_1}u(k) \\ y(k) = C_{d_1}x(k) \end{cases}$$

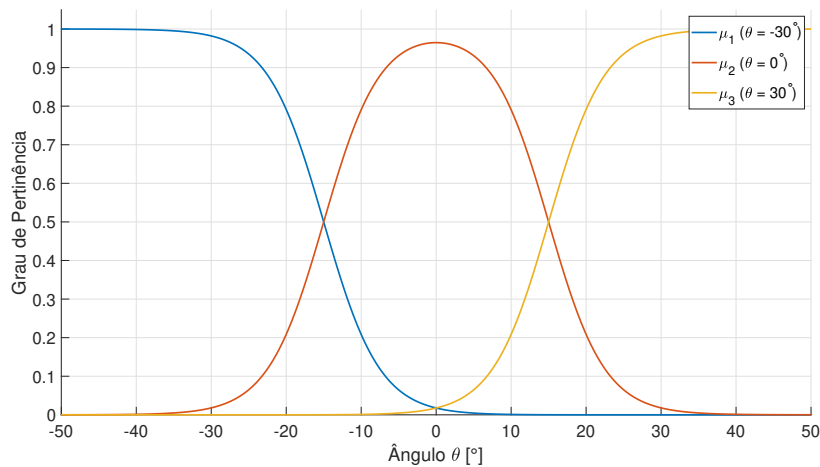
$$R_2: \text{SE } \theta \text{ for } 0, \text{ ENTÃO } \begin{cases} x(k+1) = A_{d_2}x(k) + B_{d_2}u(k) \\ y(k) = C_{d_2}x(k) \end{cases}$$

$$R_3: \text{SE } \theta \text{ for } 30^\circ, \text{ ENTÃO } \begin{cases} x(k+1) = A_{d_3}x(k) + B_{d_3}u(k) \\ y(k) = C_{d_3}x(k) \end{cases}$$

Para cada conjunto *fuzzy* definido, uma função de pertinência  $\mu_i(\theta)$  foi associada para definir o grau de pertencimento de  $x_1$  a cada região de operação. Optou-se pelo modelo de função de pertinência do tipo gaussiana, uma vez que esta apresenta transições suaves e contínuas entre os modelos locais. Cada função é centrada em um valor específico da variável de premissa  $\theta$ , definido de acordo com as regiões operacionais escolhidas, conforme a definição apresentada na [Seção 2.2](#). Dessa forma, podemos definir as seguintes funções de pertinência:

$$\mu_i(\theta) = \exp\left(-\left(\frac{\theta - c_i}{\sigma}\right)^2\right), \quad i = 1, 2, 3. \quad (3.2)$$

Figura 22 – Gráfico das funções de pertinência considerando  $\theta = 0, -30^\circ, 30^\circ$ .



onde  $c_i$  representa o centro da função, correspondente às regiões  $c_1 = -30^\circ$ ,  $c_2 = 0^\circ$  e  $c_3 = 30^\circ$ . Graficamente, as funções de pertinência definidas podem ser vistas na [Figura 22](#). A

normalização das funções de pertinência é feita com base na Equação (2.2.1), considerando o número de funções de pertinência, portanto, a normalização será expressa por:

$$h_i = \frac{\mu_i(\theta)}{\sum_{j=1}^3 \mu_j(\theta)} \quad i = 1, 2, 3.$$

A dinâmica global do sistema no domínio discreto é obtida por meio da combinação convexa dos submodelos locais, ponderada pelos graus de ativação  $h_i(\theta(k))$  associados a cada regra, conforme descrito na Seção 2.2.

$$\begin{cases} x(k+1) = \sum_{i=1}^3 h_i(\theta(k)) (A_{d_i}x(k) + B_{d_i}u(k)) \\ y(k) = \sum_{i=1}^3 h_i(\theta(k)) C_{d_i}x(k) \end{cases} \quad (3.3)$$

A síntese dos ganhos  $F_i$  do controlador em domínio discreto  $u(k)$ , associados a cada submodelo linear local, foi realizada por meio da técnica de alocação de polos, considerando critérios de desempenho definidos para a resposta dinâmica desejável do sistema. Como o modelo do pêndulo invertido sobre rodas possui uma dinâmica de quarta ordem, o projeto requer a definição de quatro polos em malha fechada. Dois desses polos são escolhidos como dominantes, determinando as características principais da resposta transitória. Para esses polos, adotaram-se um tempo de acomodação ( $t_a$ ) de aproximadamente 1,5 segundos e um sobressinal de 5 %, valores que garantem uma resposta rápida e pouco oscilatória.

A partir dessas especificações, os polos dominantes são calculados utilizando a relação associada a sistemas de segunda ordem subamortecidos:

$$s_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} \quad (3.4)$$

Na qual  $\omega_n$  representa a frequência natural não amortecida, e  $\zeta$  é o coeficiente de amortecimento. Esses parâmetros são determinados com base nas seguintes relações entre as características desejadas da resposta e os parâmetros do sistema:

- Sobressinal ( $M_p\%$ ):

$$M_p\% = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}} \times 100\%$$

- Tempo de acomodação ( $t_a$ ), considerando um critério de 2%:

$$t_a = \frac{4}{\zeta\omega_n}$$

Isolando o coeficiente de amortecimento  $\zeta$  em função do sobressinal, tem-se:

$$\zeta = \sqrt{\frac{(\ln(M_{p, \text{ decimal}}))^2}{\pi^2 + (\ln(M_{p, \text{ decimal}}))^2}}$$

Para um sobressinal de  $M_p = 5\%$ , o valor de  $\zeta$  é aproximadamente:

$$\zeta = 0,832$$

A frequência natural não amortecida  $\omega_n$  é obtida por:

$$\omega_n = \frac{4}{\zeta t_a}$$

Substituindo os valores de  $t_a$  e  $\zeta$ , obtém-se:

$$\omega_n \approx 3,605$$

Com os valores de  $\zeta$  e  $\omega_n$ , os polos dominantes são calculados como:

$$s_{1,2} = -2,9 \pm j1,8$$

Os dois polos remanescentes são posicionados de forma a terem uma dinâmica aproximadamente 3 vezes mais rápida que os polos dominantes e maior componente real, de modo a minimizar sua influência na resposta transitória:

$$s_{3,4} = -8,7 \pm j1,8$$

O polinômio característico associado aos polos desejados é expresso por:

$$\Delta(s) = (s - p_1)(s - p_2)(s - p_3)(s - p_4) = s^4 + a_1s^3 + a_2s^2 + a_3s + a_4$$

em que os coeficientes  $a_1, a_2, a_3, a_4$  são obtidos pela expansão do produto dos termos associados aos polos. Para um sistema representado em espaço de estados, ao aplicar uma lei de controle do tipo  $u(t) = -Kx(t)$ , a equação de dinâmica em malha fechada torna-se:

$$\dot{x}(t) = (A - BK)x(t)$$

A equação característica desse sistema é determinada por:

$$\det(sI - (A - BK)) = 0$$

O vetor de ganhos  $K$  deve ser calculado de forma que:

$$\det(sI - (A - BK)) = \Delta(s)$$

Para a obtenção dos polos em domínio discreto, utiliza-se a propriedade de mapeamento  $z = e^{sT_s}$  que estabelece uma correspondência entre o plano- $s$  e o plano  $z$ . Segundo

a descrição de [Aguirre \(2023\)](#), um ponto no plano  $s$  localizado conforme a (3.4) pode ser mapeado para o plano- $z$ , substituindo-se  $s = -\zeta\omega_n + j\omega_n\sqrt{1-\zeta^2}$  em  $z = e^{sT_s}$ :

$$z = e^{sT_s} = e^{(-\zeta\omega_n + j\omega_n\sqrt{1-\zeta^2})T_s} = e^{-\zeta\omega_n T_s} \cdot e^{j\omega_n T_s \sqrt{1-\zeta^2}}$$

A expressão acima representa um número complexo na forma polar:

$$z = r e^{j\vartheta} \quad \text{com} \quad r = e^{-\zeta\omega_n T_s}, \quad \vartheta = \omega_n T_s \sqrt{1-\zeta^2}$$

Tendo como base os polos mapeados em  $z$ , o método para obtenção dos ganhos em domínio discreto  $K_d$  é feito de forma análoga à obtenção dos ganhos em tempo contínuo, mas agora serão consideradas as matrizes discretizadas do modelo em espaço de estados, portanto:

$$\det(zI - (A_d - B_d K_d)) = \Delta_z$$

A partir desse critério, para cada submodelo linear discretizado do modelo Takagi-Sugeno, o vetor de  $F_i$  é obtido de forma que a equação característica do sistema em malha fechada atenda ao polinômio desejado, ou seja:

$$\det(zI - (A_{d_i} - B_{d_i} F_i)) = \Delta_z$$

Os ganhos são, então, associados às regras fuzzy definidas em função da variável premissa  $\theta$ .

$R_{1\text{PDC}}$ : SE  $\theta$  for  $-30^\circ$ , ENTÃO

$$u(k) = -F_1 x(k)$$

$R_{2\text{PDC}}$ : SE  $\theta$  for  $0$ , ENTÃO

$$u(k) = -F_2 x(k)$$

$R_{3\text{PDC}}$ : SE  $\theta$  for  $30^\circ$ , ENTÃO

$$u(k) = -F_3 x(k)$$

O controlador local, por sua vez, será a interpolação dessas ações locais, ponderadas pelos graus de pertinência  $h_i(\theta(k))$ , como segue:

$$u(k) = - \sum_{i=1}^3 h_i(\theta(k)) F_i x(k) \quad i = 1, 2, 3.$$

### 3.3 Sobre a implementação do sinal de controle

O sinal de controle  $u(k)$  calculado pelo controlador PDC, conforme descrito na seção anterior, representa o torque total teórico a ser aplicado às rodas para estabilizar o pêndulo invertido sobre rodas. Segundo [Iacob \(2020\)](#), a maneira mais comum de controlar o torque gerado pelo motor de passo é regulando as correntes dos enrolamentos do estator. Essa regulação é feita aplicando uma quantidade necessária de tensão por meio de um sinal PWM (*Pulse Width Modulation*). No entanto, os drivers utilizados para controlar os motores de passo no sistema real só conseguem controlar a velocidade. Diante disso, torna-se necessário estabelecer uma relação entre o torque calculado e o sinal físico que comanda os motores. O torque é distribuído igualmente entre as duas rodas:

$$T = \frac{u}{2}$$

Em que  $T$  representa o torque gerado por um motor. O torque de cada motor deve mover a mesma massa, nesse caso é considerada a massa do carro ( $M$ ). A aceleração linear, representada pela letra  $a$  é obtida através de sua relação com o torque gerado e a massa  $M$ .

$$a = \frac{T}{M}$$

A velocidade linear pode ser encontrada integrando a aceleração obtida ao longo do tempo. Por se tratar de um sistema digital, a integral da aceleração no tempo contínuo pode ser aproximada pela seguinte equação em tempo discreto:

$$v[kT_s] = v[(k-1)T_s] + a[kT_s]T_s$$

onde,  $k$  é a amostra e  $T_s$  é o tempo de amostragem considerado para o sistema. Uma vez determinada a velocidade linear do carro, a frequência dos pulsos de controle enviados ao driver dos motores é equivalente à velocidade do motor, expressa em passos por segundo. Essa relação é descrita pela seguinte equação:

3

$$f_{passo} = \frac{vn}{2\pi r}$$

em que,  $n$  é o número de passos necessário para que o motor gere uma revolução e  $r$  é o raio da roda utilizada. Uma velocidade positiva representa o movimento para frente em relação à origem, enquanto uma velocidade negativa representa o movimento para trás.

## 4 Resultados

Este capítulo apresenta os resultados obtidos a partir da aplicação dos métodos e das teorias desenvolvidas nos capítulos anteriores. Inicialmente, são descritos os procedimentos e critérios adotados para a obtenção dos parâmetros físicos do sistema. Em seguida, são analisados os resultados relacionados à implementação dos controladores LQR e PDC *fuzzy*, sendo que, no caso do controlador *fuzzy*, também é investigada a influência do número de funções de pertinência sobre o desempenho do sistema. Por fim, são apresentados os resultados referentes ao acompanhamento de uma entrada de referência pelo sistema controlado.

### 4.1 Parâmetros físicos

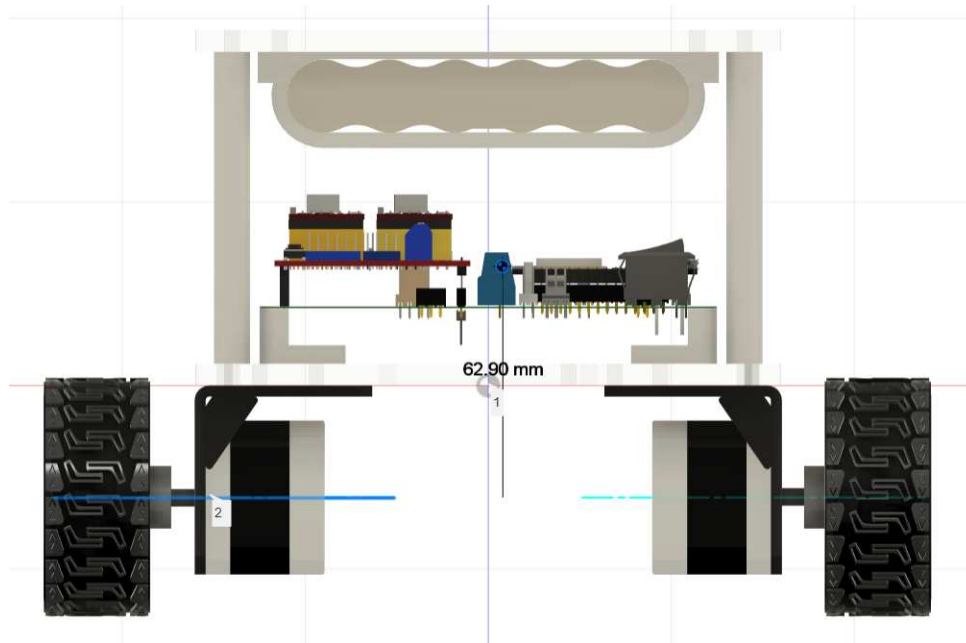
Para a obtenção dos parâmetros físicos do sistema, foram adotadas metodologias semelhantes às descritas em [Iacob \(2020\)](#). O modelo físico é composto por dois subsistemas principais: o carro, representado pelo conjunto de rodas e eixo dos motores de passo; e o pêndulo, que engloba o corpo intermediário, incluindo as bases em acrílico, a bateria, as hastes de sustentação, os suportes estruturais, o compartimento dos dispositivos eletrônicos, a placa de circuito impresso (PCI) e o sensor IMU.

A massa do carro, denotada por  $M$ , corresponde à soma das massas das duas rodas e do eixo dos motores. Através de medições realizadas com uma balança digital, determinou-se que cada conjunto roda e acoplador possui massa de 40 g. Considerando uma estimativa de 50 g para o eixo dos motores, a massa total do carro resultou em 0,180 kg. A massa do pêndulo, representada por  $m$ , foi obtida subtraindo-se do peso total do robô a massa correspondente ao carro, resultando em 1,126 kg.

Neste projeto, a distribuição de massa do corpo intermediário apresenta duas concentrações principais: na região inferior, devido aos motores, e na região superior, em função da bateria. Por essa razão, o corpo intermediário foi modelado como um corpo rígido, cuja distância  $l$  corresponde à medida entre o eixo das rodas e o centro de gravidade do sistema.

A determinação desse parâmetro foi realizada por meio do *software* de modelagem CAD utilizado no projeto. Esse recurso permite estimar a posição do centro de massa considerando as propriedades geométricas e as massas atribuídas a cada componente da estrutura. Atribuindo os valores de massa previamente obtidos para cada parte do sistema (bases, bateria, hastes, suportes, PCI, sensor IMU, entre outros), foi possível empregar a ferramenta de cálculo de centro de gravidade do *software* para estimar sua posição. A

Figura 23 – Localização do centro de gravidade em relação ao eixo das rodas em  $l = 62,90\text{mm}$ .



partir dessa estimativa, obteve-se o valor da distância  $l$ , conforme ilustrado na [Figura 23](#).

O momento de inércia do pêndulo em relação ao eixo das rodas, denotado por  $I$ , foi calculado utilizando o modelo simplificado de massa concentrada no centro de gravidade, segundo a equação:

$$I = ml^2.$$

em que  $m$  corresponde à massa do pêndulo e  $l$  à distância entre o centro de gravidade e o eixo das rodas. Aplicando os valores obtidos para  $m$  e  $l$ , o valor aproximado do momento de inércia é de  $I = 0,00455 \text{ kg} \cdot \text{m}^2$ . A aceleração da gravidade  $g$  é expressa por  $g = 9,81 \text{ m/s}^2$ . A [Tabela 1](#) resume os parâmetros do sistema.

Tabela 1 – Parâmetros do Pêndulo Invertido sobre duas rodas

Parâmetro	Valor
Massa do carro ( $M$ )	1,126 kg
Massa do pêndulo ( $m$ )	0,180 kg
Distância COG até eixo das rodas ( $l$ )	0,063 m
Aceleração da gravidade ( $g$ )	9,81 m/s <sup>2</sup>
Raio da roda ( $r$ )	0,0325 m

Substituindo os valores apresentados na [Tabela 1](#) na equação do modelo linearizado em espaço de estados, dada pela Equação (2.27), obtém-se a seguinte representação numérica do sistema pêndulo sobre duas rodas.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 113,7256 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -7,4863 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ -8,9267 \\ 0 \\ 1,3569 \end{bmatrix} u \quad (4.1)$$

## 4.2 Estabilidade em torno da origem

Nesta seção, realiza-se a análise de estabilidade do sistema. Inicialmente, a avaliação é conduzida para o modelo linearizado em espaço de estados, considerando sua operação em torno do ponto de equilíbrio na origem, sob atuação do controlador LQR. Na sequência, analisa-se o desempenho do controlador *fuzzy* baseado na abordagem Takagi-Sugeno, considerando diferentes quantidades de funções de pertinência, a fim de avaliar o impacto na resposta dinâmica do sistema.

A avaliação dos controladores será realizada considerando critérios como estabilidade do ângulo de inclinação, erro de posição em regime permanente e magnitude do sinal de controle. Além disso, analisa-se a capacidade de rejeição a distúrbios. Os ganhos discretos, que são implementados no microcontrolador ESP32 *DevKit* foram obtidos a partir da discretização do modelo contínuo representado pela Equação (2.27). Conforme detalhado anteriormente na Seção 2.3, a discretização foi realizada pelo método de Euler com  $T_s = 8$  ms, sendo essa configuração aplicada em todas as análises apresentadas nesta seção. Como resultado desse processo, obtêm-se as matrizes discretizadas apresentadas a seguir.

$$A_d = \begin{bmatrix} 1 & 0,0080 & 0 & 0 \\ 1,0157 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0,0080 \\ -0,0577 & 0 & 0 & 1 \end{bmatrix}, \quad B_d = \begin{bmatrix} 0 \\ -0,0780 \\ 0 \\ 0,0105 \end{bmatrix}$$

### 4.2.1 Controle LQR

Os ganhos do controlador LQR foram obtidos utilizando a função `lqrd` do *MATLAB*, que calcula a solução do problema para o sistema definido no domínio discreto. A função `lqrd` requer como entrada as matrizes do sistema discretizado e as matrizes de ponderação  $Q$  e  $R$ , que foram definidas conforme (IACOB, 2020).

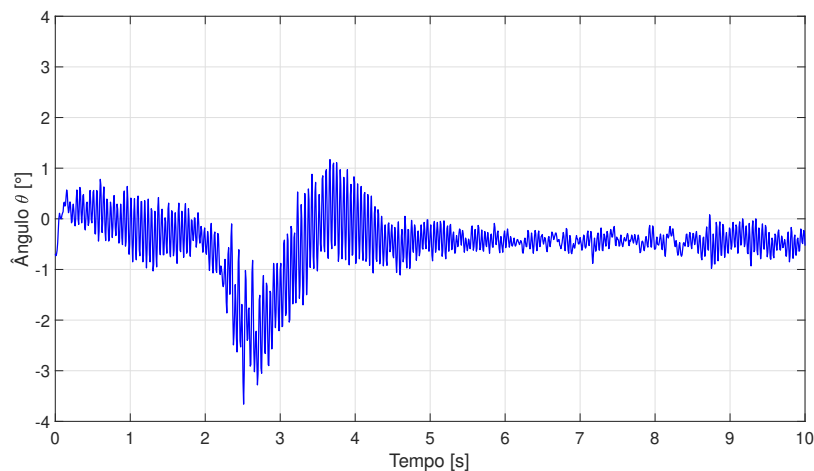
$$Q = \begin{bmatrix} 200 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 150 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R = 1$$

Os seguintes ganhos do controlador encontrados são dados por:

$$K_d = \begin{bmatrix} -38,9453 & -3,8167 & -10,8391 & -7,8956 \end{bmatrix}$$

A seguir, são apresentados os gráficos que demonstram a evolução do ângulo de inclinação  $\theta$  e da posição linear  $x$  do sistema, permitindo avaliar o desempenho do controlador LQR na estabilização do pêndulo invertido sobre duas rodas.

Figura 24 – Resposta do ângulo  $\theta$  à aplicação de distúrbios sob ação do controlador LQR.



A partir da análise da [Figura 25](#), observa-se que, diante da aplicação de pequenos distúrbios na estrutura, o controlador LQR é capaz de reagir de forma eficiente, restabelecendo o ângulo de inclinação  $\theta$  próximo ao ponto de equilíbrio em 0, mesmo na presença de ruído. No entanto, embora o sistema permaneça estabilizado em termos de inclinação, verifica-se que há um deslocamento permanente na posição linear  $x$  em relação à origem.

Figura 25 – Resposta da posição linear  $x$  à aplicação de distúrbios com ação do controlador LQR.

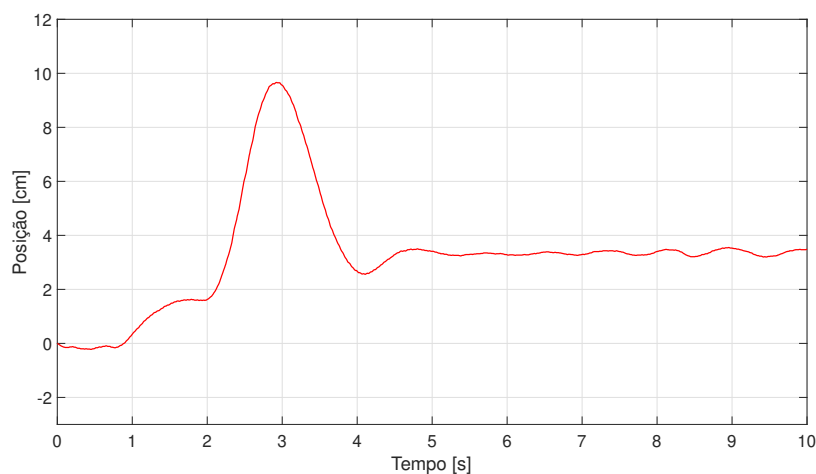
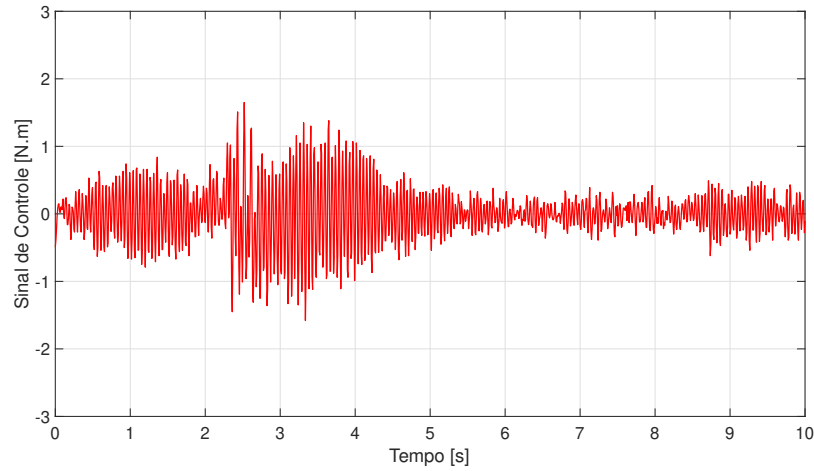


Figura 26 – Sinal de controle (Torque) com aplicação de distúrbio no sistema.



Na [Figura 26](#), observa-se que o sinal de controle apresenta variações compatíveis com o intervalo em que o distúrbio é aplicado ao sistema. Dessa forma, o controlador mantém a estabilidade em torno do ponto de equilíbrio, mas, como pode ser observado, não foi capaz de retornar à posição na origem.

#### 4.2.2 Modelo com 3 Funções de Pertinência

De acordo com o que foi descrito na [Subseção 3.2.3](#), para o modelo *fuzzy* T-S, os submodelos locais foram obtidos em relação à variável de premissa  $\theta$ . Para o modelo com 3 funções de pertinência, foram consideradas as regiões de operação de  $\theta$  entre  $-30^\circ$ ,  $0^\circ$  e  $30^\circ$ . A seguir, são apresentadas as representações numéricas em domínio discreto dos submodelos obtidos:

- Para  $\theta = -30^\circ$

$$A_{d_1} = \begin{bmatrix} 1 & 0,0080 & 0 & 0 \\ 0,5120 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0,0080 \\ -0,0130 & 0 & 0 & 1 \end{bmatrix}, \quad B_{d_1} = \begin{bmatrix} 0 \\ -0,0571 \\ 0 \\ 0,0088 \end{bmatrix}.$$

- Para  $\theta = 0^\circ$

$$A_{d_2} = \begin{bmatrix} 1 & 0,0080 & 0 & 0 \\ 1,0157 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0,0080 \\ -0,0577 & 0 & 0 & 1 \end{bmatrix}, \quad B_{d_2} = \begin{bmatrix} 0 \\ -0,0780 \\ 0 \\ 0,0105 \end{bmatrix}.$$

- Para  $\theta = 30^\circ$

$$A_{d_3} = \begin{bmatrix} 1 & 0,0080 & 0 & 0 \\ 0,5120 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0,0080 \\ -0,0130 & 0 & 0 & 1 \end{bmatrix}, \quad B_{d_3} = \begin{bmatrix} 0 \\ -0,0571 \\ 0 \\ 0,0088 \end{bmatrix}.$$

Nota-se que as matrizes associadas às regiões de  $\theta = -30^\circ$  e  $\theta = 30^\circ$  são iguais, uma vez que a dinâmica do sistema é simétrica em relação ao eixo vertical do pêndulo. A partir da descrição sobre a obtenção do vetor de ganhos através dos polos na [Subseção 3.2.3](#), para cada submodelo local, foi obtida uma matriz de ganho  $F_i$ , de forma que os polos em malha fechada do sistema discretizado fossem alocados em:

$$z = [0,9230 \pm j0,0148 \quad 0,9762 \pm j0,0156].$$

Esses polos correspondem aos polos no plano- $s$  do sistema de tempo contínuo mapeados ao plano- $z$ , considerando a taxa de amostragem de  $T_s = 8\text{ms}$ . Com auxílio da função `place` no *MATLAB*, os seguintes vetores de ganhos foram obtidos:

- Para  $\theta = -30^\circ$

$$F_1 = [-43,4827 \quad -5,6730 \quad -20,6418 \quad -13,8297]$$

- Para  $\theta = 0^\circ$

$$F_2 = [-37,6316 \quad -3,7292 \quad -12,7522 \quad -8,5438]$$

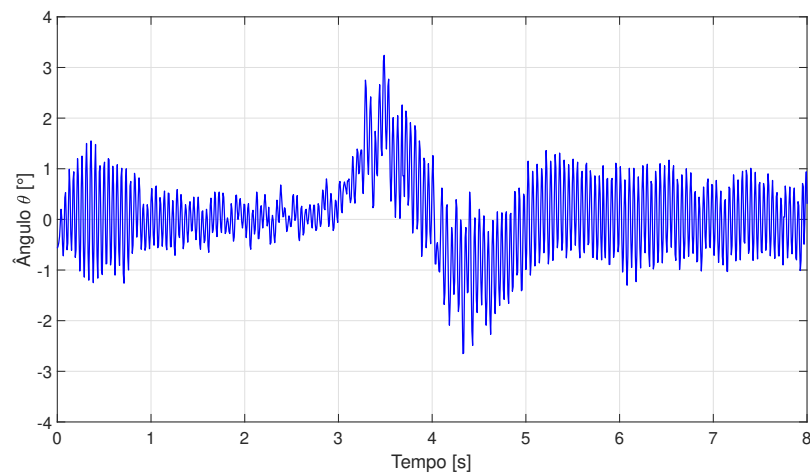
- Para  $\theta = 30^\circ$

$$F_3 = [-43,4827 \quad -5,6730 \quad -20,6418 \quad -13,8297]$$

Cada submodelo e matriz de ganho foi associada a uma função de pertinência descrita pela Equação (3.2). O centro das funções corresponde às regiões de operação definidas para a variável premissa. A largura das funções, representada pelo parâmetro  $\sigma$ , foi fixada em 15, por buscar uma distribuição simétrica ao longo do intervalo dos ângulos. A seguir, é realizada a análise da resposta do sistema sob a atuação do controlador PDC com três regras.

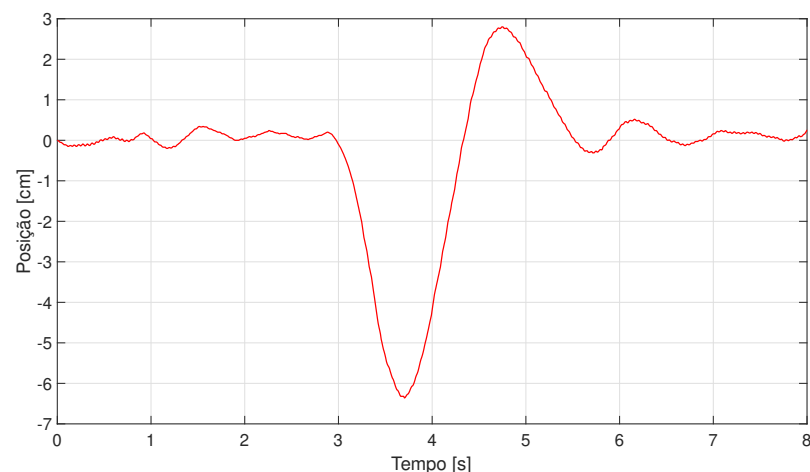
Com base na [Figura 27](#), é possível perceber que, a um distúrbio aplicado à estrutura do pêndulo, o ângulo varia a aproximadamente  $3^\circ$ , mas o controlador consegue atuar em cima dessa variação e fazer com que o sistema retorne ao equilíbrio novamente. A oscilação residual do ângulo após a compensação do distúrbio permanece dentro de uma faixa reduzida, em torno de  $\pm 1,5^\circ$ , indicando um desempenho satisfatório do controlador na rejeição a pequenos distúrbios e manutenção da estabilidade.

Figura 27 – Resposta do ângulo de inclinação  $\theta$  sob distúrbio, com ação do controlador PDC (3 regras *fuzzy*).



Analisando a [Figura 28](#), observa-se que, sem a aplicação de distúrbios, a posição do pêndulo apresenta uma leve redução nas oscilações durante a correção do ângulo  $\theta$ . Quando um distúrbio é aplicado, o sistema controlado pelo PDC *fuzzy* com 3 regras é capaz de retornar à posição de referência (origem), diferentemente do comportamento observado sob a ação do controlador LQR. Já a [Figura 29](#) evidencia que o sinal de controle do PDC apresenta oscilações residuais maiores em comparação ao controlador LQR, indicando uma resposta mais sensível.

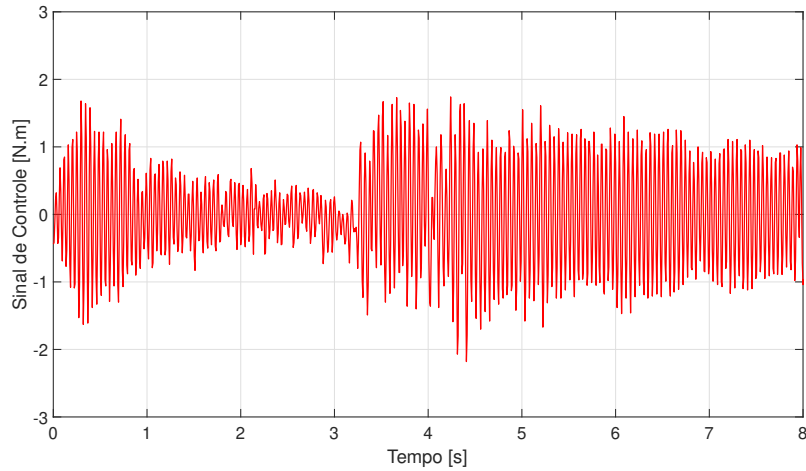
Figura 28 – Resposta da posição linear  $x$  à aplicação do distúrbio com controlador PDC (3 regras *fuzzy*).



#### 4.2.3 Modelo com 5 Funções de Pertinência

Para o modelo *fuzzy* com 5 regras, foram mantidos os três submodelos locais previamente descritos, aos quais foram adicionados mais dois submodelos correspondentes

Figura 29 – Evolução do sinal de controle em torque sob distúrbio com controlador PDC (3 regras *fuzzy*).



às regiões de operação em torno  $-45^\circ$  e  $45^\circ$ . Cobrindo uma maior região de valores de  $\theta$ . De forma que as regras sejam descritas da seguinte forma:

$$R_1: \text{SE } \theta \text{ for } -45^\circ, \text{ ENTÃO } \begin{cases} x(k+1) = A_{d_1}x(k) + B_{d_1}u(k) \\ y(k) = C_{d_1}x(k) \end{cases}$$

$$R_2: \text{SE } \theta \text{ for } -30^\circ, \text{ ENTÃO } \begin{cases} x(k+1) = A_{d_2}x(k) + B_{d_2}u(k) \\ y(k) = C_{d_2}x(k) \end{cases}$$

$$R_3: \text{SE } \theta \text{ for } 0^\circ, \text{ ENTÃO } \begin{cases} x(k+1) = A_{d_3}x(k) + B_{d_3}u(k) \\ y(k) = C_{d_3}x(k) \end{cases}$$

$$R_4: \text{SE } \theta \text{ for } 30^\circ, \text{ ENTÃO } \begin{cases} x(k+1) = A_{d_4}x(k) + B_{d_4}u(k) \\ y(k) = C_{d_4}x(k) \end{cases}$$

$$R_5: \text{SE } \theta \text{ for } 45^\circ, \text{ ENTÃO } \begin{cases} x(k+1) = A_{d_5}x(k) + B_{d_5}u(k) \\ y(k) = C_{d_5}x(k) \end{cases}$$

Com base nas regras fuzzy estabelecidas, são apresentadas a seguir as matrizes numéricas dos submodelos discretizados correspondentes aos pontos de operação próximos a  $\theta = -45^\circ$  e  $\theta = 45^\circ$ :

- Para  $\theta = -45^\circ$

$$A_{d_1} = \begin{bmatrix} 1 & 0,0080 & 0 & 0 \\ 0,2426 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0,0080 \\ -0,0113 & 0 & 0 & 1 \end{bmatrix}, \quad B_{d_1} = \begin{bmatrix} 0 \\ -0,0403 \\ 0 \\ 0,0076 \end{bmatrix}.$$

- Para  $\theta = 45^\circ$

$$A_{d_5} = \begin{bmatrix} 1 & 0,0080 & 0 & 0 \\ 0,2426 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0,0080 \\ -0,0113 & 0 & 0 & 1 \end{bmatrix}, \quad B_{d_5} = \begin{bmatrix} 0 \\ -0,0403 \\ 0 \\ 0,0076 \end{bmatrix}.$$

Mantendo o critério de alocação de polos aplicado no projeto do controlador PDC com três regras fuzzy, determinam-se a seguir as matrizes de ganho correspondentes aos dois submodelos adicionais.

- Para  $\theta = -45^\circ$

$$F_1 = [-56,7270 \quad -9,2845 \quad -33,7485 \quad -22,6110]$$

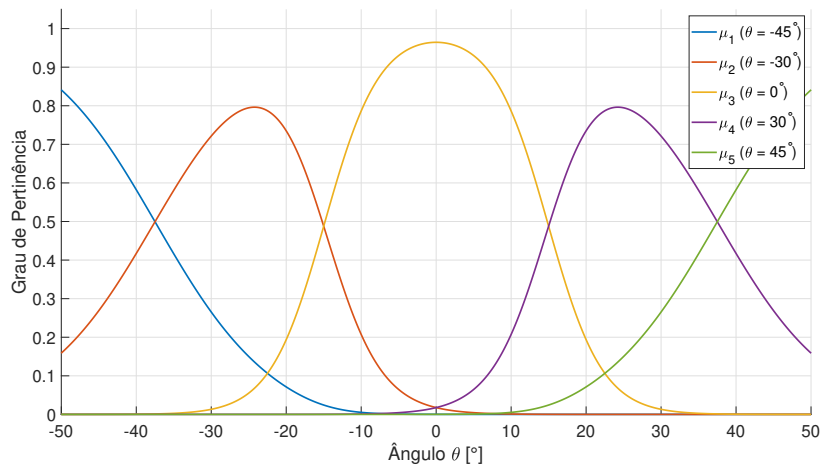
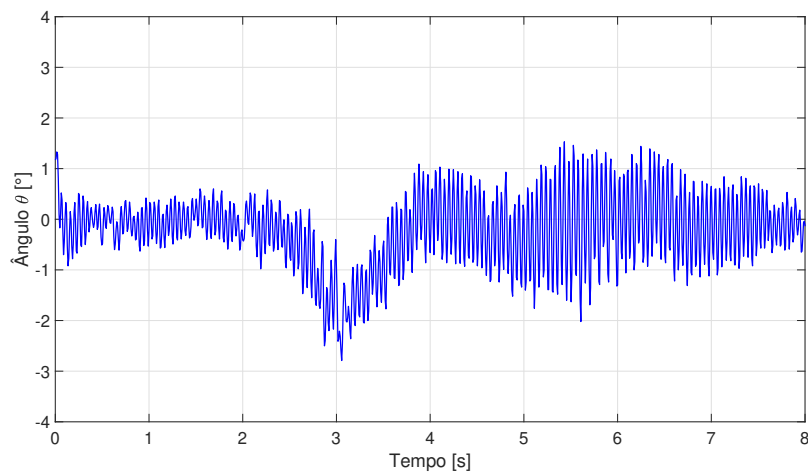
- Para  $\theta = 45^\circ$

$$F_5 = [-56,7270 \quad -9,2845 \quad -33,7485 \quad -22,6110]$$

Com a inclusão de dois novos submodelos, torna-se necessário associar cada um deles a uma função de pertinência, definida segundo o formato apresentado na Equação (3.2). Para isso, os centros das novas funções foram posicionados nas regiões correspondentes a  $\theta = -45^\circ$  e  $\theta = 45^\circ$ , de modo a garantir a cobertura das extremidades da faixa operacional. O valor da distribuição  $\sigma$  foi mantido o mesmo utilizado no modelo com apenas 3 regras. A Figura 30 ilustra graficamente a distribuição das cinco funções de pertinência utilizadas no modelo Takagi-Sugeno com cinco regras.

A Figura 31 apresenta o comportamento do ângulo de inclinação do pêndulo sob a ação do controlador PDC fuzzy com cinco funções de pertinência. Observa-se que a resposta do ângulo é semelhante à obtida com o modelo de três regras. Devido às limitações da estrutura mecânica e do atuador, o pêndulo não atingiu inclinações mais elevadas, o que resultou na não ativação de todos os submodelos lineares previstos.

Assim como na configuração anterior, a Figura 32 mostra que o sistema apresenta capacidade de retornar à região próxima da origem após a aplicação de um distúrbio, sem apresentar deslocamentos permanentes na posição. O sinal de controle do controlador

Figura 30 – Gráfico das funções de pertinência considerando  $\theta = -45^\circ, -30^\circ, 0, 30^\circ, 45^\circ$ .Figura 31 – Resposta do ângulo de inclinação  $\theta$  à aplicação do distúrbio com controlador PDC (5 regras *fuzzy*).

PDC com cinco regras, conforme apresentado na [Figura 33](#), mantém um padrão de comportamento semelhante ao do modelo com três funções de pertinência. As oscilações residuais continuam presentes e, em alguns momentos, a magnitude do sinal de controle é ligeiramente superior, o que está associado ao maior número de submodelos ativos, proporcionando uma resposta marginalmente mais sensível.

### 4.3 Rastreamento de uma entrada de referência

Nesta seção, avalia-se o desempenho dos controladores frente à aplicação de uma referência,  $r(t)$ , na posição linear  $x$ . A abordagem adotada para o rastreamento baseia-se em uma modificação direta da lei de realimentação de estados, na qual o ganho é aplicado sobre o erro em relação a uma referência pré-definida. A lei de controle é expressa por

Figura 32 – Resposta da posição linear  $x$  à aplicação do distúrbio com controlador PDC (5 regras *fuzzy*).

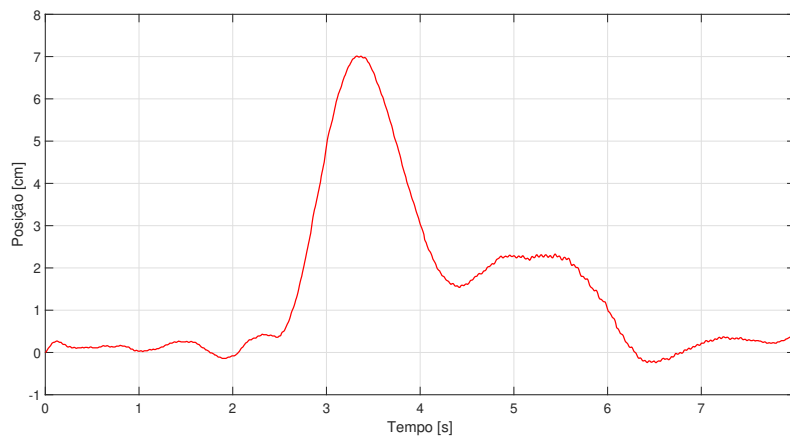
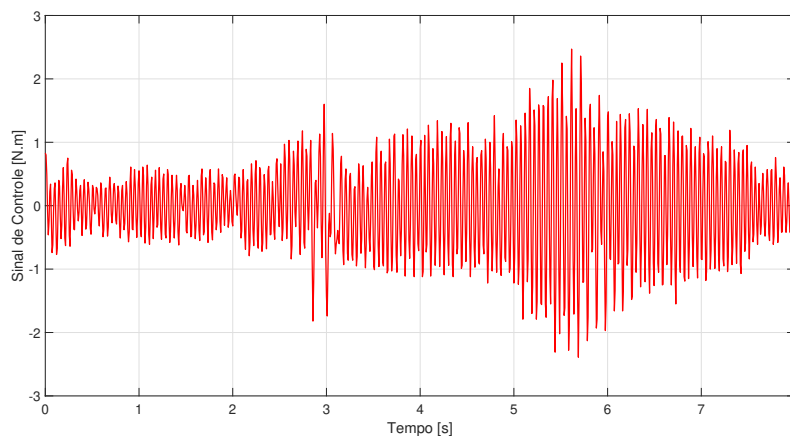


Figura 33 – Evolução do sinal de controle em torque sob distúrbio com controlador PDC (5 regras *fuzzy*).



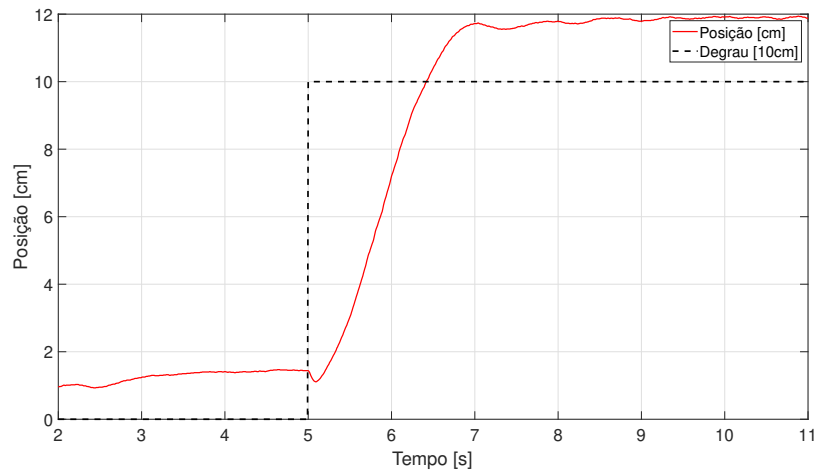
$$u(t) = -K(x(t) - r(t))$$

$r(t)$  representa o vetor de referência, aplicado especificamente ao estado de posição linear  $x$ , enquanto os demais estados permanecem com referência nula. Esse método é frequentemente observado na literatura aplicada a sistemas de pêndulo invertido sobre rodas, especialmente em implementações de baixo custo, devido à sua simplicidade e facilidade de implementação. Além disso, para deslocamentos de pequena magnitude e quando a prioridade é a manutenção do equilíbrio do pêndulo, essa abordagem se mostra suficiente para atender aos requisitos práticos do sistema.

Vale ressaltar que essa solução apresenta limitações teóricas, como a possibilidade de erro em regime permanente na posição, uma vez que o controlador não possui uma ação integral explícita nem um ganho de avanço de referência formalmente calculado. No entanto, o objetivo principal deste teste é avaliar a capacidade dos controladores em

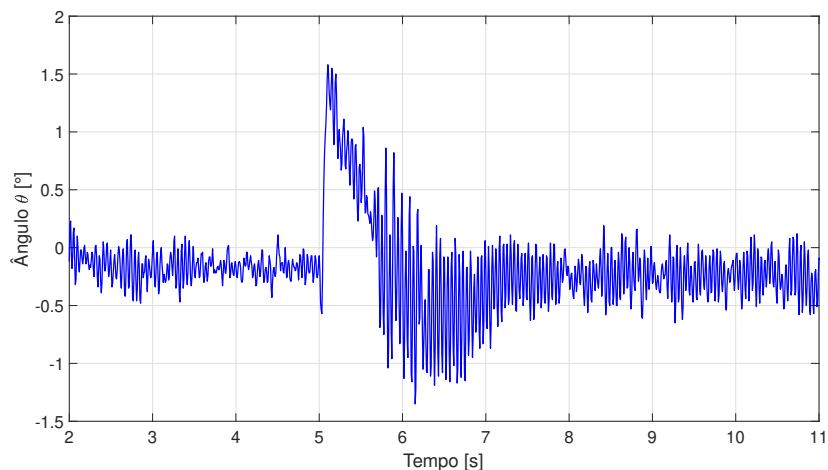
manter o equilíbrio do pêndulo durante deslocamentos.

Figura 34 – Resposta da posição linear  $x$  a um degrau de 10 cm, sob ação do controlador LQR.



Para todos os testes realizados, a referência em degrau foi aplicada após 5 segundos em relação à inicialização do sistema, permitindo que o pêndulo atingisse sua condição de equilíbrio antes de iniciar o rastreamento. Observa-se na [Figura 34](#), que sob ação do controlador LQR o sistema apresenta um erro de posição desde o instante inicial, ou seja, ao ser inicializado, o pêndulo tende a se deslocar ligeiramente para frente, estabilizando-se em uma posição deslocada da origem. Esse erro inicial se propaga ao longo da execução, de modo que, ao tentar rastrear a referência em degrau, o sistema mantém um erro de aproximadamente 2 cm em relação ao valor desejado. Conforme apresentado na [Figura 35](#), durante o deslocamento, o ângulo  $\theta$  sofre uma variação de aproximadamente  $2^\circ$ . No entanto, uma vez concluído o deslocamento, o controlador é capaz de estabilizar o pêndulo, retornando o ângulo ao ponto de equilíbrio em regime permanente.

Figura 35 – Evolução do ângulo de inclinação  $\theta$  durante o rastreamento de posição, com controlador LQR.



A partir da análise da [Figura 36](#), observa-se que, sob ação do controlador *PDC fuzzy*, o sistema apresenta um desempenho superior na manutenção da posição em torno da origem, quando comparado ao controlador LQR. O erro de posição é visivelmente menor ao longo do rastreamento da referência. Além disso, após a aplicação do degrau de referência, a posição linear, em regime permanente, apresenta um erro de aproximadamente 1 cm em relação ao valor de referência.

Figura 36 – Resposta da posição linear  $x$  a um degrau de 10 cm, sob ação do controlador *PDC* (3 regras *fuzzy*)

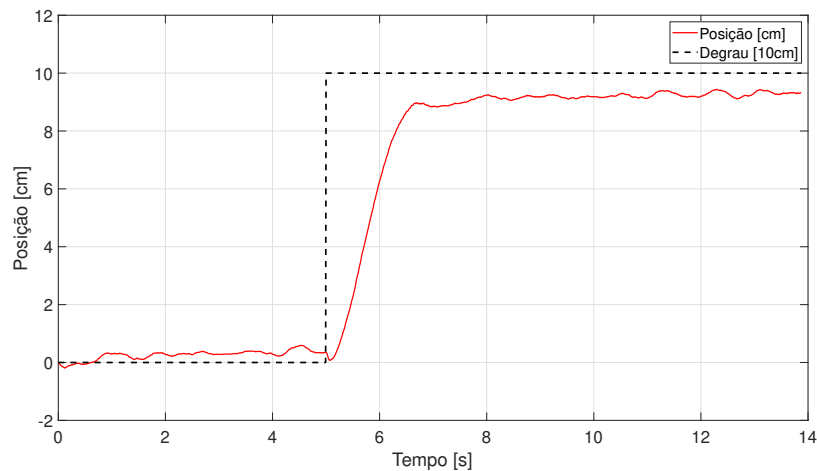
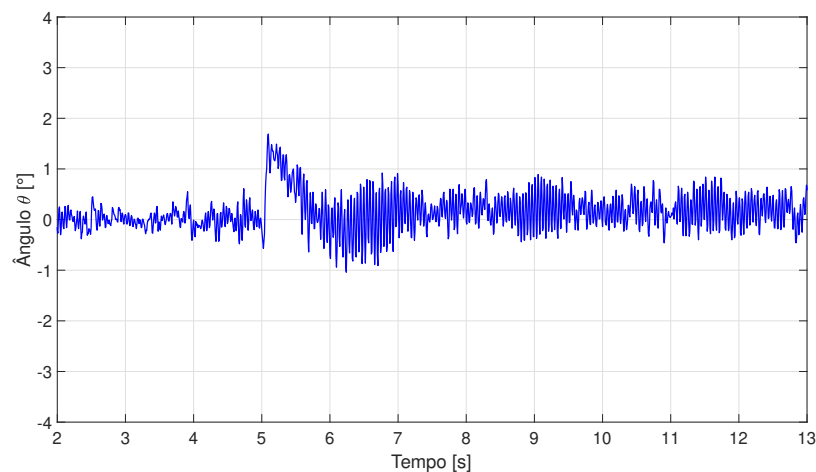


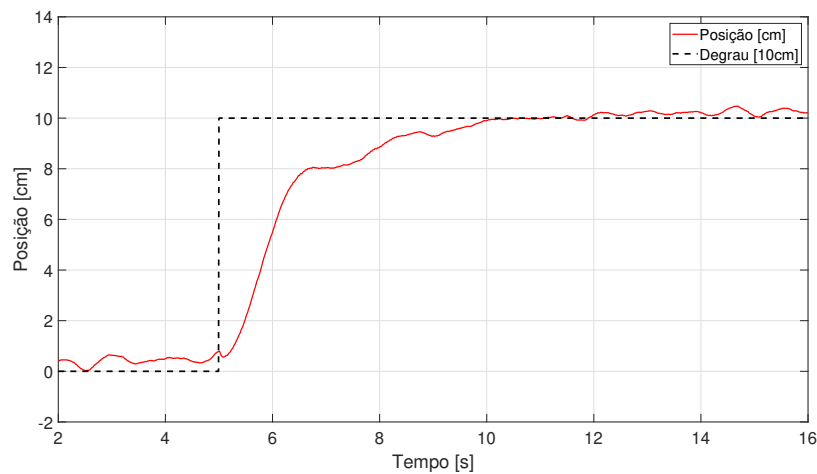
Figura 37 – Evolução do ângulo de inclinação  $\theta$  durante o rastreamento de posição, sob ação do controlador *PDC* (3 regras *fuzzy*).



Similar ao observado com o controlador LQR, o ângulo de inclinação  $\theta$  apresenta um pequeno desvio durante o deslocamento, com variação em torno de  $2^\circ$ . No entanto, conforme ilustrado na [Figura 37](#), o controlador *PDC fuzzy* é capaz de estabilizar o sistema, fazendo com que o ângulo retorne ao ponto de equilíbrio após o término do deslocamento.

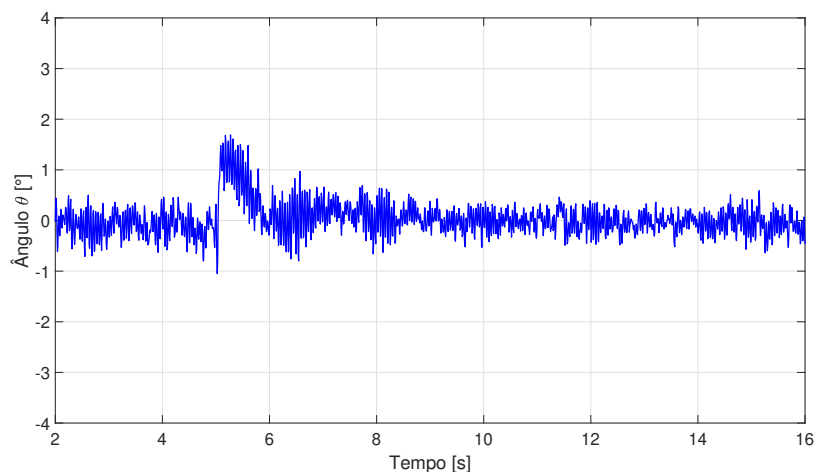
Na [Figura 38](#), observa-se que o sistema é capaz de rastrear a referência de 10 cm de forma satisfatória. Após a aplicação do degrau, a resposta apresenta um pequeno

Figura 38 – Resposta da posição linear  $x$  a um degrau de 10 cm, sob ação do controlador  $PDC$  com 5 regras  $fuzzy$



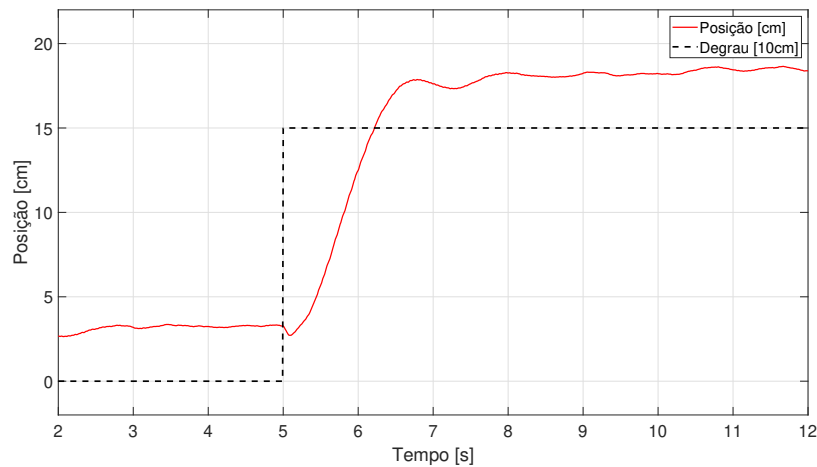
deslocamento com rápida convergência para o valor de referência. O erro em regime permanente é praticamente nulo, indicando que o controlador  $PDC$   $fuzzy$  com cinco regras mantém a capacidade de rastreamento da posição, com menor erro em relação ao modelo com três funções de pertinência. As diferenças em relação ao modelo anterior são pouco expressivas, sendo possível perceber apenas uma leve suavização nas transições e nas oscilações iniciais.

Figura 39 – Evolução do ângulo de inclinação  $\theta$  durante o rastreamento de posição, sob ação do controlador  $PDC$  com 5 regras  $fuzzy$ .



A resposta do ângulo  $\theta$  apresentada na [Figura 39](#) é coerente com o observado nas análises anteriores, tanto para o controlador  $fuzzy$  com três regras quanto para o LQR. Após a estabilização da posição, o ângulo retorna ao ponto de equilíbrio, permanecendo próximo de zero em regime permanente. Isso demonstra que o aumento no número de funções de pertinência não impactou de forma significativa o comportamento do ângulo, mantendo o desempenho na estabilização do pêndulo durante o rastreamento.

Figura 40 – Resposta da posição linear  $x$  a um degrau de 15 cm, sob ação do controlador PDC com 5 regras *fuzzy*.



Observa-se que a referência de 10 cm representa um deslocamento relativamente pequeno para o sistema, o que permite sua realização sem comprometer a estabilidade. Com base nos resultados obtidos, foram realizados testes com referências de maior amplitude, nos quais o sistema continuou apresentando capacidade de manter o equilíbrio. No entanto, para valores próximos de 20 cm, os atuadores não foram capazes de compensar adequadamente o desvio no ângulo de inclinação, resultando na perda de estabilidade. A maior distância para a qual o sistema conseguiu realizar o rastreamento, mantendo o equilíbrio do pêndulo, foi de 15 cm, sob ação do controlador *fuzzy* PDC com cinco regras, como ilustrado na [Figura 40](#).

## 5 Conclusões

Este trabalho apresentou o desenvolvimento de uma planta de controle baseada no sistema do pêndulo invertido sobre duas rodas, juntamente com a implementação de um controlador por realimentação de estados, conhecido como controlador PDC. Ao longo do processo, foi possível observar como os aspectos teóricos de Sistemas de Controle se relacionam diretamente com o comportamento do sistema em ambiente real.

A forma como a planta é construída impacta diretamente na obtenção dos parâmetros físicos do sistema, os quais são utilizados no cálculo dos ganhos do controlador e influenciam o desempenho na estabilidade e na resposta dinâmica. No sistema desenvolvido, considerou-se que o posicionamento da bateria na parte superior pode ter contribuído para a dificuldade em recuperar a estabilidade quando distúrbios de maior magnitude foram aplicados. Além disso, observou-se que a qualidade dos sinais dos estados influencia a atuação do controlador, especialmente em cenários com ruído de medição. Nesse contexto, a utilização de técnicas de estimação de estados mais precisas contribui para mitigar os efeitos do ruído e aprimorar a resposta do sistema.

A escolha dos atuadores é um fator que merece uma análise criteriosa, uma vez que eles também possuem dinâmica própria e, portanto, não podem ser considerados elementos ideais. Quando não devidamente modelados ou compensados, os atuadores podem introduzir limitações que impactam diretamente o desempenho do sistema e sua estabilidade. No sistema desenvolvido, a utilização de motores de passo demonstrou ser uma solução viável para a proposta, especialmente considerando a precisão no posicionamento. Entretanto, o controle dos motores foi implementado por meio de uma biblioteca de alto nível, o que, embora facilite a programação, pode ter restringido o acesso a parâmetros críticos de controle, como a aplicação do torque máximo disponível. Essa limitação pode ter contribuído para que a planta apresentasse maior sensibilidade a perturbações de maior amplitude, uma vez que o atuador não era capaz de gerar torque suficiente para compensar rapidamente desvios significativos na dinâmica do sistema.

O uso de modelos lineares, obtidos por meio da linearização em torno de um ponto de operação, simplifica a aplicação de técnicas de controle clássico. No entanto, essa abordagem não representa de forma completa a dinâmica do sistema fora dessa região. A utilização da lógica *fuzzy*, associada à metodologia de projeto de controle PDC, permite combinar diferentes submodelos lineares locais, oferecendo uma representação mais abrangente da dinâmica do sistema e viabilizando o uso de estratégias de controle baseadas em modelos lineares de forma segmentada.

Com base nos resultados obtidos, conclui-se que o controlador *fuzzy* PDC aplicado

ao sistema de pêndulo invertido sobre duas rodas apresentou desempenho satisfatório na estabilização e no rastreamento de referência, considerando as limitações físicas e estruturais da planta. O desenvolvimento do trabalho permitiu compreender de forma prática a influência de fatores como a modelagem do sistema, a qualidade dos sinais de realimentação e as características dos atuadores no desempenho do controle. A partir dessas observações, este estudo reforça a importância de considerar tanto os aspectos teóricos quanto as restrições práticas na implementação de sistemas de controle aplicados a sistemas reais.

## 5.1 Trabalhos Futuros

A partir da execução deste trabalho, foram identificados alguns pontos que podem ser explorados em estudos futuros, tanto para aprimoramento da implementação quanto para aprofundamento na análise dos aspectos práticos envolvidos no controle do sistema. Esses pontos estão descritos a seguir.

- Avaliar a capacidade dos motores de passo na geração de torque máximo, considerando uma implementação de controle mais refinada. Isso pode incluir o desenvolvimento de uma rotina de controle manual ou a utilização de bibliotecas alternativas que ofereçam maior acesso aos parâmetros de operação dos motores.
- Considerar a dinâmica completa do sistema, explorando os três graus de liberdade inerentes ao pêndulo invertido sobre rodas. O modelo adotado neste trabalho foi restrito a dois graus de liberdade, o que limita a representação de certos comportamentos dinâmicos. A expansão para o modelo completo pode contribuir para uma análise mais abrangente do sistema.
- Revisar e aprimorar a estratégia de rastreamento de referência, adotando metodologias que não dependam de formulações simplificadas. Isso pode permitir uma melhoria no desempenho do sistema durante deslocamentos, especialmente na redução de erros de regime permanente.
- Avaliar modificações na estrutura mecânica, visando uma melhor distribuição de massa. Alterações no posicionamento de componentes, como a bateria, podem contribuir para aumentar a estabilidade e reduzir a sensibilidade a perturbações externas.
- Implementar métodos avançados de estimação de estados na presença de ruído, como o Filtro de Kalman Estendido. Essa abordagem pode fornecer uma estimativa mais precisa dos estados, particularmente do ângulo de inclinação, quando considerada a dinâmica não linear do sistema.

- Avaliar métodos para obtenção de ganhos que garantam estabilidade em uma região mais ampla do espaço de estados. Embora o controlador LQR determine ganhos ótimos a partir da solução de um problema de otimização quadrática, essa abordagem é válida apenas na vizinhança do ponto de equilíbrio, uma vez que se baseia no modelo linearizado. Como alternativa, o uso de critérios de estabilidade globais, por meio de *Linear Matrix Inequalities (LMIs)*, pode ser aplicado ao controlador *fuzzy* PDC, possibilitando uma análise mais robusta da estabilidade do sistema e potencialmente aprimorando seu desempenho em uma faixa maior de operação.

O código-fonte desenvolvido para controle do pêndulo invertido sobre rodas está disponível em [SILVA \(2025\)](#).

# Referências

AGUIRRE, L. *Controle de Sistemas Amostrados, Segunda Edição*. [S.l.: s.n.], 2023. ISBN 978-65-8706-565-6. Citado 3 vezes nas páginas 33, 34 e 54.

BABUŠKA, R. *Fuzzy modeling for control*. [S.l.]: Springer Science & Business Media, 2012. v. 12. Citado 2 vezes nas páginas 25 e 27.

CHAVES, W. d. S. et al. *Modelagem, identificação e controle de um pêndulo invertido de duas rodas*. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, 2020. Citado na página 12.

CIRCUITSTATE. *DOIT ESP32 DevKit V1 Wi-Fi Development Board – Pinout Diagram & Arduino Reference*. 2022. Acesso em: 24 mai. 2025. Disponível em: <<https://www.circuitstate.com/pinouts/doit-esp32-devkit-v1-wifi-development-board-pinout-diagram-and-reference/>>. Citado na página 40.

ELETRONICWINGS. *MPU-6050 6-Axis Accelerometer and Gyroscope*. 2023. Acesso em: 23 mai. 2025. Disponível em: <<https://www.electronicwings.com/sensors-modules/mpu6050-gyroscope-accelerometer-temperature-sensor-module>>. Citado na página 41.

ESPRESSIF. *esp-dev-kits Documentation*. 2025. Acesso em: 24 mai. 2025. Disponível em: <<https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32/esp32-devkitc/index.html>>. Citado na página 39.

FRANKLIN, G.; POWELL, J.; WORKMAN, M. *Digital Control of Dynamic Systems-Third Edition*. [S.l.: s.n.], 2022. ISBN ISBN: 0-9791226-3-5 or ISBN13: 978-0-9791226-3-7. Citado 2 vezes nas páginas 34 e 35.

FRANKLIN, G. F. et al. *Feedback control of dynamic systems*. [S.l.]: Pearson Upper Saddle River, NJ, 2010. v. 10. Citado na página 18.

FRAUNHOFERIML. *evoBOT® - The evolution of autonomous mobile robotic systems*. 2023. Acesso em: 16 abr. 2025. Disponível em: <[https://www.inkl.fraunhofer.de/en/fields\\_of\\_activity/material-flow-systems/iot-and-embedded-systems/evobot.html](https://www.inkl.fraunhofer.de/en/fields_of_activity/material-flow-systems/iot-and-embedded-systems/evobot.html)>. Citado na página 11.

GONZALEZ-CARDENAS, Y. et al. Design and implementation of a discrete-pdc controller for stabilization of an inverted pendulum on a self-balancing car using a convex approach. *Mathematical and Computational Applications*, v. 29, p. 83, 09 2024. Citado na página 12.

GRASSER, F. et al. Joe: a mobile, inverted pendulum. *IEEE Transactions on Industrial Electronics*, v. 49, n. 1, p. 107–114, 2002. Citado 2 vezes nas páginas 12 e 36.

GUI, P.; TANG, L.; MUKHOPADHYAY, S. Mems based imu for tilting measurement: Comparison of complementary and kalman filter based data fusion. In: *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*. [S.l.: s.n.], 2015. p. 2004–2009. Citado na página 40.

- HADI, G. Review of hardware implementation for the two-wheeled self-balancing robot. v. 5, p. 114–118, 09 2024. Citado na página 42.
- HAZEM, Z. B.; BINGÜL, Z. Comprehensive review of different pendulum structures in engineering applications. *IEEE Access*, v. 11, p. 42862–42880, 2023. Citado na página 11.
- IACOB, C.-G. Linear and angular position control of a custom built stepper motor driven self-balancing robot. In: *2020 24th International Conference on System Theory, Control and Computing (ICSTCC)*. [S.l.: s.n.], 2020. p. 648–653. Citado 8 vezes nas páginas 12, 14, 15, 42, 49, 55, 56 e 58.
- InvenSense. *MPU-6000 and MPU-6050 Product Specification*. [S.l.], 2013. Document Number: PS-MPU-6000A-00, Revision: 3.4. Disponível em: <<https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>>. Citado na página 42.
- JANG, J.-S. R.; SUN, C.-T. *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*. USA: Prentice-Hall, Inc., 1996. ISBN 0132610663. Citado 2 vezes nas páginas 23 e 25.
- JAYAKODY, D.; SUCHARITHARATHNA, K. Control unit for a two-wheel self-balancing robot. *Global Journal of Researches in Engineering*, p. 7–12, 01 2019. Citado na página 46.
- JIMENEZ, A.; L., F. J.; RUGE, I. Modeling and control of a two wheeled self-balancing robot: a didactic platform for control engineering education. In: . [S.l.: s.n.], 2020. Citado 5 vezes nas páginas 11, 12, 14, 40 e 42.
- KHALIL, H. K. *Nonlinear systems; 3rd ed.* Upper Saddle River, NJ: Prentice-Hall, 2002. The book can be consulted by contacting: PH-AID: Wallet, Lionel. Disponível em: <<https://cds.cern.ch/record/1173048>>. Citado na página 18.
- MAI, T. A. et al. Development of a microcontroller-based adaptive fuzzy controller for a two-wheeled self-balancing robot. *Microsystem Technologies*, Springer, v. 24, p. 3677–3687, 2018. Citado 3 vezes nas páginas 12, 41 e 46.
- MAKERHERO. *CNC Shield V3 para Arduino*. 2025. Acesso em: 24 mai. 2025. Disponível em: <<https://www.makehero.com/produto/cnc-shield-v3-para-arduino-impressora-3d/>>. Citado na página 44.
- MAKERHERO. *Driver Motor de Passo DRV8825*. 2025. Acesso em: 24 mai. 2025. Disponível em: <<https://www.makehero.com/produto/driver-motor-de-passo-drv8825/>>. Citado na página 43.
- MAKERHERO. *Motor de Passo NEMA 17 1,7A 40mm para CNC*. 2025. Acesso em: 24 mai. 2025. Disponível em: <<https://www.makehero.com/produto/motor-de-passo-nema-17-17a-40mm-para-cnc/>>. Citado na página 43.
- MAMDANI. Application of fuzzy logic to approximate reasoning using linguistic synthesis. *IEEE Transactions on Computers*, C-26, n. 12, p. 1182–1191, 1977. Citado na página 26.
- MONESS, M.; MAHMOUD, D.; HUSSEIN, A. Real-time mamdani-like fuzzy and fusion-based fuzzy controllers for balancing two-wheeled inverted pendulum. *Journal of Ambient Intelligence and Humanized Computing*, v. 13, 07 2022. Citado na página 25.

- NGUYEN, A.-T. et al. Fuzzy control systems: Past, present and future. *IEEE Computational Intelligence Magazine*, v. 14, n. 1, p. 56–68, 2019. Citado 2 vezes nas páginas 29 e 31.
- NIEMANN, D. et al. Parallel distributed compensation for takagi-sugeno fuzzy models: New stability conditions and dynamic feedback designs. *IFAC Proceedings Volumes*, v. 32, n. 2, p. 5374–5379, 1999. ISSN 1474-6670. 14th IFAC World Congress 1999, Beijing, Chia, 5-9 July. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1474667017569153>>. Citado na página 32.
- OGATA, K. *Engenharia de controle moderno*. 5a Edição. [S.l.]: Pearson, 2010. ISBN 978-85-7605-810-6. Citado 3 vezes nas páginas 11, 17 e 49.
- PUTRA, L. G. B.; WAHAB, F.; TAMBA, T. A. Design and implementation of linear quadratic regulator control for two-wheeled self-balancing robot. *Bulletin of Electrical Engineering and Informatics*, v. 14, n. 2, p. 931–939, 2025. Citado na página 12.
- ROSS, T. Fuzzy logic with engineering applications. *Fuzzy Logic with Engineering Applications: Third Edition*, 01 2009. Citado 2 vezes nas páginas 21 e 28.
- SEGWAY. *Ninebot S2*. 2023. Acesso em: 16 abr. 2025. Disponível em: <<https://store.segway.com/ninebot-s2>>. Citado na página 11.
- SILVA, A. *TWBR\_Stepper\_Motor: Controle para pêndulo invertido sobre rodas com motor de passo*. 2025. <[https://github.com/silvaairton03/TWBR\\_Stepper\\_Motor](https://github.com/silvaairton03/TWBR_Stepper_Motor)>. Acesso em: 23 jun. 2025. Citado na página 73.
- SUGENO, M. On stability of fuzzy systems expressed by fuzzy rules with singleton consequents. *IEEE Transactions on Fuzzy Systems*, v. 7, n. 2, p. 201–224, 1999. Citado na página 26.
- SUGENO, M.; KANG, G. Fuzzy modelling and control of multilayer incinerator. *Fuzzy Sets and Systems*, v. 18, n. 3, p. 329–345, 1986. ISSN 0165-0114. Dedicated to the memory of Richard E. Bellman. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0165011486900102>>. Citado na página 32.
- TAKAGI, T.; SUGENO, M. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15, n. 1, p. 116–132, 1985. Citado 2 vezes nas páginas 26 e 29.
- TANAKA, K.; WANG, H. O. *Fuzzy control systems design and analysis: a linear matrix inequality approach*. [S.l.]: John Wiley & Sons, 2004. Citado 3 vezes nas páginas 29, 31 e 32.
- Texas Instruments. *DRV8825 Stepper Motor Controller IC Datasheet (Rev. F)*. [S.l.], 2014. SLVSA73F – Revised July 2014. Disponível em: <<https://www.ti.com/lit/gpn/drv8825>>. Citado na página 43.
- VELAZQUEZ, M.; CRUZ, D.; GARCIA, S. Velocity and motion control of a self-balancing vehicle based on a cascade control strategy. *International Journal of Advanced Robotic Systems*, v. 13, p. 1, 01 2016. Citado na página 48.
- VOLPATO, N. *Prototipagem rápida: tecnologias e aplicações*. [S.l.]: Editora Blucher, 2007. Citado na página 36.

WANG, H.; TANAKA, K.; GRIFFIN, M. An analytical framework of fuzzy modeling and control of nonlinear systems: stability and design issues. In: *Proceedings of 1995 American Control Conference - ACC'95*. [S.l.: s.n.], 1995. v. 3, p. 2272–2276 vol.3. Citado 2 vezes nas páginas 12 e 32.

ZADEH, L. Fuzzy sets. *Information and Control*, v. 8, n. 3, p. 338–353, 1965. ISSN 0019-9958. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S001999586590241X>>. Citado na página 22.

ZADEH, L. A. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3, n. 1, p. 28–44, 1973. Citado na página 26.