



**UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO  
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS DE  
COMUNICAÇÃO E AUTOMAÇÃO**

**EDPO RODRIGUES DE MORAIS**

**DESENVOLVIMENTO DE UM CONTROLADOR HÍBRIDO *FUZZY*-  
PID PARA ESTABILIZAÇÃO DE UM VANT DO TIPO  
QUADRIRROTOR NA REALIZAÇÃO DE UM VOO AUTÔNOMO  
VERTICAL**

**MOSSORÓ – RN**

**2017**

**EDPO RODRIGUES DE MORIAS**

**DESENVOLVIMENTO DE UM CONTROLADOR HÍBRIDO *FUZZY*-  
PID PARA ESTABILIZAÇÃO DE UM VANT DO TIPO  
QUADRIRROTOR NA REALIZAÇÃO DE UM VOO AUTÔNOMO  
VERTICAL**

Dissertação de mestrado acadêmico apresentada ao Programa de Pós-Graduação em Sistemas de Comunicação e Automação, como requisito para a obtenção do título de Mestre em Sistemas de Comunicação e Automação.

Orientador: Prof. Dr. Sc. Elmer Rolando Llanos Villarreal  
– UFERSA

**MOSSORÓ – RN**

**2017**

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

### FICHA CATALOGRÁFICA

827d    Morais, Edpo Rodrigues de.  
Desenvolvimento de um controlador híbrido Fuzzy-PID para estabilização de um VANT do tipo quadricóptero na realização de um voo autônomo vertical / Edpo Rodrigues de Morais. - 2017.  
118 f. : il.

Orientador: Elmer Rolando Llanos Villarreal.  
Dissertação (Mestrado) - Universidade Federal Rural do Semi-árido, Programa de Pós-graduação em Sistemas de Comunicação e Automação, 2017.

1. Quadricóptero. 2. Controle híbrido Fuzzy-PID. 3. Controle PID. 4. Voo vertical. 5. Estabilização de altura. I. Villarreal, Elmer Rolando Llanos , orient. II. Título.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

**EDPO RODRIGUES DE MORAIS**

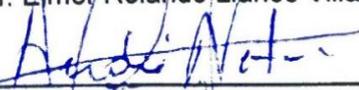
**DESENVOLVIMENTO DE UM CONTROLADOR HÍBRIDO *FUZZY*-  
PID PARA ESTABILIZAÇÃO DE UM VANT DO TIPO  
QUADRIRROTOR NA REALIZAÇÃO DE UM VOO AUTÔNOMO  
VERTICAL**

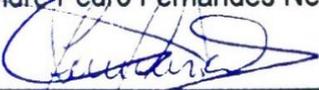
Dissertação de mestrado acadêmico apresentada ao Programa de Pós-Graduação em Sistemas de Comunicação e Automação, como requisito para a obtenção do título de Mestre em Sistemas de Comunicação e Automação.

APROVADA EM: 17 / 02 / 2017

Assinatura dos membros da Banca Examinadora:

  
\_\_\_\_\_  
Prof. Dr. Elmer Rolando Llanos Villarreal (Presidente e Orientador)

  
\_\_\_\_\_  
Prof. Dr. André Pedro Fernandes Neto (Examinador Interno - UFERSA)

  
\_\_\_\_\_  
Prof.<sup>a</sup> Dra. Tania Luna Laura (Examinadora Interna – UFERSA)

  
\_\_\_\_\_  
Prof. Dr. Marinaido Pinheiro De Sousa Neto (Examinador Externo - IFRN)

**MOSSORÓ – RN**

**2017**

## DEDICATÓRIA

A meu pai José de Arimateia Morais, minha mãe Maria da Conceição Rodrigues de Morais e a meu irmão Eduardo Rodrigues de Morais, que sempre me apoiaram dando suporte para em freter os obstáculos da vida e me permitindo chegar até aqui.

Á Fernanda Sabrina Rodrigues Vieira, que foi minha companheira ao longo dessa jornada e sempre esteve presente me ajudando quando eu mais precisava.

## AGRADECIMENTOS

Primeiramente a Deus por interceder por nós, dando o seu único filho para nos salvar e nos guiar nessa longa jornada da vida. Obrigado, meu senhor, por me dar condições de chegar até aqui, que o senhor continue olhando por nós e nos possibilite caminhar sempre em frente.

Aos meus pais, José de Arimateia Moraes e Maria da Conceição R. de Moraes e a meu irmão Eduardo R. de Moraes por todo amor e dedicação que tiveram ao longo de minha vida, vocês são o meu melhor presente, tudo que sou hoje eu devo a vocês, saibam que tenho muito orgulho de ser seu filho e os amo muito.

Um agradecimento especial a Fernanda S. R. Vieira, que me acompanhou durante todo o meu período acadêmico e me deu suporte emocional, sendo companheira, amiga, paciente e por me ajudar a levantar nos momentos difíceis e incentivar a continuar lutando.

Agradeço a todos aqueles que de maneira direta ou indireta me ajudaram a superar os obstáculos dessa jornada assim como a todos os docentes do PPGSCA que contribuíram na minha formação acadêmica.

Ao meu orientador Prof. Dr. Elmer Rolando Llanos Villarreal Andrade, pela confiança e incentivo.

Ao Prof. Dr. Alexandre Santos Brandão, por me ajudar a desenvolver o projeto, disponibilizando sua plataforma e conhecimento, meu muito obrigado.

Aos professores Dr. André Pedro Fernandes Neto, Dr(a). Tania Luna Laura, Dr. Antônio Ronaldo Gomes Garcia e ao Dr Marinaldo Pinheiro De Sousa Neto por se disponibilizarem a fazer parte da banca examinadora, assim como dar suas contribuições.

Aos meus amigos e colegas do PPGSCA, Isaac Barros, Samanta Holanda, Arimateia Magno, Jonas Rodrigo, Juan Rafael, Ana Maria Sena, Diêgo Pires, Flávia Elionara, Gregory Luid, Marks Cruz, Allyson Fiugueira, Felipe Bezerra, Cassio Falcão, Isau, Daniel Carlos. Obrigado por todo o apoio e incentivo.

Um agradecimento a UFERSA por disponibilizar estrutura e ferramentas necessárias para o aprendizado, assim com a CAPES pela confiança e investimento com auxílio financeiro.

## EPÍGRAFE

As vezes encontramos nosso destino, pelo caminho  
que tomamos para evita-lo (Robert Stephenson  
Smyth Baden Powell)

## RESUMO

Esta dissertação propõe o desenvolvimento de um controlador híbrido *Fuzzy*-PID para o controle de estabilidade de altura de um VANT do tipo quadricóptero. O trabalho ainda contempla o desenvolvimento de um controlador PID clássico, utilizado para fins de comparação com o controlador híbrido proposto. A plataforma AuRoRA é utilizada para a realização dos testes com ambas as estruturas de controle, sendo a mesma munida dos parâmetros do Drone utilizado. Com os resultados dos testes, foi possível verificar que ambos controladores, PID clássico e *Fuzzy*-PID, são capazes de realizar o controle de altura do Drone, tendo o controle híbrido obtido algumas vantagens como a capacidade de auto ajuste mediante variações no sistema, o que dispensa as sintonias manuais dos ganhos, além de apresentar uma melhoria na estabilidade da altura do Drone.

*Palavras-Chave:* Quadricóptero, Controle híbrido *Fuzzy*-PID, Controle PID, Voo vertical, Estabilização de altura.

## ABSTRACT

This dissertation proposes the development of a *Fuzzy*-PID hybrid controller for the height stability control of a four-engine type UAV. The work also contemplates the development of a classical PID controller, used to compare with the hybrid one purposed. The AuRoRA platform is used for the realization of tests with both of the control structures, being this one provided with the parameters of the Drone used. With the test results, it was possible to conclude that both of the controllers, the classical PID and the *Fuzzy*-PID, are able to perform the Drone height control, and the hybrid control obtained some advantages such as the ability of self-adjust with the system variation, that dispenses the manual tuning of the gains, besides presenting an improvement in the Drone height stability.

*Keywords:* Four-wheel drive, Fuzzy-PID hybrid controller, PID control, Vertical flight, Height stabilization.

**LISTA DE TABELAS**

Tabela 1 – Comparativo entre os três principais tipos de VANTs. Classificação: 3 é ótimo e 1 é ruim. ....	27
Tabela 2 - Efeito individual de cada ganho sob a resposta de sistema com entrada ao degrau. ....	59
Tabela 3 - Regras desenvolvidas para o controlador <i>Fuzzy</i> supervisor.....	87
Tabela 4 - Valores desejados da missão de voo para o segundo teste com os controladores...	95

## LISTA DE FIGURAS

Figura 1 - Exemplos de aplicação de VANTs: (a) Inspeção elétrica de linhas de transmissão; (b) Fotografia aérea; (c) Operações militares. ....	26
Figura 2 - Tipos de VANTs multirrotores. (a) Quadrirotor, hexacopter e octocopter. ....	28
Figura 3 – AR.Drone Parrot 2.0. ....	28
Figura 4 – Aeronave : Quadrirotor Bréguet-Richet, 1907.....	29
Figura 5 – Aeronave :“Flying Octopus” de Georges Bothezat, 1922. ....	30
Figura 6 – Aeronave: “QuadRotor Oemichen” de Etienne Oemichen, 1922.....	30
Figura 7 - Aeronave :Convertawings Model A, 1956. ....	31
Figura 8 - Aeronave: “Curtis X-19” de Etienne Oemichen, 1922.....	31
Figura 9 - Configurações de " + " e " × " do Quadrimotor. ....	33
Figura 10 - Sistema de referência inercial $g$ e do Quadrirotor $b$ . ....	34
Figura 11 - Descrição dos movimentos básicos de voo do AR.Drone Parrot 2.0.....	37
Figura 12 -Parâmetros de ajustes importantes de um sinal controlado. ....	48
Figura 13 - Diagrama de bloco de um controlador proporcional (P). ....	49
Figura 14 - Ação proporcional mediante ao erro.....	50
Figura 15 – Exemplo de saídas de um sistema de ordem zero para uma entrada em degrau onde se aplica um controle proporcional puro com ajustes de ganho $Kp3 > Kp2 > Kp1$ . .....	50
Figura 16 - Resposta de um sistema de ordem zero para uma entrada ao degrau onde se aplica um controle integral puro.....	52
Figura 17 - Diagrama de bloco de um controlador PI. ....	53
Figura 18 - Resposta de um sistema de ordem zero para uma entrada ao degrau onde se aplica um controle proporcional-integrativo (PI).....	54
Figura 19 - Diagrama de bloco de um controlador PD.....	56
Figura 20 - Resposta de um sistema de ordem zero para uma entrada ao degrau onde se aplica um controle proporcional-derivativo (PD). ....	57
Figura 21 – Diagrama de bloco de um controlador PID.....	58
Figura 22 - Resposta de um sistema de ordem zero para uma entrada em degrau onde se aplica um controle proporcional-integrativo-derivativo (PID). ....	60
Figura 23 - Diagrama de funcionamento de um controlador Fuzzy.....	62
Figura 24 - Exemplo de funções de pertinência. ....	63
Figura 25 - Exemplo 02 de funções de pertinência. ....	64

Figura 26 - Exemplo de diferentes formas de funções de pertinência.....	65
Figura 27 - Exemplo do funcionamento do operado <i>mín</i> (conectivo <b>E</b> ) nas regras de inferência Fuzzy.....	67
Figura 28 - Exemplo do funcionamento do operado <i>máx</i> (conectivo <b>OU</b> ) nas regras de inferência Fuzzy.....	68
Figura 29 – Exemplo de um esquema de controle utilizando o controlador híbrido <i>Fuzzy</i> -PID .....	71
Figura 30 – Exemplo da visualização gráfica em 3D do plano de voo da aeronave fornecido pela plataforma AuRoRA. ....	76
Figura 31 - Algoritmo da estrutura da plataforma AuRoRA.....	77
Figura 32 - Diagrama de bloco do funcionamento em terno da plataforma AuRoRA.....	78
Figura 33 - Diagrama de bloco do controlador PID de altitude do quadrirrotor. ....	80
Figura 34 - Toolbox da lógica <i>Fuzzy</i> no Matlab.....	82
Figura 35- Diagrama de bloco do controlador <i>Fuzzy</i> supervisorío .....	83
Figura 36 - Função de pertinência da primeira entrada do controlador <i>Fuzzy</i> : Erro .....	85
Figura 37- Função de pertinência da segunda entrada do controlador <i>Fuzzy</i> : Variação da Variação do Erro.....	86
Figura 38- Função de pertinência da saída variação do ganho $\Delta Kp$ do controlador Fuzzy ....	87
Figura 39 - Representação em diagrama de bloco do controlador híbrido <i>Fuzzy</i> -PID de altitude integrado ao modelo dinâmico do quadrimotor.....	88
Figura 40 – Resultados simulados do primeiro teste com o controlador PID clássico: (a) Altitude; (b) Entradas de controle indiretas.....	91
Figura 41 – Resultados simulados do primeiro teste com o controlador PID clássico: (a) Movimento no eixo x; (b) Movimento no eixo y; (c) Ângulo Phi; (d) Ângulo Psi; (e) Ângulo Theta.....	92
Figura 42 – Resultados simulados do primeiro teste com o controlador <i>Fuzzy</i> -PID: (a) Altitude; (b) Entradas de controle indiretas. ....	93
Figura 43 – Resultados simulados do primeiro teste com o controlador <i>Fuzzy</i> -PID: (a) Movimento no eixo x; (b) Movimento no eixo y; (c) Ângulo Phi; (d) Ângulo Psi; (e) Ângulo Theta.....	94
Figura 44 – Resultados simulados do segundo teste com o controlador PID clássico: (a) Altitude; (b) Entradas de controle indiretas.....	95

Figura 45 – Resultados simulados do segundo teste com o controlador PID clássico: (a) Movimento no eixo x; (b) Movimento no eixo y; (c) Ângulo Phi; (d) Ângulo Psi; (e) Ângulo Theta.....	96
Figura 46 – Resultados simulados do segundo teste com o controlador <i>Fuzzy</i> -PID: (a) Altitude; (b) Entradas de controle indiretas .....	97
Figura 47 – Fotos do momento de calibragem dos sensores do Drone .....	98
Figura 48 – Resultados experimentais do primeiro teste com o controlador PID clássico: (a) Altitude; (b) Entradas de controle indiretas. ....	99
Figura 49 – Resultados experimentais do primeiro teste com o controlador PID clássico: (a) Movimento no eixo x; (b) Movimento no eixo y; (c) Ângulo Phi; (d) Ângulo Psi; (e) Ângulo Theta.....	100
Figura 50 – Resultados experimentais do primeiro teste com o controlador <i>Fuzzy</i> -PID: (a) Altitude; (b) Entradas de controle indiretas. ....	101
Figura 51 – Resultados experimentais do segundo teste com o controlador PID clássico: (a) Altitude; (b) Entradas de controle indiretas. ....	102
Figura 52 – Resultados experimentais do segundo teste com o controlador PID clássico: (a) Movimento no eixo x; (b) Movimento no eixo y; (c) Ângulo Phi; (d) Ângulo Psi; (e) Ângulo Theta.....	103
Figura 53 – Resultados experimentais do segundo teste com o controlador <i>Fuzzy</i> -PID: (a) Altitude; (b) Entradas de controle indiretas. ....	104
Figura 54 – Resultados experimentais do segundo teste com o controlador <i>Fuzzy</i> -PID: (a) Movimento no eixo x; (b) Movimento no eixo y; (c) Ângulo Phi; (d) Ângulo Psi; (e) Ângulo Theta.....	105
Figura 49 – Comparação entre os resultados simulados do primeiro teste com controlador PID e o <i>Fuzzy</i> -PID. ....	107
Figura 50 – Comparação entre os resultados simulados do segundo teste com controlador PID e o <i>Fuzzy</i> -PID. ....	108
Figura 51 – Comparação entre os resultados práticos do primeiro teste com controlador PID e o <i>Fuzzy</i> -PID. ....	110
Figura 52 – Comparação entre os resultados práticos do segundo teste com controlador PID e o <i>Fuzzy</i> -PID. ....	111

**LISTA DE ABREVIATURAS E SIGLAS**

AuRoRA	<i>Autonomous Robots for Research and Application</i>
BLDC	<i>Brushless Direct Current</i>
CC	Corrente Contínua
MATLAB	<i>Matrix Laboratory</i>
P	Controlador Proporcional
PI	Controlador Proporcional e Integral
PD	Proporcional e Derivativo
PID	Controle Proporcional, Integral e Derivativo
PPGSCA	Programa de Pós-Graduação em Sistemas de Comunicação e Automação;
RPM	Rotação Por Minuto
UFERSA	Universidade Federal Rural do Semiárido;
VANT	Veículo Aéreo não Tripulado
GPS	Sistema Sensorial de Posicionamento Global
UAV	<i>Unmanned Aerial Vehicle</i>

## LISTA DE SÍMBOLOS

${}^g\dot{\eta}_z^T$	Formato padrão de escrita dos símbolos
$\eta$	Vetor das coordenadas rotacionais
$g$	Gravidade
$T$	Transposta
$+$	Estrutura da aeronave em formato de Cruz
$\times$	Estrutura da aeronave em forma de cruz com a frente deslocada em 45°
$\langle g \rangle$	Sistema de referência inercial,
$\langle b \rangle$	Sistema de referência do veículo
${}^g\mathbf{q}$	Vetor das coordenadas generalizadas
$\xi$	Vetor de coordenadas translacionais,
$x \ y \ z$	Posição de um veículo no espaço
$\phi \ \theta \ \psi$	Orientação de um veículo no espaço (Ângulo de Rolagem, Arfagem e Guinada, respectivamente)
${}^b\mathbf{v}$	Vetor das coordenadas generalizadas da velocidade do Drone
$Y$	Vetor das velocidades lineares do veículo
$\Omega$	Vetor das velocidades angulares do veículo
$u$	Velocidade linear na direção do eixo x
$v$	Velocidade linear na direção do eixo y
$w$	Velocidade linear na direção do eixo z
$p$	Velocidade angula em torno do eixo x
$q$	Velocidade angula em torno do eixo y
$r$	Velocidade angula em torno do eixo z
$f_1 \ f_2 \ f_3 \ f_4$	Forças de propulsão gerada por cada motor perpendicular ao plano
$J(\cdot)$	Matriz jacobiana
$\mathcal{R}$	Matriz de rotação
$W_\eta$	Matriz de Euler
$L$	Função Lagrangiano
$K_{Total}$	Energia cinética total do sistema
$P$	Energia potencial do sistema
$\dot{\mathbf{q}}$	Primeira derivada das coordenadas generalizadas com respeito ao tempo
${}^g\mathbf{f}$	Representa o vetor de forças aplicadas ao sistema

${}^g\boldsymbol{\tau}$	Representa o vetor de torques aplicadas ao sistema;
$m$	Massa total do quadrimotor
$\mathcal{A}_t$	Matriz de arrasto
$b_i$	Coefficiente de empuxo do motor
$\omega_i^2$	Velocidade angular desenvolvida pelo motor
$U_1$	Entrada indireta do controle de altitude
$U_2$	Entradas indiretas do controle de ângulo de rolagem
$U_3$	Entradas indiretas do controle de ângulo de arfagem
$U_4$	Entradas indiretas do controle de ângulo de guinada
$\mathbf{I}$	Matriz de inércia
$\mathbf{M}_r$	Matriz de inercia rotacional
$l$	Distância entre o centro do quadrimotor e o centro da hélice
$d$	Representa o coeficiente de arrasto
$\mathbf{C}_r(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})$	Matriz de Coriolis e forças centrípetas
$\mathbf{G}$	Vetor de forças gravitacionais
$\mathbf{D}_t \mathbf{D}_r$	Vetor de distúrbio
$u$	Sinais de controle
$e$	Sinal de erro atuante
$(s)$	Variável está no domínio da frequência
$(t)$	Variável está no domínio do tempo
$E(s)$	Erro atuantes no domínio da frequência
$e(t)$	Erro atuantes no domínio do tempo
$U_p(s)$	Termo proporcionais no domínio da frequência
$u_p(t)$	Termo proporcionais no domínio do tempo
$K_p$	Ganho proporcional
$K_i$	Ganho integral
$K_d$	Ganho derivativo
$t_{sq}$	Tempo de amostragem
$\Delta K_p$	Varição dos ganhos proporcionais
$\Delta K_i$	Varição dos ganhos integrativos
$\Delta K$	Varição dos ganhos derivativos

## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>19</b>
1.1. DEFINIÇÃO DOS PROBLEMAS .....	20
1.2. OBJETIVOS .....	21
<b>1.2.1. Objetivos Gerais .....</b>	<b>21</b>
<b>1.2.2. Objetivos Específicos.....</b>	<b>21</b>
1.3. JUSTIFICATIVA E LIMITAÇÕES.....	22
1.4. CONTRIBUIÇÕES .....	23
1.5. ORGANIZAÇÃO GERAL DA DISSERTAÇÃO .....	23
<b>2. REVISÃO DA LITERATURA .....</b>	<b>25</b>
2.1. INTRODUÇÃO AOS VANTs .....	26
<b>2.1.1. Histórico dos quadricópteros .....</b>	<b>29</b>
2.2. MODELAGEM DINÂMICA DE UM VANT DO TIPO QUADRICÓPTER.....	32
<b>2.2.1. Sistema de referência.....</b>	<b>33</b>
<b>2.2.2. Princípio de funcionamento.....</b>	<b>36</b>
<b>2.2.3. Modelo cinemático.....</b>	<b>38</b>
<b>2.2.4. Modelo de Euler-Lagrange.....</b>	<b>39</b>
2.3. TÉCNICAS DE CONTROLE APLICADAS A VANTs .....	46
<b>2.3.1. Controlador de três termos (PID) .....</b>	<b>46</b>
<b>2.3.2. Controlador <i>Fuzzy</i> .....</b>	<b>61</b>
<b>2.3.3. Controlador Híbrido <i>Fuzzy</i>-PID .....</b>	<b>70</b>
<b>3. MATERIAIS E MÉTODOS.....</b>	<b>73</b>
3.1. AR.DRONE PARROT 2.0 .....	73
3.2. PLATAFORMA AuRoRA .....	75
<b>3.2.1. Estrutura e funcionamento da plataforma AuRoRA.....</b>	<b>76</b>
3.3. DESENVOLVIMENTO DO CONTROLADOR HÍBRIDO <i>FUZZY</i> -PID .....	79
<b>3.3.1. Construção do controlador PID.....</b>	<b>79</b>

3.3.2. Desenvolvimento do controlador <i>Fuzzy</i> supervisor.....	81
<b>4. RESULTADOS E DISCUSSÃO .....</b>	<b>89</b>
4.1. RESULTADOS COMPUTACIONAIS OU SIMULADOS.....	90
4.2. RESULTADOS DOS TESTES PRÁTICOS.....	97
4.3. COMPARAÇÃO COM CONTROLADOR CLÁSSICO PID .....	105
<b>5. CONCLUSÕES E TRABALHOS FUTUROS .....</b>	<b>112</b>
<b>6. REFERÊNCIAS .....</b>	<b>115</b>

## 1. INTRODUÇÃO

Nas últimas décadas, o esforço dedicado à pesquisa referente aos Veículos Aéreos Não Tripulados (VANTs) teve um acréscimo significativo, fato este decorrente da sua vasta aplicabilidade que vai desde uso popular para o entretenimento até operações complexas de busca e salvamento.

Um dos tipos de VANTs mais utilizados é o quadricóptero popularmente conhecido como Drone, originado do inglês que significa Zangão<sup>1</sup>. Esse veículo possui uma estrutura básica constituída por dois eixos dispostos em forma cruciforme contendo em cada extremidade um motor e uma hélice que são os responsáveis pela geração do empuxo da aeronave, o que permite a capacidade de decolar e pousar verticalmente, realizar manobras precisas em altas e baixas velocidades além de realizar voos pairados (*hovering*).

O controle desse tipo de veículo pode ser realizado manualmente através de um rádio controle ou de forma autônoma por intermédio de controladores embarcados, sendo que o segundo caso é o principal foco de estudo. Para conseguir uma navegação autônoma de um quadricóptero, é necessário a construção de um controlador que garanta tanto a estabilidade quanto o desempenho do VANT, dessa forma, uma série de estratégias de controle se encontram dispostas na literatura, tais como: Controlador clássico Proporcional Integrativo Derivativo (PID), rede neurais artificiais, algoritmos genéticos, controle Backstepping e Lógica *Fuzzy*.

Dentro desse contexto esta dissertação propõe o desenvolvimento de um controlador híbrido *Fuzzy*-PID ao qual incorpora tanto as características do controle clássico PID quanto às vantagens do sistema de lógica *Fuzzy* para ajuste de seus parâmetros, com a junção dessas duas características o projeto tende a promover a estabilidade de altura de um quadricóptero na realização de manobras autônomas de decolagem e aterrissagem (controle de altitude). Além disso, essa dissertação irá mostrar um breve histórico sobre os quadricópteros, o princípio de funcionamento de um VANT, apresentar o modelo matemático que represente a dinâmica do sistema de um quadrimotor segundo o formalismo proposto por Euler-Lagrange, realizar os testes para verificar a funcionalidade do controlador proposto, simulando-o em ambiente computacional através da plataforma AuRoRA, realizar os testes reais do controlador aplicando-o ao VANT Ar.Drone Parrot 2.0 e comparar os valores medidos e simulados com controlador PID clássico (também desenvolvida nesse trabalho) evidenciando as melhorias e as possíveis limitações do controlador *Fuzzy*-PID desenvolvido.

---

<sup>1</sup> Apelido dado a esta aeronave devido ao barulho que ele faz que se assemelha ao voo de uma abelha.

## 1.1. DEFINIÇÃO DOS PROBLEMAS

A problemática em se construir um controlador para um quadricóptero, está no fato de que esse tipo de veículo apresenta uma natureza instável e um número considerável de variáveis a serem controladas, sendo também susceptíveis a perturbações externas (SUZUKI, 2009). Além disso, esses veículos apresentam um sistema subatuado onde o número de atuadores (os quatro motores) é menor que o número de graus de liberdade, isso faz com que seus movimentos, no espaço tridimensional, requeiram a correlação de dois ou mais atuadores (BRANDÃO, 2013). Outro problema está no fato de que com a miniaturização do VANT para redução de custos, acaba implicando em limitações de desempenho o que torna a estrutura mais difícil de ser controlada (BOUABDALLAH, 2007). A não-linearidade torna o processo de controle desse tipo de veículo ainda mais complexo, o que requer uma ferramenta de simulação completa, capaz de reproduzir o máximo de variantes possível para que se possa validar o controlador e reduzir o tempo de projeto.

A construção do controlador híbrido *Fuzzy*-PID traz as vantagens de ambas às técnicas de controle, porém também requer o conhecimento aprofundado em ambos os assuntos, o que exige do programador a experiência na elaboração das funções de pertinência, das técnicas de fuzzificação, defuzzificação e o conhecimento heurístico da planta do sistema, para que dessa forma possa desenvolver o conjunto de regras que seja suficiente para fazer com que o *Fuzzy* supervisor realize com eficiência o controle dos ganhos  $K_p$ ,  $K_i$  e  $K_d$ , sem que isso prejudique o desempenho do controlador PID.

Esse projeto abrange ainda uma análise experimental do controlador proposto, onde o mesmo é aplicado a um Drone real, realizando inúmeros testes para evidenciar a sua funcionalidade e comprovar sua validação. Como a estrutura é aplicada a um ambiente fechado (indoor) os testes são limitados pelo espaço confinado de uma sala, o que leva a um aumento substancial para obtenção dos resultados com os testes práticos, já que o pouco espaço e os obstáculos dificultam as manobras executadas pelo Drone.

## 1.2. OBJETIVOS

### 1.2.1. Objetivos Gerais

Esta dissertação tem como objetivo principal desenvolver um controlador híbrido *Fuzzy*-PID para o controle de estabilidade de altura de um VANT do tipo quadrimotor.

### 1.2.2. Objetivos Específicos

- Apresentar o modelo matemático que represente a dinâmica do sistema de um VANT do tipo quadrimotor segundo a formulação de Euler-Lagrange.
- Desenvolver uma estrutura de controle híbrido *Fuzzy*-PID que permita estabilizar a altura de um Drone, quando o mesmo realiza um voo autônomo vertical (movimento de decolagem e aterrissagem) mantendo-se em uma postura constante por um determinado período de tempo.
- Desenvolver um controlador PID puro, para fins de comparação com estrutura híbrida proposta.
- Encontrar os valores iniciais de ganhos  $K_p$ ,  $K_i$  e  $K_d$ , que melhor sintoniza o controlador PID.
- Desenvolver um controlador *Fuzzy*, utilizando *Fuzzy Logic Toolbox* do MATLAB, que atenda as características do sistema e consiga de forma autônoma encontrar os valores de ganho que melhor ajusta o controlador PID.
- Realizar testes de simulação para verificar a funcionalidade do controlador, simulando-o através da plataforma AuRoRa (concedida para aplicação desta dissertação).
- Realizar os testes práticos com o Drone, aplicando o controle clássico PID e a variação híbrida *Fuzzy*-PID proposta neste trabalho.
- Comparar os valores medidos e simulados com trabalhos já existentes na literatura, evidenciando as melhorias e as possíveis limitações do controlador proposto quando comparado ao controlador clássico PID.

### 1.3. JUSTIFICATIVA E LIMITAÇÕES

A primeira justificativa para o estudo de controle dos VANTs de asas rotativas (do tipo quadricóptero) deve-se ao fato das inúmeras vantagens aos quais esses veículos apresentam, tais como: a miniaturização que permite locomover-se em locais com pouco espaço, sua capacidade de realizar manobras precisas em baixa e alta velocidade, poder levantar voo vertical, aterrissar em locais de difícil acesso, não requer tripulação, além de possuir a capacidade realizar voos pairados, o que permite expor uma outra justificativa que é a extensa área de aplicações aos quais esses veículos podem ser submetidos, seja para fins de segurança, vigilância, militar, agricultura ou até mesmo entretenimento.

Na literatura existem diversos estudos referentes à construção, modelagem e controles de VANTs. No geral, o foco principal dessas pesquisas estão nas técnicas de controle aplicadas para navegação autônoma desses veículos. Esse estudo é caracterizado por muitos pesquisadores como sendo o desafio tecnológico mais recente, provocando o interesse de alunos das áreas de controle linear, não linear e inteligente (BRANDÃO, 2013).

Como justificativa adicional, o projeto propõe o desenvolvimento de uma estrutura de controle híbrida envolvendo duas estratégias de controle, o PID clássico e a lógica *Fuzzy*, afim de realizar o controle de altitude de um Drone, permitindo demonstrar as vantagens e possíveis melhorias desse controlador híbrido quando comparado ao controle puramente PID.

Vale mencionar ainda como justificativa, trabalhos existentes na literatura que obtiveram êxito no controle de VANTs, contudo, é importante ressaltando que a técnica de controle híbrida *Fuzzy*-PID, aplicada ao controle de VANTs de asas rotativas, mostrou-se até o momento, a existência de poucos trabalhos desenvolvidos sobre esse assunto. Dentre as pesquisas bibliográficas estudadas destacam-se alguns trabalhos que obtiveram êxito na construção de controladores para VANTs e que deram suporte na elaboração do projeto desta dissertação. Trabalhos como Brandão (2013), em que é proposto o projeto e implementação de controladores não lineares de alto nível, capazes de guiar um VANT de pás rotativas em voos autônomos. Em Sá (2012), é proposto a construção de um VANT do tipo quadricóptero, seu modelo dinâmico, assim como a elaboração de um controlador PID clássico e uma variação desse controlador para estabilização desse veículo. Em SUZUKI (2009), é proposto um controlador híbrido *Fuzzy*-PID para estabilização de um VANT de asas fixas.

No que se refere às limitações, pode-se dizer que este trabalho abrange ainda uma área inicial do controle de estabilidade de Drones, sendo limitado somente ao controle de altitude, tendo em vista que o controle de atitude requer um nível maior de controle e um controlador

mais robusto, sendo propostas para trabalhos futuros. Ainda nas limitações, o âmbito deste trabalho se dá em ambiente interno confinado, na tentativa de inibir o máximo de perturbações externas, fato este que impede o Drone de usar o seu sistema sensorial de posicionamento global (GPS), devido as incertezas inerentes as paredes das estruturas da sala, sendo que o mesmo passa a ser orientado pelo sensor ultrassônico, limitando seu controle para baixas altitudes.

#### 1.4. CONTRIBUIÇÕES

Este trabalho da continuidade as pesquisas referente a controle de estabilidade de VANTs na Universidade Federal Rural do Semi-Árido (UFERSA). Esta área de pesquisa ainda se encontra muito difusa na instituição, logo a contribuição principal desse trabalho é incorporar ao acervo acadêmico duas estruturas robustas de controle, capazes de acrescentar novas possibilidades em estudo e aplicação para esse tipo de veículo (VANTs). O projeto foca em uma proposta ainda pouco esplanada nesta linha de estudo, que é a implementação de uma estrutura híbrida de controle, ao qual envolve o trabalho em conjunto de dois tipos de controladores (controlador PID clássico, e o controle *Fuzzy*), além do desenvolvimento (para fins de comparação) de uma estrutura clássica denotada pelo controlador PID clássico.

Na prática, o trabalho apresenta um teste pioneiro, tendo em vista que até o presente momento não havia na instituição projetos relacionados dos quais tenham realizado o teste prático com o VANT, uma vez que, toda a arquitetura do projeto foi simulada, validada e aplicada em uma aeronave real (o VANT de teste foi o AR.Drone Parrot 2.0).

No que se refere ao conhecimento explanado nesta dissertação, a mesma é munida de todo um referencial teórico a respeito do assunto envolvendo as duas estratégias de controle (PID, *Fuzzy* e o *Fuzzy*-PID), nela é mostrada todo o processo de elaboração dos controladores (quantidade de regras, funções de pertinência, etc.) as dificuldades e soluções adotadas para a elaboração do projeto, permite que o conhecimento e experiências concebidas possam ser reproduzidas e ampliadas em novas possibilidades de estratégias de controle.

#### 1.5. ORGANIZAÇÃO GERAL DA DISSERTAÇÃO

Esta dissertação está dividida em cinco capítulos principais: introdução, revisão de literatura, materiais e métodos, resultados e discussão, conclusões e trabalhos futuros. Tais capítulos dividem as etapas do desenvolvimento do projeto, apresentando como foi dado a elaboração do controlador proposto e as estratégias de controle utilizadas. Os capítulos foram distribuídos da seguinte forma:

No Capítulo 2 é destinado ao levantamento teórico dos principais assuntos que serão tomados para o desenvolvimento do projeto desta dissertação. Dessa forma, o objetivo principal deste capítulo é fazer interligação do leitor com as informações necessárias para a clara compreensão do projeto. Nele é feita uma breve introdução sobre os VANTs dando ênfase ao modelo de estrutura quadrirrotórica e aos motivos que levaram a escolha desse tipo de aeronave, além de mostrar um breve histórico sobre eles.

O Capítulo ainda contempla a modelagem dinâmica de um quadrirrotor, mostrando os aspectos importantes para sua representação matemática, seguindo formalismo de Euler-Lagrange (modelo utilizado pela plataforma AuRoRA), além é claro da apresentação individual das técnicas de controle utilizadas para o desenvolvimento do controlador proposto, tais como o controle clássico PID, o controlador *Fuzzy* supervisor e a junção dos dois formando controle híbrido *Fuzzy*-PID.

No Capítulo 3 é descrito quais foram os materiais utilizados e os métodos empregados para o desenvolvimento do controlador proposto, tais como o modelo da aeronave utilizada, funcionamento da plataforma de teste, além dos detalhes de elaboração do projeto, onde é mostrada todas as estratégias empregadas na sua elaboração, tais como: as funções de pertinências, regras de inferência, variáveis de entrada e saída, funcionamento do controlador dentre outros pontos importantes.

O Capítulo 4 descreve os resultados obtidos com a implantação do controlador desenvolvido nesta dissertação, nele é feita a apresentação dos resultados dos testes iniciais em simulação através da plataforma AuRoRa, analisando não só o comportamento do controlador *Fuzzy*-PID proposto mais também a de um controlador PID clássico, de forma a realizar um comparativo entre as duas estruturas de controle a fim de evidenciar possíveis melhorias em relação a estrutura híbrida desenvolvida. Neste Capítulo ainda está contido os resultados práticos da aplicação com o Drone real, validando ambas as estruturas de controle.

No Capítulo 5 é descrito as conclusões obtidas no decorrer deste trabalho evidenciando as dificuldades e superações obtidas com esse projeto. Destacam-se também os pontos importantes que foram observados e o potencial do controlador, possibilitando a indagação de possíveis melhorias e novas implementações como propostas para trabalhos futuros.

## 2. REVISÃO DA LITERATURA

Este capítulo foi dividido em três tópicos principais, cada um com suas respectivas seções.

No tópico 2.1 é realizada uma introdução textual, com objetivo de familiarizar o leitor com o objeto de estudo desta dissertação, neste caso, trata-se de um veículo aéreo não tripulado, mais especificamente um quadrimotor, com isso, este tópico apresenta um breve histórico sobre o mesmo.

No tópico 2.2 é mostrado a modelagem dinâmica de um quadrimotor, a fim de se obter o modelo matemático que possa representar o movimento desses veículos no espaço tridimensional. Dessa forma, o objetivo deste tópico é mostrar ao leitor algumas das equações matemáticas que estão contidas na plataforma AuRoRA (vale ressaltar, que esta plataforma foi disponibilizada pelo seu idealizador para uso estritamente didático desta dissertação, sendo utilizada no desenvolvimento do controlador proposto). Este tópico ainda se divide em três seções, na seção 2.2.1 é determinado o sistema de referência do quadrimotor, que será utilizado nas equações do modelo dinâmico do sistema. Na seção 2.2.2 é dedicada à descrição do princípio de funcionamento do quadrimotor, mostrando sua possibilidade de manobras no espaço tridimensional e familiarizando o leitor com a capacidade de voo desses veículos. Na seção 2.2.3 é mostrado o modelo cinemático do quadrimotor onde é possível ver como é feita a transformação das velocidades rotacionais no referencial global para o referencial do inercial do veículo. Por fim a seção 2.2.4 é destinada a demonstração matemática das equações que descreve o movimento de um corpo rígido no espaço tridimensional, através do formalismo descrito por Euler-Lagrange.

No último tópico 2.3 mostra as características das técnicas de controle que foram utilizadas no projeto dessa dissertação. O objetivo desse tópico é mostrar ao leitor as técnicas de controle que foram utilizadas no projeto, mostrando as vantagens e desvantagens das mesmas. Por se tratar de um controle híbrido que envolve a junção de duas técnicas de controle, optou-se por dividir este tópico em três seções, na seção 2.3.1 fala sobre as características de controle do controlador de três termos ou controlador clássico PID. Já na seção 2.3.2, é abordado os conceitos que range o controlador baseado em lógica *Fuzzy*. Por fim a seção 2.3.3 é dedicada à explanação da junção dessas duas técnicas de controle, dando origem a um novo controlador, o controlador híbrido *Fuzzy*-PID.

## 2.1. INTRODUÇÃO AOS VANTs

Um Veículo Aéreo Não Tripulado (VANT), traduzido do inglês UAV (Unmanned Aerial Vehicle), é todo e qualquer tipo de aeronave que não necessita de piloto embarcado para controlá-lo. Esses veículos têm nas últimas décadas desenvolvido um interesse crescente tanto na área científica quanto industrial. Diversas empresas pelo mundo vêm dedicando tempo e disponibilizando recursos para o aprimoramento das pesquisas relacionadas à navegação dessas aeronaves. A vantagem de não requerer tripulação, faz dos VANTs uma opção promissora em realização de tarefas que podem trazer risco a saúde do ser humano. São inúmeras possibilidades de uso, a seguir estão destacadas algumas das atividades aos quais esses veículos podem realizar, tarefas como: Vigilância costeira, vigilância de multidões, gerenciamento de tráfego urbano, controle de patrimônio, monitoramento agrícola, mapeamento e monitoramento de desastres, gerenciamento de riscos naturais, vigilância marítima, espionagem, vigilância e cobertura de eventos, inspeção de incêndios, operações antiterroristas, inspeção elétrica, monitoramento de banhistas (salva-vidas), controle de pestes (mosquitos, gafanhotos, larvas) dentre outras. Na Figura 1 ilustram alguns desses exemplos de aplicação dos VANTs. (PIZETTA, 2013; PIZETTA, et al. 2012).

Figura 1 - Exemplos de aplicação de VANTs: (a) Inspeção elétrica de linhas de transmissão; (b) Fotografia aérea; (c) Operações militares.



(a)



(b)



(c)

**Fonte:** PIZETTA, 2013.

Atualmente existem vários modelos de VANTs disponíveis no mercado, eles podem variar em forma, tamanhos e funcionamento, podendo ser mais leve que o ar, tais como dirigíveis e balões ou mais pesados que o ar, esses ainda se dividem em duas categorias que são os de asas fixas, como aviões e planadores, e os de asas rotativas, tais como helicóptero e quadrimotores. Cada veículo possui suas particularidades, das quais podem se tornar vantagem ou desvantagem, dependendo de sua aplicação, exemplo, os VANTs de asas fixas possuem a

vantagem de realizar voos em alta velocidade, que são ótimos para missões de longo alcance, no entanto necessitam de um amplo espaço para decolagem, aterrissagem e manobras, impossibilitando seu uso em ambientes fechados (*indoor*) com pouco espaço, já os VANTs mais leves que o ar, possuem a vantagem do baixo consumo de energia, além da capacidade de decolar e pousar verticalmente, realizar voos pairados e voos em baixa velocidade (bons na aplicação de tarefas que exigem longos períodos de observação), porém, possuem a desvantagem de serem grandes, lentos com baixa manobrabilidade o que também dificulta seu uso *indoor*. Os VANTs de asas rotativas, quando comparados aos dois modelos mencionados anteriormente, perdem apenas no quesito velocidade, para os de asas fixas, e na eficiência do consumo de energia, para os mais leves que o ar, levando vantagem nos demais quesitos como, grande flexibilidade de manobras, permitindo realizar voos precisos em médias e baixas velocidades, também possuem a capacidade de decolar e pousar verticalmente, realizar voos pairados, além da miniaturização, ao qual permite sua fabricação em diversos modelos e tamanhos possibilitando sua utilização em ambientes fechados com pouco espaço. A Tabela 1 abaixo mostra um comparativo entre esses três tipos clássicos de VANTs.

Tabela 1 – Comparativo entre os três principais tipos de VANTs. Classificação: 3 é ótimo e 1 é ruim.

<b>Características básicas de voo</b>	<b>Mais leves que o ar</b>	<b>Asas fixas</b>	<b>Asas rotativas</b>
Consumo de Energia	3	2	1
Manobrabilidade	1	2	3
Voo pairado	3	1	3
Voo em baixas velocidades	3	1	3
Voo em altas velocidades	1	3	2
Miniaturização	1	2	3
Decolagem vertical	3	1	3
Utilização <i>indoor</i>	2	1	3
<b>Total</b>	17	12	21

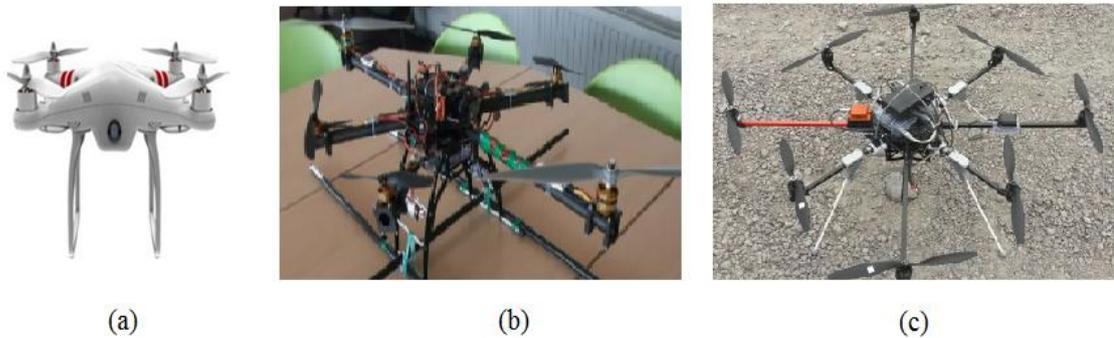
**Fonte:** Adaptado de BRANDÃO, 2013; BOUABDALLAH, 2007.

Como foi possível observar os veículos de asas rotativas reúnem o maior número de pontos favoráveis em características básicas de voo, apresentando vantagens que sobrepõem as demais aeronaves comparadas anteriormente, tais características fazem deste tipo de aeronave a escolha mais adequada para o estudo em laboratório.

Ainda dentre dos tipos de VANTs de asas rotativas, há um destaque maior na categoria de multirrotores, que são aeronaves com mais de um rotor central, tais como, quadrimotores ou

quadricoptero (que possuem quatro motores), hexacooper (que possuem seis motores) e octocopter (que possuem oito motores), conforme pode ser visto na Figura 2.

Figura 2 - Tipos de VANTs multirrotor. (a) Quadricóptero, hexacooper e octocopter.



**Fonte:** Adaptado de SILVA, 2015; VERBEKE *et al*, 2014; OSCARSON, 2015.

Com foi mencionando anteriormente esses veículos são também conhecidos como Drones, eles possuem um número maior de atuadores (motores), apresentam maior estabilidade e maior flexibilidade de manobras em comparação com os helicópteros, motivos que levaram a escolha desse tipo de VANT para a realização do trabalho desta dissertação, sendo que a estrutura escolhida foi o quadrimotor, mais precisamente o modelo AR.Drone Parrot 2.0 (veja a Figura 3).

Figura 3 – AR.Drone Parrot 2.0.



**Fonte:** PARROT, 2014.

### 2.1.1. Histórico dos quadrirrotores

A história relata que os primeiros modelos de quadrirrotor são bem antigos, que datam da época de 1907, quando ainda não se imaginava existirem motores elétricos sem escovas (motores de brushless<sup>2</sup>). O primeiro modelo foi desenvolvido por Charles Richet e os irmãos Breguet, ele apresentava uma estrutura simples, com um rotor em cada extremidade cuja potência era fornecida por gasolina e sua transmissão distribuída por meio de correias, esse veículo almejava realizar um voo vertical no transporte de uma pessoa (veja Figura 4). As primeiras tentativas de voo foram sem sucesso, devido a sua robustez, o mesmo atingiu a altura de apenas 1,5 metros. (LEISHMAN, 2002).

Figura 4 – Aeronave : Quadrirrotor Bréguet-Richet, 1907.



**Fonte:** LEISHMAN, 2000.

Anos mais tarde, em 1922, o exército dos Estados Unidos da América (EUA) contratou um imigrante russo chamado Georges de Bothezat, que desenvolveu um dos maiores helicópteros de seu tempo, tal máquina tratava-se de um quadrirrotor com estrutura central em forma de cruz, do qual possuía em todas as extremidades um rotor fixo, cujo mesmo está engrenado a um conjunto de quatro rotores menores que serviam para auxiliar no controle da máquina. Tal invento foi chamado de "Octopus Voador", essa criação apresentou bons resultados de voo, contudo, os voos eram sempre em baixas altitudes e velocidade (veja Figura 5), fato este, que gera um baixo desempenho quando comparados ao seu alto custo financeiro, o que levou ao cancelamento do projeto. (SÁ, 2012).

---

<sup>2</sup> Os motores de brushless, são motores elétricos síncronos de corrente contínua (CC) sem escovas também conhecidos como motores BLDC (*Brushless Direct Current*). (COSTA, 2008).

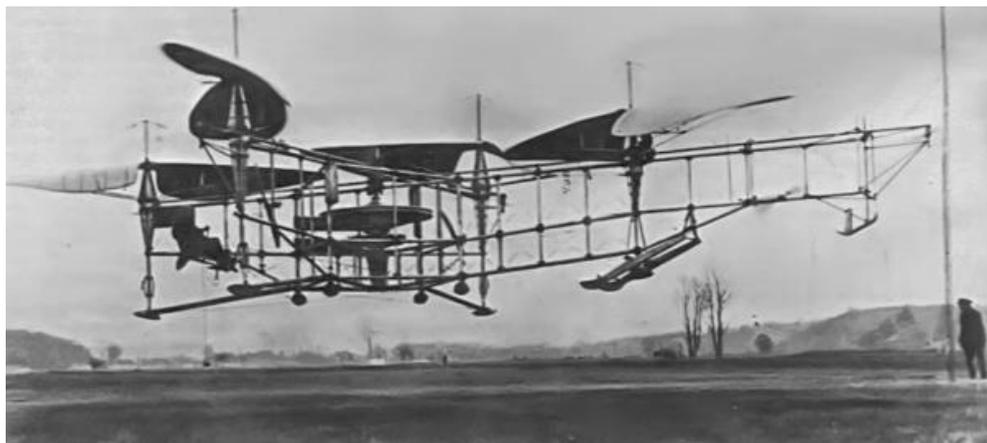
Figura 5 – Aeronave :“Flying Octopus” de Georges Bothezat, 1922.



**Fonte:** LEISHMAN, 2000.

Na mesma época, de 1920 a 1922, havia outro pesquisador que almejava o desenvolvimento de tais aeronaves (quadrirrotores), o engenheiro da Peugeot Etienne Oemichen, que já havia conseguido obter alguns experimentos com o quadrimotor. Depois de muitas tentativas e erros, no ano de 1922 surgia o que Etienne Oemichen batizou de “QuadRotor Oemichen”, uma máquina voadora que pesando seus 800kg (veja Figura 6). Oemichen ainda conseguiu provar, em 1924, que uma máquinas de voo vertical poderia ser estável e pouco manobrável apesar de seu peso, o que concedeu-lhe alguns prêmios na época (SÁ, 2012).

Figura 6 – Aeronave: “QuadRotor Oemichen” de Etienne Oemichen, 1922.



**Fonte:** BENIGNO, 2015.

Conforme surgiam novos modelos de aeronaves, outras tecnologias eram incorporadas a esses veículos a fim de se obter melhores avanços nessa linha de pesquisa, desta forma os conceitos das aeronaves de Oemichen e Bothezat deram origem a um novo projeto que foi

desenvolvido no ano de 1956, com a construção do chamado “Convertawings Model A” (Figura 7), tal projeto promoveu inúmeros voos com êxito.

Figura 7 - Aeronave :Convertawings Model A, 1956.



**Fonte:** SÁ, 2012.

Dando continuidade aos avanços, em 1960 a Curtiss-Wright Corporation desenvolveu um protótipo de aeronave chamando “Curtis X-19” (veja Figura 8) ao qual tinha como principal objetivo o transporte de passageiros. Contudo o sistema ainda apresentava dificuldades na transição entre o voo vertical e o voo horizontal devido a correlação dos ângulos referente aos seis graus de liberdade, presente neste tipo de veículo, logo a aeronave revelou não ser funcional, pois constituía de uma transição muito lenta entre os sistemas mecânicos (JENKINS *et al.*, 2012)

Figura 8 - Aeronave: “Curtis X-19” de Etienne Oemichen, 1922.



**Fonte:** JENKINS *et al.*, 2012.

Outros modelos ainda foram desenvolvidos, entretanto sem sucesso, o que leva a diminuição do interesse pelo projeto de quadrimotores, principalmente pela dificuldade de se obter uma correção mecânica eficiente para transição do movimento vertical para o longitudinal, e ainda assim, manter uma boa estabilidade de voo.

O estudo desses quadrimotores ressurgiu na década de 80 e 90 sob a forma de veículos aéreos não tripulados (VANTs) ou popularmente conhecidos como Drone, bastante utilizado para fins de espionagem e reconhecimento de campo pelas forças militares. Atualmente essas aeronaves vêm ganhando destaque na mídia sendo bastante usados na área acadêmica e no uso popular para entretenimento.

## 2.2. MODELAGEM DINÂMICA DE UM VANT DO TIPO QUADRIRROTOR

Para realizar o controle de um quadricóptero é necessário primeiro representar tal veículo por meio de um modelo matemático. Essa modelagem se baseia no equacionamento dinâmico que envolve o movimento de um corpo rígido no espaço tridimensional, que está sujeito a perturbações. Na literatura existem duas abordagens clássicas para a modelagem matemática de veículos aéreos: uma baseada no formalismo de Newton-Euler (SÁ *et al.*, 2013; ABBASI *et al.*, 2013; BOUABDALLAH, 2007) e outra no formalismo de Euler-Lagrange (BRANDÃO, 2013; BRANDÃO *et al.*, 2012a; BRANDÃO *et al.*, 2012b; BENIGNO, 2015). Ambas as técnicas podem representar o modelo dinâmico de um corpo rígido, apesar de possuírem equações diferentes, esses modelos não são excludentes, ou seja, um pode ser usado para simplificar o outro, podendo dessa forma, através de manipulação matemática, passar de uma representação para outra (BRANDÃO, 2013).

O modelo baseado no formalismo de Newton-Euler utiliza as equações de movimento da mecânica para representação do sistema físico, onde o sistema de referência está localizado no centro de massa da aeronave. Já o modelo proposto por Euler-Lagrange se baseia nos dados de excitação e de resposta do sistema, em que seu referencial coincide com o referencial inercial (BRANDÃO, 2013).

Para o desenvolvimento e validação do controlador proposto nesta dissertação, foi necessário a utilização da plataforma AuRoRA (utilizada nos trabalhos de Brandão, 2013, e Pizetta, 2013), cujo seu algoritmo de programação foi disponibilizada para uso didático nesta dissertação. Essa plataforma foi desenvolvida utilizando as equações segundo o modelo proposto por Euler-Lagrange, dessa forma, esta seção será destinada a fazer uma apresentação

sobre algumas considerações e equações que estão contidas nesta plataforma, incluindo a modelagem dinâmica utilizada para o quadrimotor.

### 2.2.1. Sistema de referência

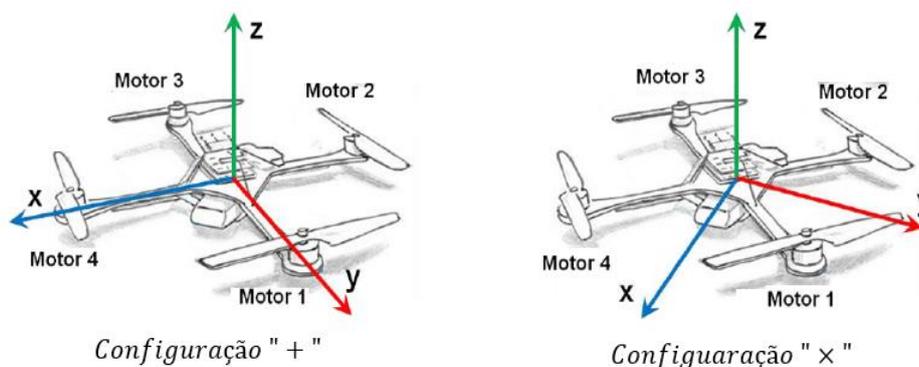
Esse tópico é dedicado para determinar o sistema de referência que será adotado para a modelagem do quadrimotor no espaço tridimensional. Vale ainda ressaltar que, devido à complexidade de se trabalhar com um sistema com 6 (seis) graus de liberdade, existem na literatura diversas nomenclaturas/notações que são utilizadas para representar tal modelagem (POUNDS, 2007), mediante este fato e visando a melhor compreensão do leitor, esta dissertação optou-se por adotar o seguinte padrão de escrita:

$$g\dot{\eta}_z^T \quad (1)$$

Onde o vetor das coordenadas rotacionais ( $\eta$ ) é a variável base, a quantidade de pontos sobre a variável base indica a ordem da derivada, neste caso é a primeira derivada de  $\eta$ . O termo sobrescrito na margem superior esquerda indica o sistema de referência adotado ( $g$ ), o sobrescrito na margem superior direita indica o valor do expoente ou uma especificação textual, neste caso o  $T$  simboliza a transposta, por fim o sobrescrito na margem inferior direita,  $z$ , diz que esta variável está fazendo referência à coordenada  $z$  do sistema de referência adotado.

Para que se possa determinar o sistema de referência do quadrimotor primeiramente tem-se que saber em qual configuração a estrutura do Drone está disposta, se é em forma de "+" ou em forma de "x", como demonstrado na Figura 9.

Figura 9 - Configurações de "+" e "x" do Quadrimotor.



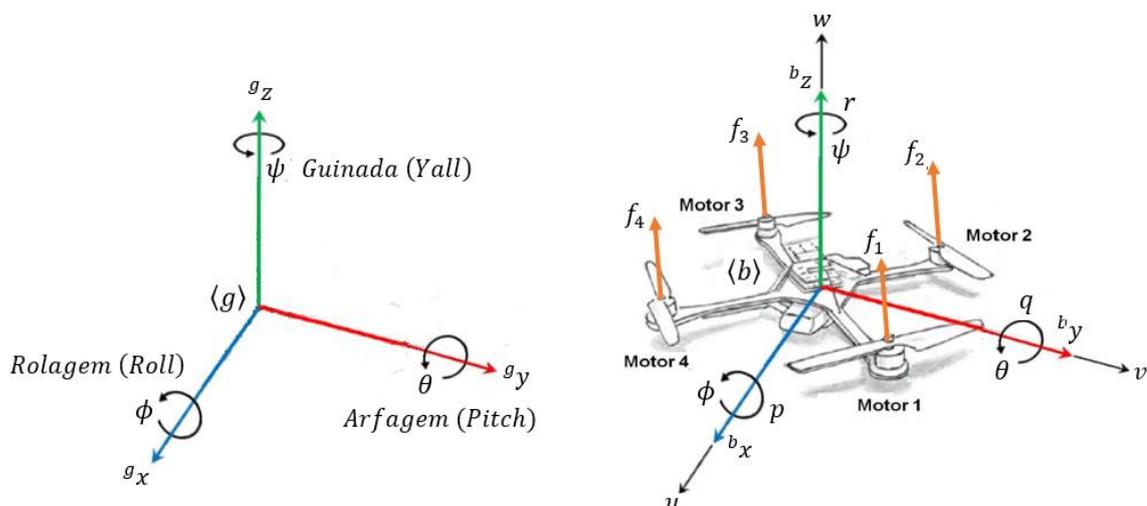
**Fonte:** Adaptado de POUNDS, 2007.

Esta identificação é importante, pois cada configuração apresenta um sistema de referência próprio para o veículo, onde ela irá determinar o sentido positivo da movimentação do Drone e quais os motores serão correlacionados para gerar o empuxo necessário a cada manobra no espaço tridimensional. Tendo em vista que esta necessidade de se ter a relação em conjunta de múltiplos motores é decorrente da dinâmica altamente acoplada presente neste tipo de veículos.

Como já mencionado em capítulos anteriores, foi utilizado para esta dissertação o quadricóptero do modelo AR.Drone Parrot 2.0. Esse Drone possui duas câmeras de captação de imagem, sendo uma apontada para baixo, no sentido negativo do eixo  $z$ , e outra apontada para frente, no sentido positivo do eixo  $x$ , localizada no centro entre os dois rotores conforme foi visto na Figura 3, logo o deslocamento longitudinal do Drone ocorre segundo a orientação da câmera frontal, sendo o eixo  $x$  deslocado  $45^\circ$  conforme visto na Figura 9, desta forma a estrutura do Drone utilizado está caracterizado na configuração "X". Sendo assim, toda a dinâmica do sistema será modelada seguindo estas especificações.

Para determinar o sistema de referência inercial, considere  $\langle g \rangle$  como sendo um ponto fixo referente a terra onde seu centro será adotado como sendo a origem do sistema cartesiano tridimensional, da mesma forma para determinar o sistema de referência do veículo, considere  $\langle b \rangle$  como sendo um ponto fixo no centro de massa do quadricóptero em que as coordenadas dos seus eixos indicarão sua postura/posição/orientação em relação ao sistema inercial da terra, conforme demonstra a Figura 10. (BRANDÃO *et al*, 2012a; BENIHNO, 2015)

Figura 10 - Sistema de referência inercial  $\langle g \rangle$  e do Quadricóptero  $\langle b \rangle$ .



**Fonte:** Adaptado de POUNDS, 2007; BRANDÃO *et al*, 2012a.

Para que se possa determinar a posição do veículo no espaço cartesiano são necessários de no mínimo seis variáveis, que corresponde aos graus de liberdade de um objeto quando se move no espaço tridimensional, logo, juntas essas variáveis formam o sistema de referência inercial  $\langle g \rangle$  do espaço tridimensional do quadrimotor que é definido por:

$${}^g\mathbf{q} = [\xi \ \eta]^T \quad (02)$$

Sendo que três variáveis são responsáveis por determinar as coordenadas translacionais, representando seu deslocamento longitudinal, lateral e normal (respectivamente), definidos pelo vetor  $\xi$ , onde:

$$\xi = [x \ y \ z]^T \in \mathbb{R}^3 \quad (03)$$

E três variáveis responsáveis por determinar as coordenadas rotacionais, representadas pelos ângulos de Euler que são os ângulos de rolagem, arfagem e guinada (respectivamente), definidos pelo vetor  $\eta$ :

$$\eta = [\phi \ \theta \ \psi]^T \in \mathbb{R}^3 \quad (04)$$

Ainda na Figura 10, agora tendo como base o referencial do veículo  $\langle b \rangle$ , é possível determinar o vetor das velocidades do quadrimotor que é definido por:

$${}^b\mathbf{v} = [\Upsilon \ \Omega]^T \quad (05)$$

Onde o vetor  $\Upsilon$  representa as velocidades lineares, dadas por:

$$\Upsilon = [u \ v \ w]^T \in \mathbb{R}^3 \quad (06)$$

E o vetor  $\Omega$  representa as velocidades angulares do veículo, dadas por:

$$\Omega = [p \ q \ r]^T \in \mathbb{R}^3 \quad (07)$$

Dessa forma, as coordenadas generalizadas (que são as variáveis independentes escolhidas para representar os graus de liberdade do sistema), que formam o sistema de referência inercial  $\langle g \rangle$  da posição do espaço tridimensional do quadrimotor, é dado por  ${}^g\mathbf{q} = [\xi \ \eta]^T \in \mathbb{R}^6$ , que pode ser rescrita como:

$${}^g\mathbf{q} = [x \ y \ z \ \phi \ \theta \ \psi]^T \in \mathbb{R}^6 \quad (08)$$

Da mesma forma, as coordenadas generalizadas da velocidade do Drone, é dado por  ${}^b\mathbf{v} = [\Upsilon \ \Omega]^T \in \mathbb{R}^6$ , que também pode ser rescrita como:

$${}^b\mathbf{v} = [u \ v \ w \ p \ q \ r]^T \in \mathbb{R}^6 \quad (09)$$

Após a compreensão do sistema de referência do Drone utilizado, será mostrado a seguir o princípio de funcionamento do mesmo, destacando as possibilidades de manobras possíveis de serem realizadas por este veículo.

### 2.2.2. Princípio de funcionamento

Neste tópico será explanado os princípios básico de funcionamento do AR.Drone Parrot, assim como a descrição dos principais movimentos ao qual essa aeronave pode realizar no espaço tridimensional.

O quadricóptero possui uma estrutura composta por um conjunto de quatro motores e dois pares de hélices simetricamente distribuídos, sendo que dois motores opostos são projetados para girarem no sentido horário e os outros dois no sentido anti-horário, de forma que a rotação em conjunto de todos os motores na mesma velocidade angular elimina o efeito anti-torque<sup>3</sup>. Os motores são acionados de forma independente e suas hélices possuem um ângulo de ataque fixo, sendo as forças de propulsão gerada por cada motor  $(f_1 \ f_2 \ f_3 \ f_4)$  apontadas sempre no sentido positivo do eixo  ${}^b z$  do Drone, conforme foi visto na Figura 10. Dessa forma, para que o Drone levante voo à soma vetorial das quatro forças deve ser igual ou superior a força peso exercida pela gravidade sobre o mesmo. (BRANDÃO *et al*, 2012a, BRANDÃO, 2013).

Como discutido em capítulos anteriores, o quadricóptero escolhido é caracterizado por possuir um sistema subatuado com estrutura em configuração "x", o que torna sua dinâmica altamente acoplada, isso faz com que qualquer movimento no espaço tridimensional necessite da ação em conjunta de todos os motores. Dessa forma, para que se possa compreender seu funcionamento será ilustrado na Figura 11 os movimentos básicos de voo do AR.Drone, destacando quais motores deverão ter sua velocidade angular modificada para gerar força correspondente a cada movimento. A largura das setas e a cor indicam a intensidade de giro de

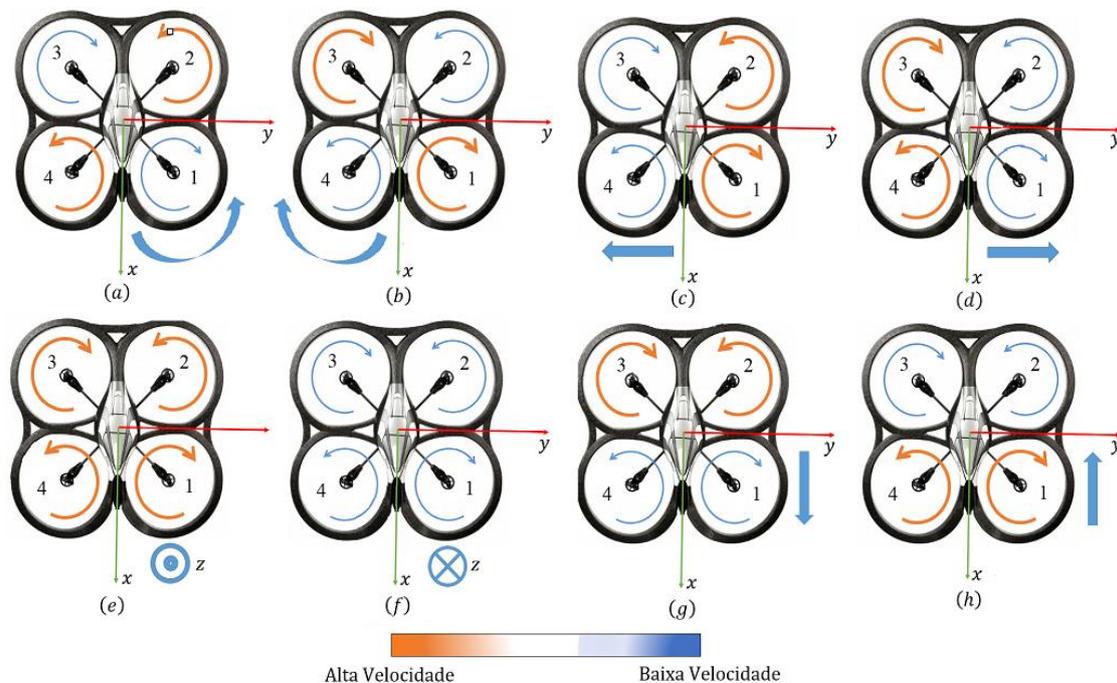
---

<sup>3</sup> Efeito esse que é inerente a rotação das hélices e quando não equilibrado faz o Drone girar em torno do eixo  ${}^b z$ . (BOUABDALLAH, 2004)

cada rotor sendo que as setas largas e de cor laranja indicam alta velocidade dos rotores, as setas finas e de cor azul indica baixa velocidade dos rotores, as setas azuis que estão fora do corpo do Drone indicam o movimento que está sendo realizado. Note que no AR.Drone Parrot os motores opostos 1 e 3 giram no sentido horário enquanto que os motores 2 e 4 giram no sentido anti-horário eliminando o efeito anti-torque mencionado anteriormente.

Para que o Drone realize o movimento de guinada (giro em torno do eixo  $^b z$ ) no sentido anti-horário deve-se fixar a velocidade dos motores 1 e 3 enquanto aumenta a velocidade dos motores 2 e 4, como mostra a Figura 11(a). O contrário ocorre quando se inverte a escolha dos motores, fixando a velocidade dos motores 2 e 4 enquanto aumenta a velocidade dos motores 1 e 3, isso faz o Drone realizar o movimento de guinada só que agora no sentido horário mostrado na Figura 11(b). Para fazer o Drone se movimentar para direita ou esquerda é necessário realizar a manobra de rolagem (giro em torno do eixo  $^b x$ ), dessa forma, deve-se manter a velocidade dos motores 3 e 4 e aumentar a velocidade dos motores 2 e 1 fazendo o Drone realizar a rolagem no sentido horário mostrada na Figura 11(c), de maneira análoga, ao fixar a velocidade dos motores 2 e 1 e aumentar a velocidade dos motores 3 e 4 ele realizará o movimento de rolagem no sentido anti-horário, conforme demonstrada na Figura 11(d).

Figura 11 - Discrição dos movimentos basicos de voo do AR.Drone Parrot 2.0.



Fonte: Adpatado de PARROT, 2014; BRANDÃO, 2013.

Para que o Drone decole ou aterrisse (movimento de subida e descida ao longo do eixo  ${}^b z$ ) é preciso o acionamento de todos os motores simultaneamente, assim se aumentar igualmente a velocidade de todos os motores ele irá subir, da mesma forma ao se reduzir igualmente a velocidade de todos os motores o mesmo irá descer, como mostra a Figura 11(e) e 11(f).

Para fazer o Drone se deslocar para frente (sentido positivo do  ${}^b x$ ) é necessário realizar o movimento de arfagem no sentido anti-horário (giro em torno do eixo  ${}^b y$ ), para isso deve-se aumentar a velocidade dos motores 2 e 3 enquanto mantem a velocidade dos motores 1 e 4, Figura 11 (g). Por fim, para que o Drone se mova para trás (sentido negativo do  ${}^b x$ ) é realizado o movimento de arfagem no sentido horário, aumentando a velocidade dos motores 1 e 4 enquanto mantêm a velocidade dos motores 2 e 3, Figura 11(h).

### 2.2.3. Modelo cinemático

Conforme visto na seção 2.2.1, o sistema referencial de um quadricóptero pode ser representado pelas coordenadas generalizadas, dadas pelas equações (08) e (09). Segundo Brandão (2013), o modelo cinemático do quadrimotor pode ser definido como sendo a relação de velocidades entre os sistemas de referencial inercial  $\langle g \rangle$  e o do corpo do veículo  $\langle b \rangle$ , expresso por:

$${}^b \mathbf{v} = J(\eta)^g \dot{\mathbf{q}} \quad (10)$$

Onde  $\dot{\mathbf{q}}$  é a primeira derivada das coordenadas generalizadas da posição com respeito ao tempo, e  $J(\cdot)$  é a matriz jacobiana que relaciona tais sistemas de referência, essa matriz é dada por:

$$J(\eta) = \begin{bmatrix} \mathcal{R} & 0 \\ 0 & W_\eta \end{bmatrix} \in \mathbb{R}^{6 \times 6} \quad (11)$$

O  $\mathcal{R}$  é a matriz de rotação, que segundo Sá (2012), essa matriz projeta os vetores do sistema de coordenadas  $\langle g \rangle$  no sistema de coordenadas  $\langle b \rangle$  para os ângulos de inclinação, e é dada por:

$$\mathcal{R} = \mathcal{R}(\psi, z)\mathcal{R}(\theta, y)\mathcal{R}(\phi, x) \quad (12)$$

$$\mathcal{R} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (13)$$

$$\mathcal{R} = \begin{bmatrix} c\theta \cdot c\psi & c\psi \cdot s\phi \cdot s\theta - c\phi \cdot s\psi & s\phi \cdot s\psi + c\phi \cdot c\psi \cdot s\theta \\ c\theta \cdot s\psi & c\phi \cdot c\psi + s\phi \cdot s\theta \cdot s\psi & c\phi \cdot s\theta \cdot s\psi - c\psi \cdot s\phi \\ -s\theta & c\theta \cdot s\phi & c\phi \cdot c\theta \end{bmatrix} \quad (14)$$

Segundo Silva (2015), a matriz de Euler ( $W_\eta$ ) relaciona o vetor de velocidades angulares ( $\Omega$ ) com o vetor de velocidades angulares generalizadas ( $\dot{\eta}$ ), em que:

$$W_\eta = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cdot \cos \theta \\ 0 & -\sin \phi & \cos \phi \cdot \cos \theta \end{bmatrix} \quad (15)$$

Com isso, é possível realizar a transformação das velocidades rotacionais no referencial global para o referencial do veículo, através da equação (16):

$$\Omega = W_\eta \dot{\eta} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cdot \cos \theta \\ 0 & -\sin \phi & \cos \phi \cdot \cos \theta \end{bmatrix} \cdot \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (16)$$

Compreendido a cinemática do quadrimotor, agora será demonstrado o modelo dinâmico da aeronave segundo o formalismo de Euler-Lagrange.

#### 2.2.4. Modelo de Euler-Lagrange

Para a representação do modelo dinâmico de um quadrimotor segundo o formalismo de Euler-Lagrange, deve-se considerar a aeronave como sendo um corpo rígido sujeito à ação de forças e torques externos (BRANDÃO, 2013). Logo sua equação é baseada na energia cinética e na energia potencial do sistema (BOUBDALLAH, 2007), podendo ser escrita pela função Lagrangiano  $L$ , que deve satisfazer a restrição de Euler-Lagrange, dada por:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial L}{\partial \mathbf{q}} = \begin{bmatrix} {}^g \mathbf{f} \\ {}^g \boldsymbol{\tau} \end{bmatrix} \quad (17)$$

Onde:

$L$  = Função Lagrangiano é definida como sendo a diferença entre a energia cinética total e a energia potencial do sistema ( $L = K_{Total} - P$ );

$K_{Total}$  = Energia cinética total do sistema (dada pela energia cinética translacional mais a energia cinética rotacional);

$P$  = Energia potencial do sistema;

$\mathbf{q}$  = Coordenadas generalizadas (veja equação 2.8);

$\dot{\mathbf{q}}$  = Primeira derivada das coordenadas generalizadas com respeito ao tempo;

${}^g\mathbf{f}$  = Representa o vetor de forças aplicadas ao sistema;

${}^g\boldsymbol{\tau}$  = Representa o vetor de torques aplicadas ao sistema;

Segundo Brandão (2013), não há uma relação explícita entre  $\dot{\xi}$  e  $\dot{\eta}$ , de forma que a equação de Euler-Lagrange pode ser dividida em duas, na dinâmica translacional que resulta no vetor de forças  $\mathbf{f}$  (que representa as entradas de controle translacionais indiretas) e na dinâmica rotacional que resulta no vetor de torques  $\boldsymbol{\tau}$  (que representa as entradas de controle rotacional indiretas).

#### ➤ Dinâmica translacional

Sabendo-se que o Lagrangiano  $L$  é dado por:

$$L = K_{Total} - P \quad (18)$$

Tem-se que a energia cinética total, é dada pelo somatório da energia cinética translacional e a energia cinética rotacional do veículo:

$$K_{Total} = K_{Trans} + K_{Rot} \quad (19)$$

Primeiramente será deduzido as contribuições translacionais e posteriormente as rotacionais, dessa forma a energia cinética e a energia potencial de translação são dadas pelas equações (20) e (21), respectivamente:

$$K_{Trans} = \frac{1}{2} m \dot{\xi}^T \dot{\xi} = \frac{1}{2} m (\dot{x}^2 + \dot{y}^2 + \dot{z}^2) \quad (20)$$

$$P = mgz \quad (21)$$

Onde  $m$  é a massa total do quadrimotor,  $g$  é a aceleração gravitacional e  $z$  a altura do veículo. Dessa forma, pode-se reescrever a equação (18) mostrando apenas o Lagrangiano translacional, conforme descrito abaixo:

$$L_{Trans} = K_{Trans} - P \quad (22)$$

$$L_{Trans} = \frac{1}{2} m(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) - mgz \quad (23)$$

Aplicando a equação (23) na restrição de Euler-Lagrange, dada pela equação (17), e resolvendo as derivadas parciais, é possível obter a dinâmica translacional, que é dada por:

$$\frac{d}{dt} \left( \frac{\partial L_{Trans}}{\partial \dot{\xi}} \right) - \frac{\partial L_{Trans}}{\partial \xi} = m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} + g \end{bmatrix} = \mathbf{f} \quad (24)$$

O vetor de forças  $\mathbf{f}$ , representa as entradas de controle translacionais indiretas, conforme mostrado a seguir:

$$\mathbf{f} = [f_x \ f_y \ f_z]^T = \mathcal{R} \mathcal{A}_t [f_1 \ f_2 \ f_3 \ f_4]^T \quad (25)$$

Onde o  $\mathcal{R}$  é a matriz de rotação dada pela equação (14) e o  $\mathcal{A}_t$  é a matriz que relaciona o arrasto gerado pelos motores do quadrimotor e a força total que está atuando sobre ele, neste caso, representadas por  $f_1 \ f_2 \ f_3 \ f_4$ , que são as forças de propulsão (ou força de empuxo) gerada por cada motor, elas estão sempre apontadas para cima, na direção do eixo  ${}^b z$ , conforme visto na Figura 10, logo essa matriz  $\mathcal{A}_t$  é escrita como:

$$\mathcal{A}_t = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (26)$$

As forças de propulsão  $f_i$  gerada por cada motor ( $i = 1, 2, 3, 4$ ) é dada por:

$$f_i = b_i \omega_i^2 \quad (27)$$

Onde  $b_i$  é coeficiente de empuxo do motor, e  $\omega_i^2$  é a velocidade angular desenvolvida pelo motor.

Dessa forma, ao substituir a equação (24) na equação (25), resulta em:

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} + g \end{bmatrix} = \mathcal{RA}_t [f_1 \ f_2 \ f_3 \ f_4]^T \quad (28)$$

Continuando o desenvolvimento da equação (28) agora adicionando a matriz de rotação descrita pela equação (14) e a matriz de arrasto descrita pela equação (26), tem-se que:

$$\begin{aligned} m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} + g \end{bmatrix} &= \begin{bmatrix} c\theta \cdot c\psi & c\psi \cdot s\phi \cdot s\theta - c\phi \cdot s\psi & s\phi \cdot s\psi + c\phi \cdot c\psi \cdot s\theta \\ c\theta \cdot s\psi & c\phi \cdot c\psi + s\phi \cdot s\theta \cdot s\psi & c\phi \cdot s\theta \cdot s\psi - c\psi \cdot s\phi \\ -s\theta & c\theta \cdot s\phi & c\phi \cdot c\theta \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \\ &= \begin{bmatrix} c\theta \cdot c\psi & c\psi \cdot s\phi \cdot s\theta - c\phi \cdot s\psi & s\phi \cdot s\psi + c\phi \cdot c\psi \cdot s\theta \\ c\theta \cdot s\psi & c\phi \cdot c\psi + s\phi \cdot s\theta \cdot s\psi & c\phi \cdot s\theta \cdot s\psi - c\psi \cdot s\phi \\ -s\theta & c\theta \cdot s\phi & c\phi \cdot c\theta \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ f_1 + f_2 + f_3 + f_4 \end{bmatrix} \\ &= \begin{bmatrix} c\theta \cdot c\psi & c\psi \cdot s\phi \cdot s\theta - c\phi \cdot s\psi & s\phi \cdot s\psi + c\phi \cdot c\psi \cdot s\theta \\ c\theta \cdot s\psi & c\phi \cdot c\psi + s\phi \cdot s\theta \cdot s\psi & c\phi \cdot s\theta \cdot s\psi - c\psi \cdot s\phi \\ -s\theta & c\theta \cdot s\phi & c\phi \cdot c\theta \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} \quad (29) \end{aligned}$$

Onde  $U_1$  é a entrada indireta do controle de altitude, representando a força de empuxo aplicado a estrutura do quadrimotor, esse termo é usado para simplificar os cálculos, onde:

$$U_1 = f_1 + f_2 + f_3 + f_4 \quad (30)$$

Substituindo na equação (27) :

$$U_1 = b_1\omega_1^2 + b_2\omega_2^2 + b_3\omega_3^2 + b_4\omega_4^2 \quad (31)$$

Simplificando:

$$U_1 = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \quad (32)$$

Logo, voltando à resolução da equação (29), é possível obter a simplificação das equações da dinâmica translacional dada pelas equações (33), (34) e (35) (SÁ, 2012; BENIGNO, 2015; SILVA, 2015; ABBASI, *et al.*, 2013).

$$\ddot{x} = (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \frac{1}{m} U_1 \quad (33)$$

$$\ddot{y} = (\cos \phi \sin \theta \cos \psi - \sin \phi \sin \psi) \frac{1}{m} U_1 \quad (34)$$

$$\ddot{x} = -g + (\cos \phi \cos \theta) \frac{1}{m} U_1 \quad (35)$$

➤ Dinâmica rotacional

Voltando para a equação (19), agora aplicando as contribuições referentes à dinâmica rotacional, tem-se que, a energia cinética rotacional do veículo é dada por:

$$K_{Rot} = \frac{1}{2} \Omega^T \mathbf{I} \Omega \quad (36)$$

Dessa forma, pode se reescrever a equação (18) mostrando apenas o Lagrangiano rotacional, que fica dessa forma:

$$L_{Rot} = K_{Rot} \quad (37)$$

$$L_{Rot} = \frac{1}{2} \Omega^T \mathbf{I} \Omega = \frac{1}{2} \dot{\eta}^T W_{\eta}^T \mathbf{I} W_{\eta} \dot{\eta} \quad (38)$$

Onde  $\mathbf{I}$  é a matriz de inércia dada pela equação (39), quando considerado que a estrutura do quadrimotor é simétrica (BRANDÃO, 2013).

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (39)$$

Aplicando a equação (38) na restrição de Euler-Lagrange dada pela equação (17), e resolvendo as derivadas parciais, é possível obter a dinâmica rotacional, que segundo Brandão (2013) é dada por:

$$\begin{aligned} \frac{d}{dt} \left( \frac{\partial L_{Rot}}{\partial \dot{\eta}} \right) - \frac{\partial L_{Rot}}{\partial \eta} &= \frac{d}{dt} \left[ \frac{\partial}{\partial \dot{\eta}} \left( \frac{1}{2} \dot{\eta}^T W_{\eta}^T \mathbf{I} W_{\eta} \dot{\eta} \right) \right] - \frac{\partial}{\partial \eta} \left( \frac{1}{2} \dot{\eta}^T W_{\eta}^T \mathbf{I} W_{\eta} \dot{\eta} \right) \\ &= \frac{d}{dt} \left[ \frac{\partial}{\partial \dot{\eta}} \left( \frac{1}{2} \dot{\eta}^T \mathbf{M}_r \dot{\eta} \right) \right] - \frac{\partial}{\partial \eta} \left( \frac{1}{2} \dot{\eta}^T \mathbf{M}_r \dot{\eta} \right) \end{aligned}$$

$$= \mathbf{M}_r \ddot{\eta} + \dot{\mathbf{M}}_r \dot{\eta} - \frac{1}{2} \dot{\eta}^T \frac{\partial \mathbf{M}_r}{\partial \eta} \dot{\eta} = \boldsymbol{\tau} \quad (40)$$

Onde  $\underline{\dot{\eta}} = \begin{bmatrix} \dot{\eta} & 0 & 0 \\ 0 & \dot{\eta} & 0 \\ 0 & 0 & \dot{\eta} \end{bmatrix} \in \mathbb{R}^{9 \times 3}$  e  $\mathbf{M}_r$  é a matriz de inércia rotacional dada por:

$$\mathbf{M}_r = W_\eta^T \mathbf{I} W_\eta \quad (41)$$

Em que, substituindo as equações (15) e (39) na equação (41), tem-se:

$$\mathbf{M}_r = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cdot \cos \theta \\ 0 & -\sin \phi & \cos \phi \cdot \cos \theta \end{bmatrix}^T \cdot \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cdot \cos \theta \\ 0 & -\sin \phi & \cos \phi \cdot \cos \theta \end{bmatrix}$$

$$\mathbf{M}_r = \begin{bmatrix} I_{xx} & 0 & -I_{xx}s\theta \\ 0 & I_{yy}c^2\phi + I_{zz}s^2\phi & (I_{yy} - I_{zz})c\phi s\phi c\theta \\ -I_{xx}s\theta & (I_{yy} - I_{zz})c\phi s\phi c\theta & I_{xx}s^2\theta + I_{yy}s^2\phi c^2\theta + I_{zz}c^2\phi c^2\theta \end{bmatrix} \quad (42)$$

O vetor de torques  $\boldsymbol{\tau}$ , representa as entradas de controle rotacional indiretas, e é dado por:

$$\boldsymbol{\tau} = [\tau_\phi \ \tau_\theta \ \tau_\psi]^T = \mathcal{A}_r [f_1 \ f_2 \ f_3 \ f_4]^T \quad (43)$$

Onde  $\mathcal{A}_r$  é a matriz que corresponde aos braços das forças responsáveis por gerar o torque no quadrimotor, dada por:

$$\mathcal{A}_t = \begin{bmatrix} l & l & -l & -l \\ -l & l & l & -l \\ d & -d & d & -d \end{bmatrix} \quad (44)$$

O  $l$  representa a distância entre o centro do quadrimotor e o centro da hélice,  $d$  representa o coeficiente de arrasto. Dessa forma adicionando a matriz dada pela equação (44) na equação (45) do vetor do torque mostrado na equação (26), tem-se que:

$$\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} l & l & -l & -l \\ -l & l & l & -l \\ d & -d & d & -d \end{bmatrix} \cdot \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}$$

$$\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} l.f_1 + l.f_2 - l.f_3 - l.f_4 \\ -l.f_1 + l.f_2 + l.f_3 - l.f_4 \\ d.f_1 - d.f_2 + d.f_3 - d.f_4 \end{bmatrix}$$

Substituindo a equação (27) da força de propulsão dos motores e isolando os termos:

$$\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} lb(\omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2) \\ lb(-\omega_1^2 + \omega_2^2 + \omega_3^2 - \omega_4^2) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} = \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} \quad (45)$$

Onde  $U_1$ , dada pela equação (32), mais  $U_2, U_3$  e  $U_4$ , formam as entradas indiretas do controle de altitude, ângulo de rolagem, arfagem e guinada, respectivamente (SÁ, 2012):

$$U_2 = lb(\omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2) \quad (46)$$

$$U_3 = lb(-\omega_1^2 + \omega_2^2 + \omega_3^2 - \omega_4^2) \quad (47)$$

$$U_4 = d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \quad (48)$$

Em virtude da complexidade da equação (40) que descreve a dinâmica rotacional, a mesma não será feita a simplificações como ocorrido na dinâmica translacional, sabendo-se que este não é o foco do trabalho e tais equações já estão contidas dentro da plataforma AuRoRA, utilizada para validação do controlador proposto nessa dissertação. Dessa forma, voltando para a função Lagrangiano da equação (18), e substituindo as contribuições translacionais e rotacionais dadas pelas equações (20) e (36), respectivamente, a mesma pode ser reescrita como:

$$L = \frac{1}{2} m \dot{\xi}^T \dot{\xi} + \frac{1}{2} \Omega^T \mathbf{I} \Omega - mgz \quad (49)$$

Por fim em Brandão (2013) é apresentado o modelo de alto nível da dinâmica não linear de um quadrimotor segundo o formalismo de Euler-Lagrange:

$$\begin{bmatrix} mI_{3 \times 3} & 0 \\ 0 & \mathbf{M}_r(\eta) \end{bmatrix} \begin{bmatrix} \ddot{\xi} \\ \ddot{\eta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{C}_r(\eta, \dot{\eta}) \end{bmatrix} \begin{bmatrix} \dot{\xi} \\ \dot{\eta} \end{bmatrix} + \begin{bmatrix} \mathbf{G}(g) \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \boldsymbol{\tau} \end{bmatrix} - \begin{bmatrix} \mathbf{D}_t \\ \mathbf{D}_r \end{bmatrix} \quad (50)$$

Onde  $\mathbf{C}_r(\eta, \dot{\eta})$  é a matriz de Coriolis e forças centrípetas, que define os efeitos giroscópios e o sistema centrífugo da aeronave, dada por:

$$\mathbf{C}_r(\eta, \dot{\eta}) = \mathbf{M}_r \dot{\eta} - \frac{1}{2} \dot{\eta}^T \frac{\partial \mathbf{M}_r}{\partial \eta} \in \mathbb{R}^{3 \times 3} \quad (51)$$

O  $\mathbf{G}$  é o vetor de forças gravitacionais dado por:

$$\mathbf{G} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (52)$$

E  $\mathbf{D}_t$  e  $\mathbf{D}_r$  representam os vetores de distúrbios translacionais e rotacionais, respectivamente. Através dessas equações é possível descrever matematicamente o modelo dinâmico de um VANT do tipo quadrimotor quando o mesmo atua no espaço cartesiano tridimensional.

### 2.3. TÉCNICAS DE CONTROLE APLICADAS A VANTs

O controle dos veículos aéreos não tripulados pode ser realizado de duas formas, manualmente, através de um rádio controle, ou de forma autônoma por intermédio de controladores embarcados, sendo que o segundo caso é o principal foco do seu estudo. Para conseguir uma navegação autônoma desses veículos, é necessário a construção de um controlador que garanta tanto a estabilidade quanto um bom desempenho do mesmo, dessa forma uma série de estratégias de controle se encontram dispostas na literatura, tais como: Controlador de três termos ou Controlador Proporcional-Integrativo-Derivativo (PID), Redes Neurais e Artificiais, Algoritmos Genéticos, Controle Backstepping, Lógica *Fuzzy*, dentre outros mais.

Dentro desse contexto, está dissertação propõe o desenvolvimento de um controlador híbrido *Fuzzy*-PID ao qual incorpora tanto as características do controle clássico PID quanto às vantagens do sistema de lógica *Fuzzy* para ajuste de seus parâmetros. Este tópico será dedicado a uma breve explanação sobre o funcionamento desses três controladores, o controlador clássico (PID), o controlador *Fuzzy* supervisor e a junção das duas técnicas que forma o controlador híbrido *Fuzzy*-PID.

#### 2.3.1. Controlador de três termos (PID)

Um dos algoritmos de controle *feedback* mais utilizado no setor industrial é o chamado controlador de três termos ou controlador PID (*Proporcional-Integral-Derivativo*). A popularidade desse controlador pode ser atribuída parcialmente as suas características básicas

de projeto como, simplicidade, baixo custo, facilidade de implementação e robustez, porém, o que de fato torna esse controle a escolha preferida dos programadores é sua capacidade em aplicações, ao qual, apresentando bom desempenho para uma ampla faixa de sistemas. (DORF, 2011; OGATA, 2011).

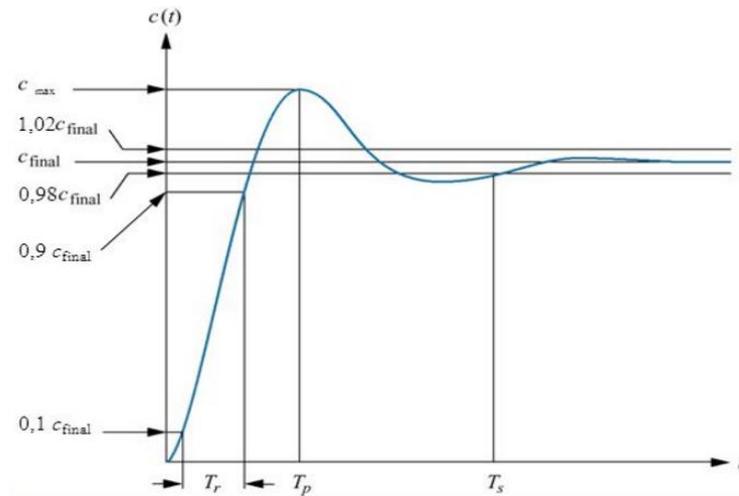
Esse controlador PID, como o próprio nome sugere, é uma técnica que combina as três ações de controle, proporcional, integral e derivativo. A combinação dessas ações fornece as vantagens individuais de cada uma das três técnicas, juntando todas em um único controlador. (OGATA, 2011).

Contudo, apesar de possuir essas três ações de controle, às vezes o sistema ao qual vai ser controlado não se faz necessário da utilização de todos os parâmetros, ocasionando dessa forma a fragmentação do controlador PID. Assim é possível encontrar os seguintes tipos de controladores:

- ✓ Controlador Proporcional (P);
- ✓ Controlador Proporcional e Integral (PI);
- ✓ Controlador Proporcional e Derivativo (PD);
- ✓ Controle Proporcional, Integral e Derivativo (PID);

Antes de fazer uma explanação sobre cada ação de controle, é interessante falar sobre alguns parâmetros de ajustes importantes do sinal que geralmente são controlados pelo programador. A Figura 12 mostra a curva de saída de um sistema qualquer cujos parâmetros estão associados a uma resposta subamortecida (que é caracterizada pela sequência de sobre saltos em torno do valor desejado, onde os níveis de oscilações diminuem com passar do tempo), esse tipo de resposta é característica de sistemas amortecidos de segunda ordem, eles apresentam alguns parâmetros importantes que são analisados conforme o estudo da planta em questão, parâmetros como: tempo de subida, tempo de pico, o sobressinal (*overshoot*) e o tempo de assentamento, são obtidos pela análise do sinal e ajudam no controle do sistema.

Figura 12 -Parâmetros de ajustes importantes de um sinal controlado.



Fonte: NISE, 2009, p.141.

Onde:

- Tempo de subida ou *Rise Time* ( $T_r$ ): É o tempo necessário para a resposta ao degrau subir de 0.1 (10%) até 0.9 (90%) de seu valor final (NISE, 2009, p. 140).
- Tempo de pico ( $T_p$ ): É o tempo necessário para se atingir o primeiro pico (valor máximo) (NISE, 2009, p. 140).
- Sobressinal ou Sobrevalor Percentual ou Overshoot (%SP): Representa o quanto a forma de onda ultrapassa o valor em regime permanente, ou final, no tempo de pico, esse valor é expresso como uma porcentagem do valor em regime permanente (NISE, 2009, p. 140).
- Tempo de Assentamento ou Tempo de Acomodação ou *Settling Time* ( $T_s$ ): É o tempo necessário para que as oscilações amortecidas transientes da resposta ao degrau, atinjam e permaneçam em torno do valor de regime permanente, dentro de uma faixa de  $\pm 2\%$  desse valor (NISE, 2009, p. 140).

➤ Controlador Proporcional (P):

Nesse tipo de controlador a ação de controle, ou seja, a saída do controlador  $u(t)$  é proporcional ao sinal de erro atuante  $e(t)$ , a relação entre eles representada no domínio do tempo é dada por:

$$u_p(t) = K_p e(t) \quad (53)$$

A transformada de Laplace ou representação no domínio da frequência é dada por:

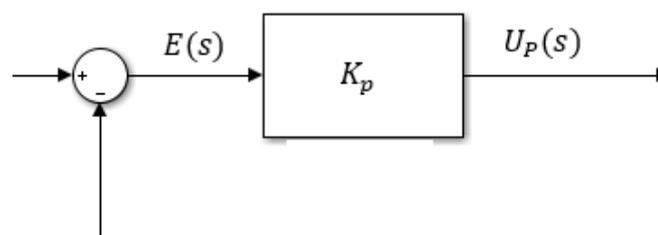
$$U_p(s) = K_p E(s) \quad (54)$$

A função de transferência é:

$$\frac{U_p(s)}{E(s)} = K_p \quad (55)$$

Onde  $K_p$  é denominado ganho proporcional, gerado pela razão entre a saída e a entrada do controlador. A letra maiúscula e o termo “(s)” indicam que a variável está no domínio da frequência, da mesma forma, a letra minúscula e o termo “(t)” indicam que a variável está no domínio do tempo. O  $E(s)$  e  $e(t)$  são os erros atuantes no domínio da frequência e no tempo, respectivamente, esses erros são gerados pela diferença entre o sinal de entrada e o sinal de saída do sistema, quando o mesmo se encontra em regime permanente. O  $U_p(s)$  ou  $u_p(t)$  são os termos proporcionais no domínio da frequência e no tempo, respectivamente, ambos representam o sinal de saída do controlador. A Figura 13 mostra um diagrama de bloco de um controlador proporcional (P).

Figura 13 - Diagrama de bloco de um controlador proporcional (P).

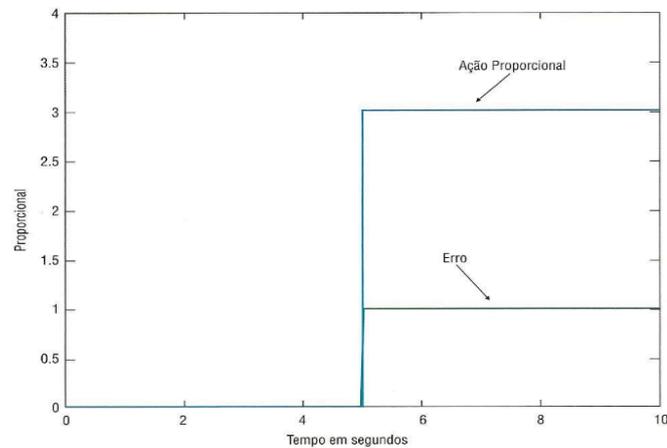


**Fonte:** Adaptado de OGATA, 2011, p. 201.

Segundo Ogata (2011) e Campos (2010) para qualquer que seja o mecanismo real e o tipo de energia que está sendo utilizada na operação de controle, o controlador proporcional é na essência um simples amplificador com um ganho constante, ou seja, se em algum instante de tempo o sinal do erro for grande, devido a proporcionalidade, nesse mesmo instante de tempo a saída do controlador também será grande. Tal afirmação pode ser vista Figura 14, onde uma ação proporcional é aplicada a um erro em degrau, nota-se que saída  $u(t)$  apresenta mesma distância que o sinal de erro  $e(t)$  (um degrau com amplitude 1, resultante da diferença entre a saída  $u(t) = 3$  e o *set-point* que é 4), e sua amplitude é proporcional ao erro multiplicado pelo

ganho  $K_p$  (nesse caso é 3). Dessa forma se não houver variação do erro também não haverá variação da saída  $u(t)$ , permitindo o erro em regime constante.

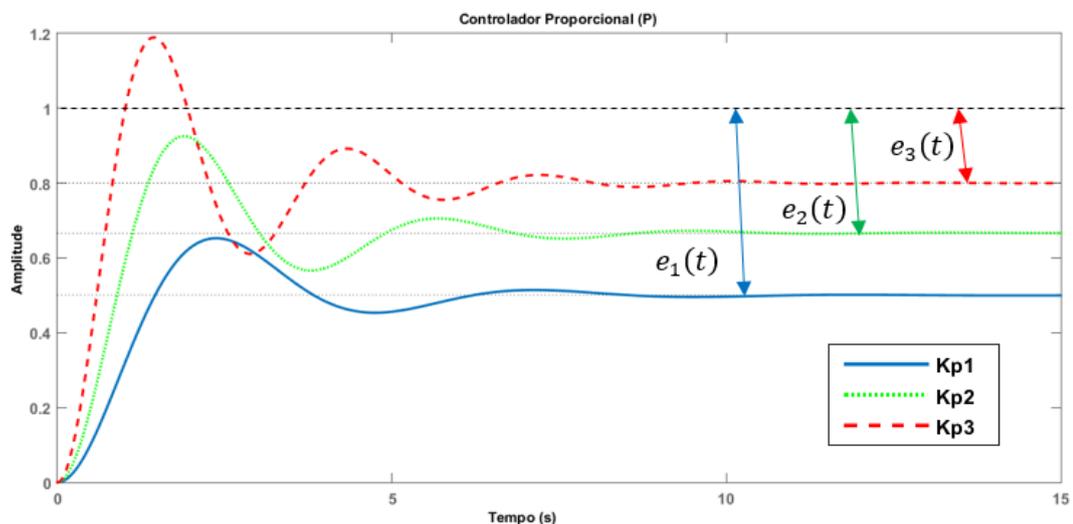
Figura 14 - Ação proporcional mediante ao erro



Fonte: CAMPOS (2010).

A Figura 15 mostra a um exemplo da resposta de saída de um sistema de ordem zero para uma entrada em degrau (imagem gerada através do *software* MATLAB), enquanto se varia apenas o valor do ganho proporcional  $K_p$ . É possível observar que a medida em que se aumenta o valor do ganho ( $K_{p3} > K_{p2} > K_{p1}$ ) o erro diminui ( $e_3(t) < e_2(t) < e_1(t)$ ), porém o transitório também aumenta.

Figura 15 – Exemplo de saídas de um sistema de ordem zero para uma entrada em degrau onde se aplica um controle proporcional puro com ajustes de ganho  $K_{p3} > K_{p2} > K_{p1}$ .



Fonte: Autoria própria.

A utilização apenas de um ganho proporcional torna o controle do sistema bastante limitado. Essa limitação é atribuída, em geral, a incapacidade desse controlador em eliminar completamente o erro em regime permanente, chamado também de erro residual, pois na medida em que o erro tende a zero o sinal de controle também tende a zero, de forma que irá chegar a um ponto em que o aumento do ganho não conseguirá mais atenuar o erro, deixando-o constantes no regime estacionário, porém, se persistir em aumentar o ganho tendendo-o ao infinito, o mesmo irá aumentar a oscilação no regime transitório podendo, em certos casos, tornar o sistema instável (OGATA, 2011).

➤ Controlador proporcional e integral (PI):

Esse tipo de controlador possui a ação conjunta de dois tipos de controle, o proporcional (explicado no tópico anterior) e o integral. Segundo Ogata (2011), na ação de controle integral, o valor de saída do controlador  $u_i(t)$  é modificado a uma taxa de variação proporcional ao sinal atuante  $e(t)$ , a relação entre eles representada no domínio do tempo é dada por:

$$\frac{du_i(t)}{dt} = K_i e(t) \quad (56)$$

Ou:

$$u_i(t) = K_i \int_0^t e(t) dt \quad (57)$$

Ou também:

$$u_i(t) = \frac{1}{T_i} \int_0^t e(t) dt \quad (58)$$

Onde  $T_i$  é chamado de tempo *integrativo* ou tempo de *reset*, geralmente expresso em minutos. A função de transferência do controlador integral é:

$$\frac{U_i(s)}{E(s)} = \frac{K_i}{s} \quad (59)$$

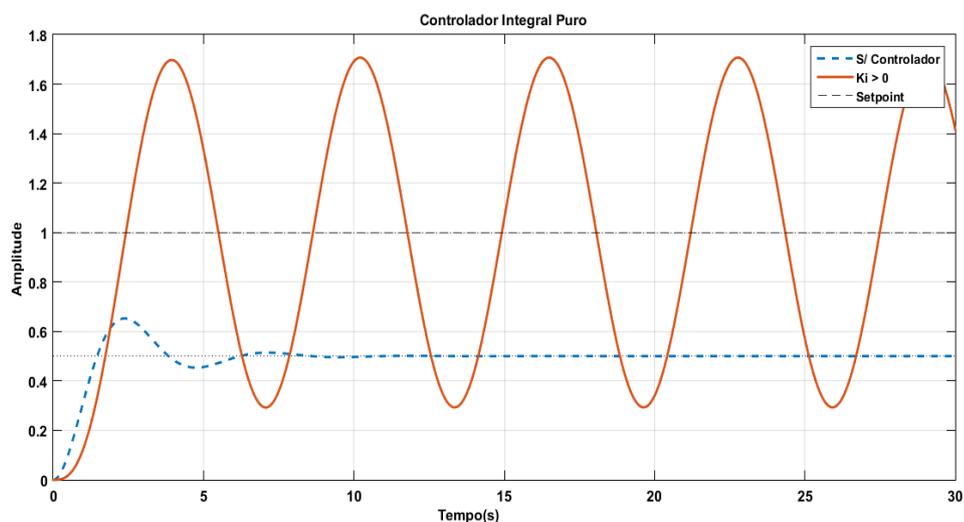
Onde  $K_i$  constante ajustável denominada de ganho integral,  $E(s)$  e  $e(t)$  são os erros atuantes no domínio da frequência e no tempo, respectivamente. O  $U_i(s)$  ou  $u_i(t)$  são os termos integrais no domínio da frequência e no tempo, respectivamente, ambos representando o sinal de saída do controlador.

Como visto no t3pico anterior, quando o sinal do erro em um sistema fechado vai a zero, o sinal de sa3da do controlador proporcional tamb3m vai 0 zero, por3m, j0 na a3o de controle integral o sinal de sa3da  $u_i(t)$  permanece constante. A utiliza3o de um integrador como controlador, torna a resposta do sistema mais lenta, pois o mesmo fica dependente do ac3mulo do sinal de erro na entrada do sistema (SUZUKY, 2009). Com a utiliza3o da a3o de controle integral o sistema passa a ser acrescido de um p3lo na origem, o que muda a ordem do sistema, logo, sabendo-se que o erro em regime estacion0rio para uma entrada ao degrau s3 existe em sistemas cuja a fun3o seja de ordem zero, a altera3o dessa ordem pelo incremento da a3o de controle integral faz com que o erro em regime passe a n0 mais existir, eliminando o erro residual, que antes n0 era poss3vel com a utiliza3o apenas da a3o de controle proporcional puro, um vez que o mesmo n0 acrescenta nem p3los nem zeros ao sistema (OGATA, 2011).

Dessa forma, raramente a a3o de controle integral 3 utilizada sozinha, pois, para que a vari0vel controlada atinja um valor significativo, 3 preciso que o erro residual permane3a por um certo tempo, no entanto, como a a3o integral elimina o erro em regime, ele acaba por piorar a estabilidade do sistema, como pode ser visto na Figura 16.

A Figura 16 mostra a resposta de um sistema de ordem zero para uma entrada em degrau onde se aplica um controle integral puro. A curva em azul (pontilhado) representa a resposta do sistema sem controlador e a curva em vermelho (cont3nua) mostra a resposta desse sistema com um ganho integral puro, onde o mesmo torna-se inst0vel.

Figura 16 - Resposta de um sistema de ordem zero para uma entrada ao degrau onde se aplica um controle integral puro.



**Fonte:** Autoria Pr3pria.

Contudo, ao combinar uma ação de controle proporcional com a ação de controle integral, torna-se possível ajustar o valor da variável de controle através da variação do ganho  $K_p$  e ainda assim eliminar o erro em regime através do ajuste do ganho  $K_i$ , formando o controlador proporcional-integrativo (PI). (CAMPOS, 2010).

A definição no domínio do tempo da ação de controle de um controlador proporcional-integral é dada pela equação (60)

$$u_{PI}(t) = K_p \left( e(t) + \frac{1}{T_i} \int_0^t e(t) dt \right) \quad (60)$$

A função de transferência do controlador PI é:

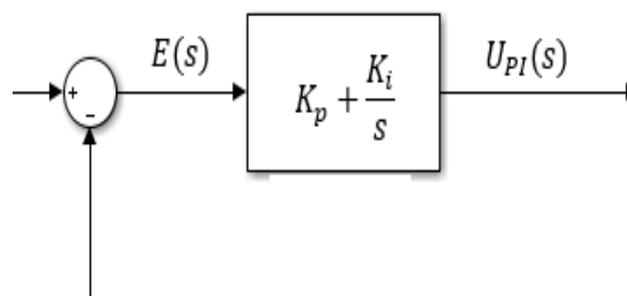
$$\frac{U_{PI}(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i s} \right) \quad (61)$$

Ou:

$$\frac{U_{PI}(s)}{E(s)} = K_p + \frac{K_i}{s} \quad (62)$$

A Figura 17 mostra o diagrama de bloco de um controlador proporcional-integrativo (PI).

Figura 17 - Diagrama de bloco de um controlador PI.

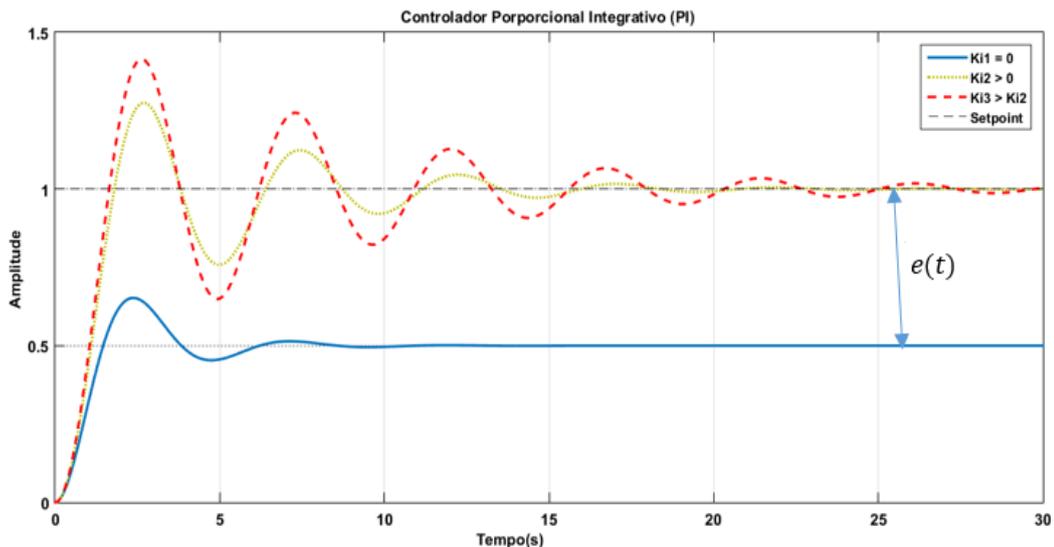


**Fonte:** Adaptado de OGATA, 2011, p. 198.

A Figura 18 mostra a resposta da saída de um sistema de ordem zero para uma entrada ao degrau, aplicando o controlador PI e variando apenas o valor do ganho integral  $K_i$ . A curva azul (contínua) representa a resposta do sistema com apenas um ganho proporcional onde o ganho integral é zero ( $K_{i1} = 0$ ), dessa forma pela Figura 18 é possível notar que existe um erro em regime permanente. A curva em amarelo (pontilhado) representa a aplicação de um ganho

integrativo  $K_{i1} > 0$ , a adição de um ganho integrativo elimina o erro em regime, porém aumenta a oscilação no regime transitório e o tempo de acomodação no sinal. À medida que continua a aumentar o valor do ganho integral, aumenta-se também a amplitude das oscilações, o número de envoltórias e o tempo de acomodação do sinal, demonstrado pela curva em amarelo (tracejado) onde ( $K_{i3} > K_{i2}$ ).

Figura 18 - Resposta de um sistema de ordem zero para uma entrada ao degrau onde se aplica um controle proporcional-integrativo (PI).



**Fonte:** Autoria própria.

A desvantagem do PI, como foi visto, é que ele interfere no transitório do sinal, pois o mesmo acrescenta um pólo ao sistema, como consequência tende a desestabilizá-lo, além de aumentar o tempo de acomodação do sinal (NISE, 2009, p 356).

#### ➤ Controle Proporcional-Derivativo (PD)

O controlador PD, assim como controlador PI, possui duas ações de controle, no entanto uma é a ação proporcional e uma derivativa. Na ação de controle derivativa, o valor de saída do controlador  $u_d(t)$  responde a uma taxa de variação do sinal de erro atuante  $e(t)$ . Essa ação de controle pode realizar uma correção significativa do erro antes que ele se torne muito elevado, ou seja, ele permite prever o erro atuante e iniciar uma correção antecipada (OGATA, 2011, p. 201). A representação da ação derivativa no domínio do tempo é dada por:

$$u_d(t) = K_d \frac{e(t)}{dt} \quad (63)$$

Ou:

$$u_i(t) = T_d \frac{e(t)}{dt} \quad (64)$$

Onde  $T_d$  é chamado de tempo derivativo, geralmente expresso em minutos. Ele representa o avanço introduzido pela ação derivativa, ao qual permite o controlador ter a capacidade de antecipar à ocorrência de um erro futuro.

A função de transferência é:

$$\frac{U_d(s)}{E(s)} = K_d s \quad (65)$$

Ou também pode ser escrita como:

$$\frac{U_d(s)}{E(s)} = T_d s \quad (66)$$

Onde  $K_d$  é a constante ajustável denominada de ganho derivativo,  $E(s)$  e  $e(t)$  são os erros atuantes no domínio da frequência e no tempo, respectivamente. O  $U_d(s)$  ou  $u_d(t)$  são os termos derivativos no domínio da frequência e no tempo, respectivamente, ambos representam o sinal de saída do controlador.

A ação derivativa tem o efeito direto no regime transitório do sistema, ela pode reduzir a velocidade de variação da variável controlada, evitando que a mesma se eleve ou reduza muito rápido durante as perturbações ou na partida do processo (CAMPOS, 2010, p. 30). Porém, a ação de controle derivativa nunca é utilizada sozinha, pois como a mesma atua sobre a taxa de variação do erro atuante e não sobre o próprio erro atuante, se esse erro for constante a saída do controlador derivativo puro será nulo, logo essa ação de controle é sempre usada em conjunto com um controlador proporcional, resultando no controlado proporcional-derivativo (PD), ou junto com o controlador PI, resultando no controlador PID (OGATA, 2011, p. 201).

A definição no domínio do tempo da ação de controle de um controlador proporcional-derivativo é dada pela equação (67)

$$u_{PD}(t) = K_p \left( e(t) + T_d \frac{de(t)}{dt} \right) \quad (67)$$

A função de transferência do controlador PD é:

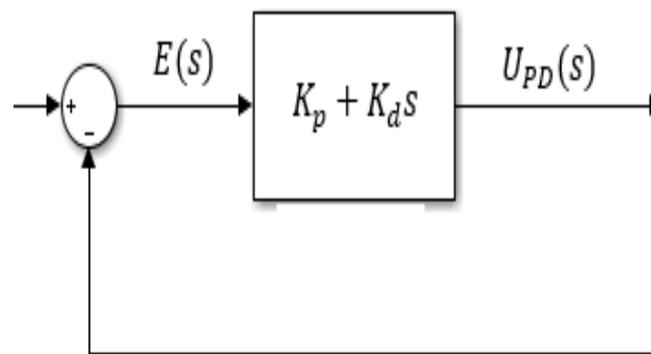
$$\frac{U_{PD}(s)}{E(s)} = K_p(1 + T_d s) \quad (68)$$

Também expressa como:

$$\frac{U_{PD}(s)}{E(s)} = K_p + K_d s \quad (69)$$

A Figura 19 mostra um diagrama de bloco de um controlador proporcional-derivativo (PD).

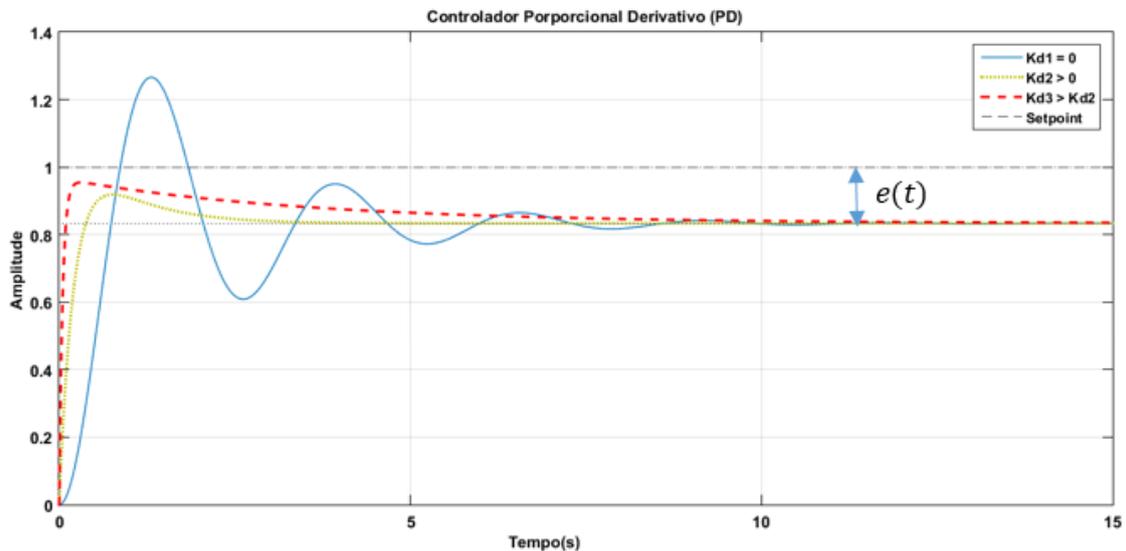
Figura 19 - Diagrama de bloco de um controlador PD.



**Fonte:** Adaptado de OGATA, 2011, p. 202.

A Figura 20 mostra a resposta da saída de um sistema de ordem zero para uma entrada em degrau, aplicando o controlador PD variando apenas o valor do ganho derivativo  $K_d$ . A curva azul (contínua) representa a resposta do sistema com apenas um ganho proporcional, sem o ganho derivativo ( $K_{d1} = 0$ ). A curva em amarelo (pontilhado) representa a aplicação de um ganho derivativo  $K_{d1} > 0$ , nota-se que a parte transitória do sistema fica mais suave, o tempo de acomodação diminui e o tempo de subida diminui, no entanto, se continuar a aumentar o ganho derivativo ( $K_{d3} > K_{d2}$ ), o tempo de subida vai diminuir, porém, também irá aumentar o *overshoot* (sobressinal) e o tempo de acomodação do sinal, como mostrada na curva em vermelho (tracejado).

Figura 20 - Resposta de um sistema de ordem zero para uma entrada ao degrau onde se aplica um controle proporcional-derivativo (PD).



Fonte: Autoria Própria.

A implementação do controlador PD na planta resulta em um acréscimo de um zero ao sistema, dessa forma, o efeito desse controlador atua de forma benéfica no regime transitório do sinal, ele tende a reduzir a velocidade de variação do sobressinal, ou seja, ele tende a aumentar a estabilidade relativa do sistema reduzindo o tempo de acomodação e tempo de subida, no entanto, o aumento excessivo do ganho derivativo pode causar o aumento do tempo de resposta do sinal, além do que, esse controlador não corrige erro em regime.

#### ➤ Controlador Proporcional Integrativo Derivativo (PID)

O PID nada mais é do que a junção das três ações de controle, proporcional, integrativo e derivativo em um só controlador. Dessa forma, o PID é capaz de atuar tanto no regime transitório quanto no regime permanente do sinal de saída, controlando todos os parâmetros do sistema através dos ajustes dos ganhos  $K_p$ ,  $K_i$  e  $K_d$ . A equação para a saída do controlador PID no domínio do tempo é dada pela equação (70):

$$u_{PID}(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (70)$$

A mesma pode ser reescrita da seguinte forma:

$$u_{PID}(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (71)$$

A função de transferência do PID é:

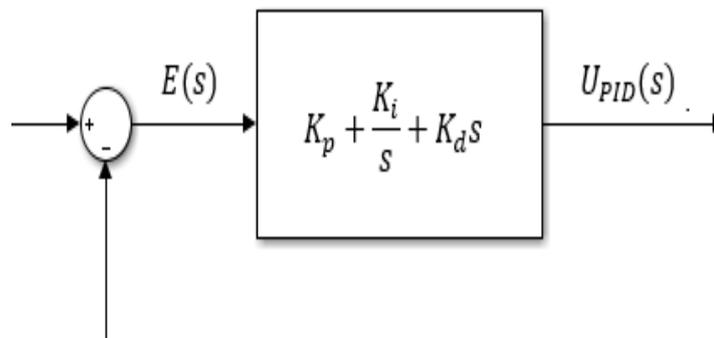
$$\frac{U_{PID}(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \quad (72)$$

Que também pode ser escrita como:

$$\frac{U_{PID}(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) \quad (73)$$

A Figura 21 mostra o diagrama de bloco de um controlador proporcional-integrativo-derivativo (PID).

Figura 21 – Diagrama de bloco de um controlador PID.



**Fonte:** Autoria própria.

Os controladores PID são controladores autônomos que apresentam bom desempenho em sistemas cujo processo se comporta de forma razoavelmente linear, ou seja, onde uma mudança na entrada gera uma mudança proporcional na saída. Contudo se a relação entre a entrada e saída do sistema for altamente não-linear, ou quando são usados na malha de realimentação elementos de controle ou atuadores cujo o comportamento é substancialmente não-linear, ou até mesmo quando a modelagem matemática do processo não é conhecido (seja pela falta de conhecimento ou complexidade da planta), o controlador PID apresenta um desempenho pobre (SIMÕES, 2007, p. 66).

Em processos levemente não-linear ou quase-linear, onde o desvio da linearidade não é tão grande, ou até mesmo quando a função não-linear de entrada e saída do processo apresenta um comportamento suave, livre de saltos súbitos e descontinuidade, dentro de uma pequena região nas vizinhanças do ponto de operação, o sistema pode ser considerado linear e o controlador PID pode ser usado com despenho satisfatório, entretanto o mesmo ainda necessitará de ajustes periódicos de seus parâmetros (SIMÕES, 2007, p. 66).

O controlador PID pode controlar todos os parâmetros do sinal de saída do sistema, através dos ajustes dos ganhos proporcional ( $K_p$ ), integrativo ( $K_i$ ) e derivativo ( $K_d$ ). Quando o processo é linear, esses ajustes são independentes uns dos outros de forma que podem ser manuseados separadamente, um por vez (SIMÕES, 2007, p. 68). Cada ganho promove um efeito na saída do sistema, como pode ser visto na Tabela 2, em que mostra o efeito individual dos ganhos em relação aos parâmetros do sistema quando o mesmo é submetido a uma entrada ao degrau.

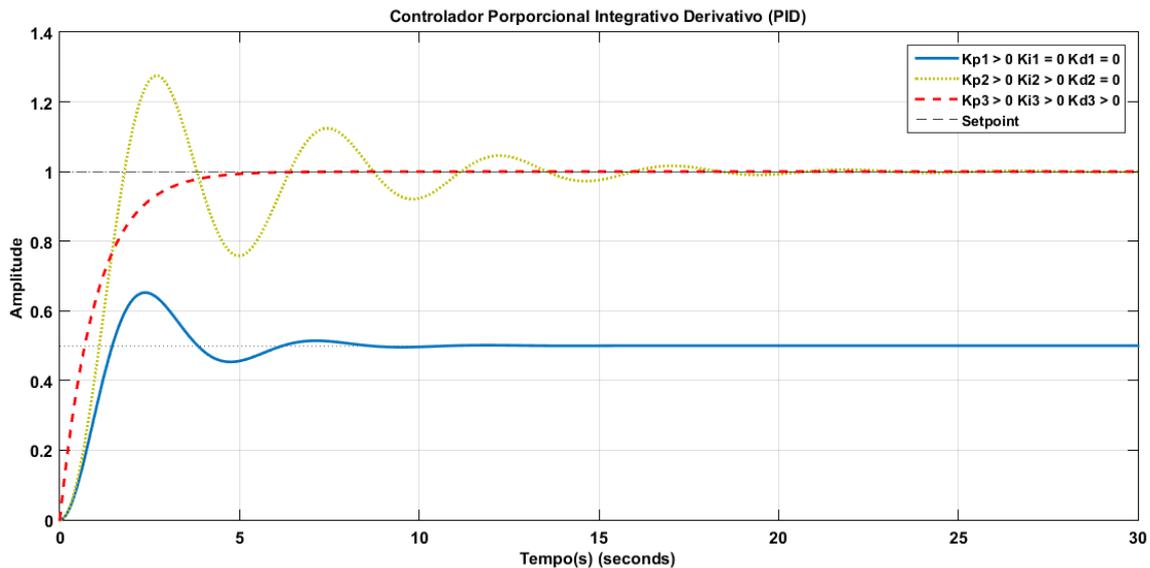
Tabela 2 - Efeito individual de cada ganho sob a resposta de sistema com entrada ao degrau.

Parâmetro	Tempo de Subida	Sobressinal	Tempo de Acomodação	Erro Estacionário
$K_p$	Diminuição	Aumento	Pequena mudança	Diminuição
$K_i$	Diminuição	Aumento	Aumento	Diminuição Significativa
$K_d$	Pequena diminuição	Diminuição	Diminuição	Sem efeito em teoria

**Fonte:** Adaptado de SILVA, 2015, p. 38.

A Figura 22 mostra a resposta da saída de um sistema de ordem zero para uma entrada em degrau, aplicando o controlador PID e variando os valores dos ganhos proporcional ( $K_p$ ), integrativo ( $K_i$ ) e derivativo ( $K_d$ ). A curva azul (continua) representa a resposta do sistema com apenas um ganho proporcional ( $K_{p1} > 0$ ), sem o ganho integral ( $K_{i1} = 0$ ) e nem o derivativo ( $K_{d1} = 0$ ). A curva em amarela (pontilhado) é a resposta do sistema quando se mantêm o ganho proporcional ( $K_{p1} > 0$ ), o derivativo ( $K_{d1} = 0$ ) e se incrementa ganho integral ( $K_{i1} > 0$ ). Por fim a curva em vermelho (tracejado) é a resposta do sistema quando se mantêm o ganho proporcional ( $K_{p1} > 0$ ), o ganho integral ( $K_{i1} > 0$ ) e se incrementa ganho derivativo ( $K_{d1} > 0$ ).

Figura 22 - Resposta de um sistema de ordem zero para uma entrada em degrau onde se aplica um controle proporcional-integrativo-derivativo (PID).



Fonte: Autoria Própria.

O ajuste desses ganhos de forma a obter as especificações de desempenho desejado no projeto de um controlador é chamando de sintonia do controlador, essa sintonia pode ser realizada manualmente por meio de tentativa e erro, através do conhecimento heurístico do sistema e manipulação dos ganhos ou aplicando alguma das técnicas matemáticas desenvolvidas que auxiliam o programador a encontrar mais rapidamente os valores de ajustes necessários. (OGATA, 2011, p. 522).

Na literatura existem várias técnicas de projeto de controladores PID, a maioria delas só podem ser aplicadas quando já se tem o modelo matemático da planta. No entanto existem casos em que o sistema é muito complexo, de modo que seu modelo matemático não pode ser obtido facilmente, nesses casos é necessário recorrer a abordagens experimentais de sintonia de controladores PID.

A fim de se evitar o trabalho excessivo decorrente da sintonia manual do controlador PID, os pesquisadores Ziegler e Nichol sugeriram regras que ajudam o programador a realizar essas sintonias experimentais dos controladores PID. Eles propuseram dois métodos de sintonia, o primeiro método propõem os ajustes conforme os valores de  $K_p$ ,  $T_i$  e  $T_d$ , baseadas na resposta experimental ao degrau da saída do sistema, o segundo método a sintonia é feita com base no valor de  $K_p$  que resulta em uma estabilidade marginal, quando só uma ação proporcional de controle é usada (ou seja,  $T_i = \infty$  e  $T_d = \infty$ ) (OGATA, 2011, p. 522).

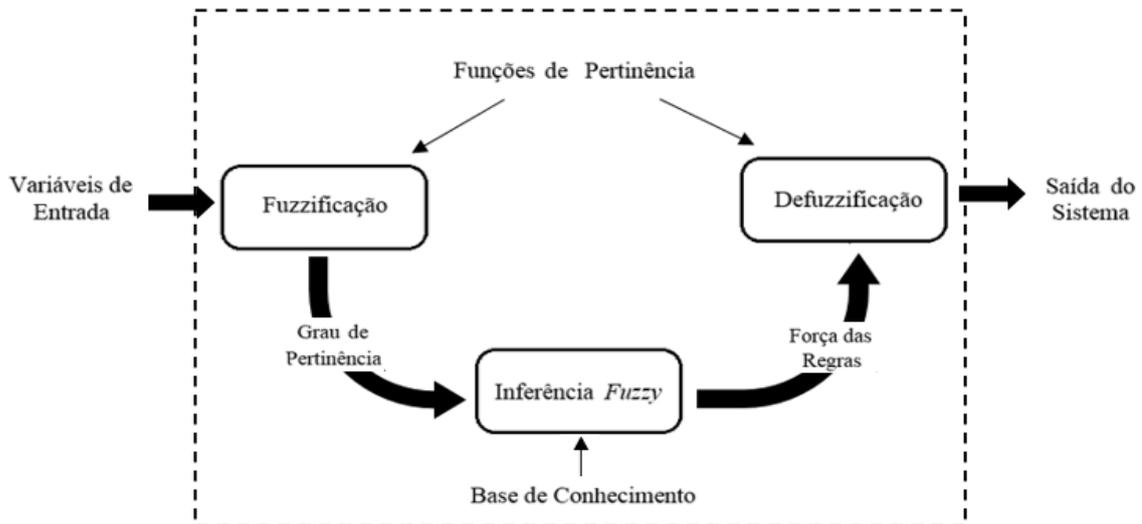
Em geral as regras de Ziegler-Nichol, são úteis quando não se conhece o modelo matemático da planta (o que não significa que não possa ser usado quando se dispõe do mesmo), porém, ele fornece apenas uma estimativa para os valores definitivos de sintonia, aos quais, na maioria dos casos, precisam passar por uma série de ajustes finos até que se alcance os parâmetros de saída desejado do projeto.

### **2.3.2. Controlador *Fuzzy***

A ciência tem buscado ao longo dos anos diversas formas de estudar e compreender a mente humana. Essas pesquisas são relevantes pois elas trazem com sigio um leque de possibilidade em aplicações, que podem ser usadas tanto na medicina, para tratamento psicológicos de pacientes, e até mesmo a utilização prática nas engenharias, como por exemplo, o desenvolvimento de controladores inteligentes capazes de simular a tomada de decisão humana, permitindo substituir ou auxiliar o homem em tarefas como monitoramento, controle, segurança dentre outras. No entanto simular o raciocínio humano é um desafio bastante árduo, a mente do homem é uma ferramenta complexa capaz de trabalhar com fatores ambíguos e imprecisos, processando informações incertas com valores aproximados e ainda sim, pode refratar tais pensamentos em termos linguísticos e tomar decisões importantes para resolução de problemas com base em sua experiência (SUZUKI, 2012). Dessa forma, com o objetivo de compreender e replicar o raciocínio humano, as engenharias junto com ciência da computação deram origem alguns seguimentos de estudo nessa área, tais como, redes neurais, sistemas especiais, algoritmos genéticos e a lógica *Fuzzy*, onde juntas formam esse novo conceito chamado de sistemas inteligentes (SIMÕES, 2007).

Na Figura 23 mostra o esquema funcional de um controlador *Fuzzy*, o mesmo pode ser dividido em três blocos funcionais básicos: Fuzzificação ou codificação; Inferência *Fuzzy* (conjunto de regras e base de conhecimento) e a Defuzzificação ou Decodificação.

Figura 23 - Diagrama de funcionamento de um controlador Fuzzy.



**Fonte:** Adaptado de NETTO (2005).

Em resumo o controlador *Fuzzy* funciona da seguinte forma, primeiro é feita a etapa de fuzzificação onde as variáveis de entradas são codificadas e transformadas em variáveis *Fuzzy*, que nada mais é do que termos linguísticos naturais. Para realizar essa fuzzificação é necessário a utilização das funções de pertinência, que são responsáveis por classificar cada entrada em seu determinado conjunto *Fuzzy* com base no grau de pertinência atribuído a cada elemento. Após a fuzzificação das entradas é feita a etapa de inferência *Fuzzy*, onde se é realizado o cálculo de tomada de decisão com base nas regras pré-definidas pelo programador (o mesmo deve possuir um conhecimento heurístico sobre a dinâmica do processo da planta e transferir sua experiência de tomada de decisão em forma de um conjunto de regras de “SE” e “ENTÃO”), por fim é realizado a etapa de defuzzificação onde o valor dos cálculos são decodificados e transformados novamente em valores reais que serão interpretados como nova ação a ser tomada.

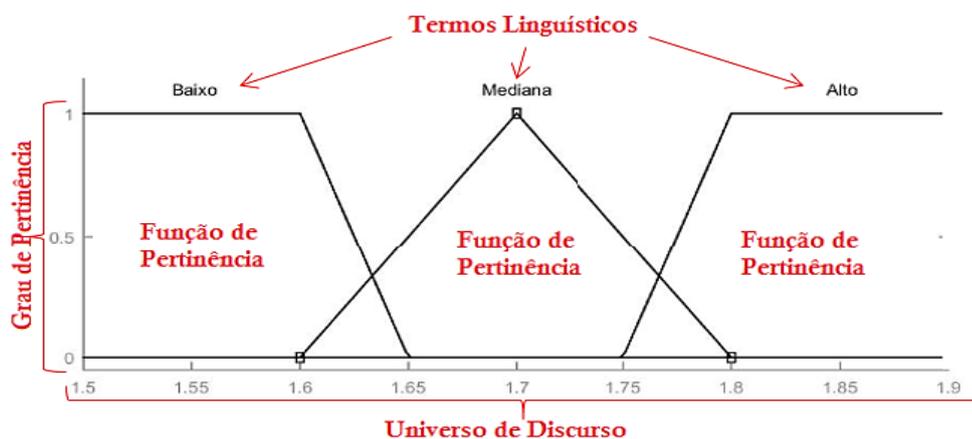
### ➤ Fuzzificação

A fuzzificação é a primeira etapa do processo do controlador *Fuzzy*, nela é feito codificação das variáveis de entrada para o domínio *Fuzzy*, transformando-as no que se chama de variáveis *Fuzzy*. Essas variáveis de entrada são geralmente valores reais discretos tais como dados de sensores e grandezas físicas enquanto as variáveis *Fuzzy* são termos linguísticos naturais usados para classificar um determinado conjunto, tais como: alto, baixo, grande,

pequeno, ruim, bom, etc. Para classificar essas variáveis de entrada nos conjuntos *Fuzzy*, a fuzzificação utiliza das chamadas funções de pertinência, essas funções são responsáveis por atribuir um grau de pertinência *Fuzzy* (valor entre 0 e 1), para cada elemento de entrada, esse grau de pertinência descreve o quanto é possível esse elemento pertencer a um determinado conjunto *Fuzzy*.

Na Figura 24 mostra um exemplo da utilização das funções de pertinência para classificação de uma variável, neste caso a classificação se refere à altura de uma pessoa. Para isso o programador usou os seguintes termos linguísticos para designar os conjuntos *Fuzzy*, que foram: “Baixo”, “Mediano” e “Alto”, nota-se que esses termos, como já discutido em tópicos anteriores, quantificam valores imprecisos em que sua definição pode variar de pessoa para pessoa, porém, diferentemente da lógica clássica, a lógica *Fuzzy* permite trabalhar com esse grau de incerteza. Assim como os termos linguísticos usados, o universo de discurso que preenche os conjuntos *Fuzzy*, também é escolhido pelo programador, o mesmo deve representar o intervalo numérico correspondente a todos os possíveis valores reais que a variável que está sendo analisada pode assumir. Para o exemplo em questão o intervalo escolhido foi entre 1,5 m e 1,9 m, em que três funções de pertinência foram utilizadas, duas retangulares e uma triangular, o intervalo de cada função também é definida com base na experiência do programador.

Figura 24 - Exemplo de funções de pertinência.



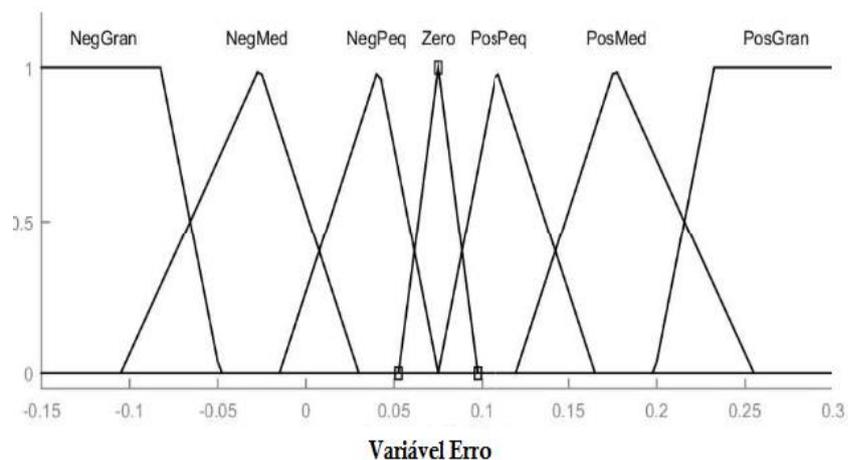
**Fonte:** Autoria própria.

Ainda na análise da Figura 24 é possível observar que a lógica *Fuzzy* também possibilita a superposição entre as funções de pertinência, o que permite aos elementos, a capacidade de pertencer a dois conjuntos simultaneamente. Essa intercessão entre os conjuntos afeta a precisão da resposta do controlador, dessa forma, recomenda-se o uso de um fator de intercessão de no

mínimo 25% e um máximo de 75%, para os primeiros testes, sendo 50% um fator razoável de utilização. (SIMÕES, 2007).

Outro ponto que está associado a precisão da resposta do controlador *Fuzzy* é a quantidade de funções de pertinência utilizadas. Essa quantidade também é definida pelo programador, que pode utilizar o número de funções que ele achar necessário para descrever a variável (veja a Figura 25), sendo que quanto maior o número de conjuntos, maior a precisão do controlador, porém, maior será também a demanda computacional exigida. Estudos apontam que uma quantidade maior do que sete funções de pertinência por variável não traz melhorias significativas nos resultados, apenas irá causar o aumentando no tempo de processamento, sendo um valor prático utilizado entre 2 e 7 funções por variável. (SIMÕES, 2007).

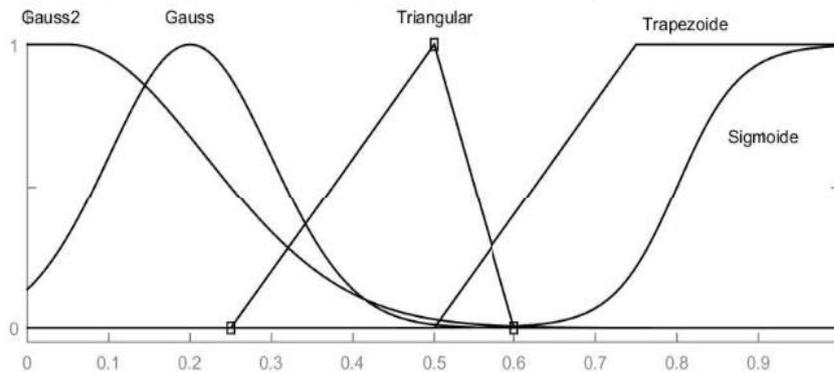
Figura 25 - Exemplo 02 de funções de pertinência.



**Fonte:** Autoria própria.

Além disso as funções de pertinência ainda podem variar em diferentes formas, os tipos mais comuns são os triângulos e trapezoides, mais também pode-se encontrá-las nas formas gaussiana, gaussiana<sup>2</sup>,  $\cos(x)$ ,  $\cos^2(x)$ , sigmoide dentre outras (veja a Figura 26). Cada uma dessas formas pode ser usada na representação de uma variável, a escolha da mais adequada varia com base na experiência e da perspectiva do programador em relação a dinâmica do sistema. (SIMÕES, 2007).

Figura 26 - Exemplo de diferentes formas de funções de pertinência.



**Fonte:** Autoria própria.

Com base na Figura 26 ainda é possível observar que as funções de pertinência não precisam ser obrigatoriamente simétricas ou igualmente espaçadas, cada variável pode ter um conjunto diferente de funções de pertinência que podem variar tanto no intervalo do universo de discurso como nos formatos e nas distribuições escolhidas para cada função.

Em resumo, a fuzzificação é responsável por gerar um vetor de pertinência para cada variável de entrada, onde esses vetores serão entregues posteriormente a etapa de inferência *Fuzzy*, que realizará a análise, através da base de conhecimento, efetuando uma tomada de decisão. (NETTO, 2005).

### ➤ Inferência Fuzzy

Após realizado todo o processo de fuzzificação das variáveis, entra agora etapa de inferência *Fuzzy*. Aqui será feito um relacionamento entre os elementos de entrada e os elementos de saída do sistema, através da base de conhecimento atribuída as regras de inferência *Fuzzy*. Essa base de conhecimento é adquirida por meio da consulta com um operador humano qualificado que tenha conhecimento sobre a dinâmica do sistema que está sendo controlado, esse especialista deve transferir a sua experiência profissional através de um conjunto de regras, essas regras devem expressar a sua tomada de decisão para o máximo de eventualidades que possa vir a surgir no sistema.

Segundo Simões (2007), as regras de inferência *Fuzzy* devem armazenar o máximo de conhecimento sobre o sistema *Fuzzy* e sua aplicação, elas devem estar no formato:

**SE** < condições > **ENTÃO** < conclusão >

Ou:

**SE** < *antecedente* > **ENTÃO** < *consequente* >

As < *condições* > ainda podem relacionar-se a valores linguísticos de uma ou mais variáveis através de conectivos:

**SE** < *P é O* > **E** < *U é L* > **ENTÃO** < *Q é K* >

Ou:

**SE** < *P é O* > **OU** < *U é L* > **ENTÃO** < *Q é K* >

Onde:

- *P*, *U* e *Q* são variáveis linguísticas que represente o próprio universo de discurso **P**, **U** e **Q**.
- *O*, *L* e *K* são conjuntos *Fuzzy* que representando valores linguísticos atribuídos as variáveis *P*, *U* e *Q* respectivamente.
- **E** e **OU** são conectivos de ligação, responsáveis pela combinação dos conjuntos.

Dois exemplos de construções de regras podem ser vistos nas frases a baixo:

– SE a bomba é grande ENTÃO a explosão é maior

↓ ↓      ↓ ↓  
Variável Valor      Variável Valor  
linguística linguístico      linguística linguístico

Conectivo

↑

– SE a temperatura esta alta E o volume está baixo ENTÃO o alarme é ligado

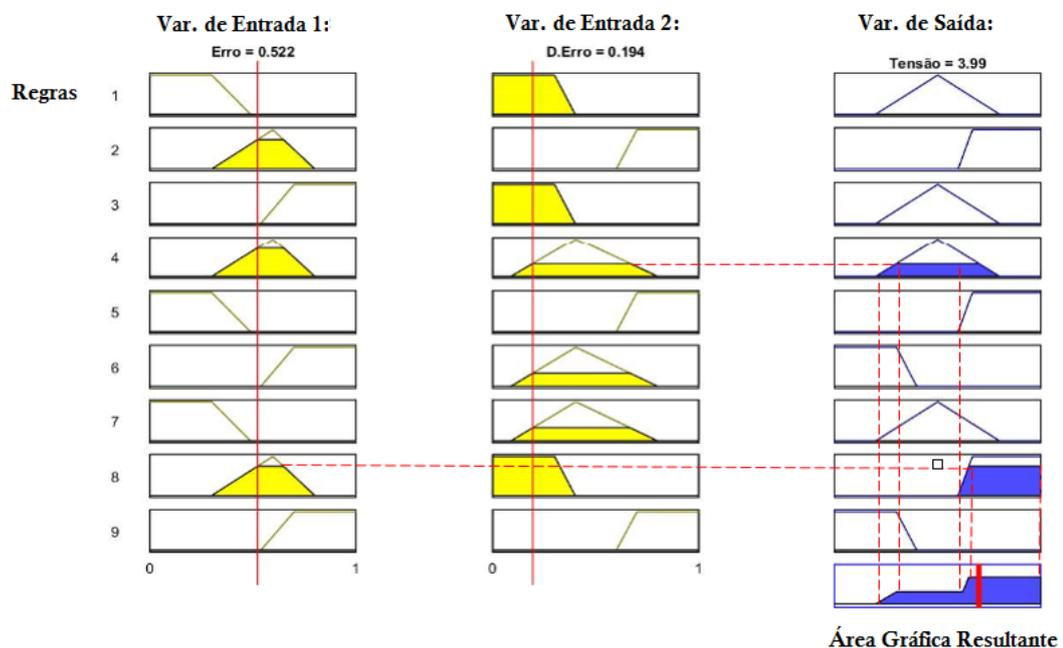
↓ ↓ ↓ ↓      ↓ ↓  
Variável Valor Variável Valor      Variável Valor  
linguística linguístico linguística linguístico      linguística linguístico

Segundo Simões (2007) existem vários operadores para combinar conjuntos associados às variáveis, a escolha adequada resulta em um sistema mais robusto com melhor funcionalidade. Os operadores mais comuns, utilizados em controladores *Fuzzy*, são os operadores *máx-mín*.

- *máx-mín*:

O *mín* ou operador de intersecção *Fuzzy*, está associado ao conectivo **E**, sua operação é chamada de agregação. Esse operador compara os valores de pertinência atribuído as variáveis que estão sendo associada pelo conectivo **E**, e trunca no menor valor entre as duas, de forma que sua saída é a área gráfica que está abaixo do valor de truncamento, conforme demonstrado na Figura 27.

Figura 27 - Exemplo do funcionamento do operado *mín* (conectivo **E**) nas regras de inferência *Fuzzy*.

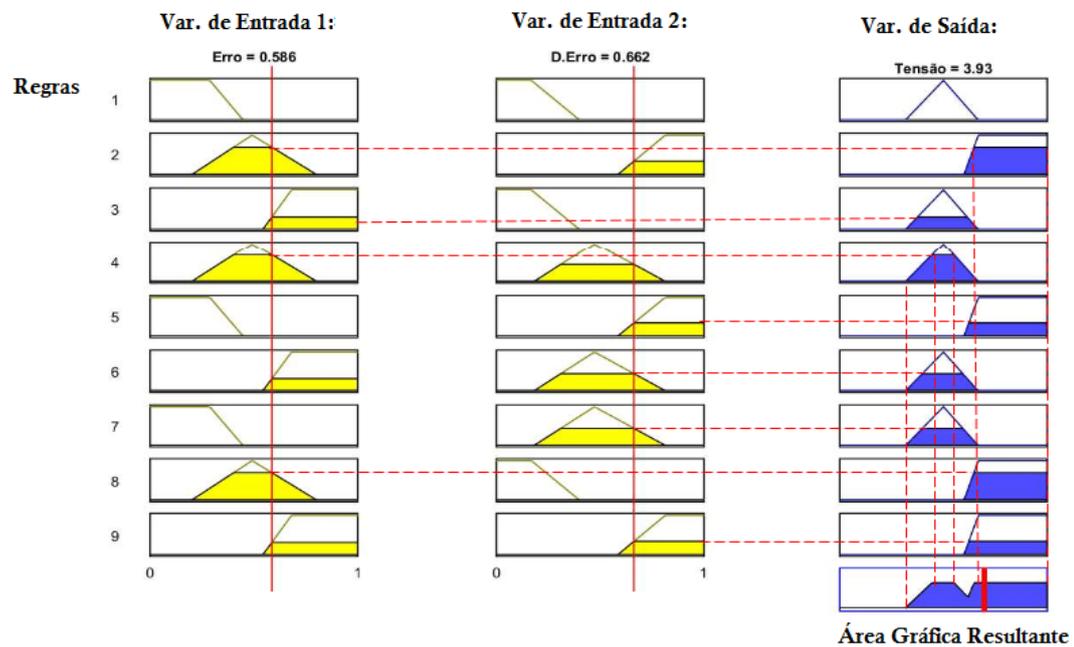


**Fonte:** Autoria própria.

Da mesma forma que no anterior, o operador *máx* compara os valores de pertinência que estão sendo atribuído as variáveis associadas nas regras, no entanto esse agora é representado pelo conectivo **OU**. A diferença é que truncamento ocorre no maior valor entre as duas variáveis, e não no menor, e sua saída, assim como no operador *mín*, é a área gráfica que está abaixo do valor de truncamento, conforme demonstrado na Figura 28.

Esses operadores podem ser usados juntos na elaboração do conjunto de regras, a escolha vai depender do sistema, sua aplicação e do conhecimento do especialista sobre o mesmo.

Figura 28 - Exemplo do funcionamento do operado *máx* (conectivo **OU**) nas regras de inferência Fuzzy.



**Fonte:** Autoria própria.

Em resumo a etapa de inferência *Fuzzy* ocorre da seguinte forma: As regras *Fuzzy* são processadas através de um bloco lógico onde todas elas são ativadas em paralelo, ou seja, simultaneamente. Dessa forma, quando fornecida uma entrada a um controlador *Fuzzy*, o mesmo aciona todas as regras ao mesmo tempo (chamada de inferência associativa paralela), cada regra é associada a um fator de pertinência, que indica o grau de validade das informações. O valor final resultante da associação paralela de todas as regras vai depender dos conectivos utilizados para elaboração de cada regra individualmente, mais no geral, esse bloco lógico resulta na inferência de uma área gráfica que representa o vetor das possíveis soluções para o sistema. Essa área gráfica resultante é gerada através da soma aritmética de todas as áreas individuais das regras que foram acionadas e ela simboliza a força das regras. Esse gráfico é enviado para defuzzificação onde será decodificada (através de um dos métodos de defuzzificação) e encontrado um valor discreto não-*Fuzzy* corresponde a resposta real do sistema (NETTO, 2005).

Existem vários modelos de inferência *Fuzzy*, os mais comuns são o método de Mamdani, que usa os operadores *máx-mín* para obter as conclusões das regras *Fuzzy*, e Takagi-Sugeno em que seu modelo dispensa a utilização de um método para defuzzificação, pois sua saída já é uma função que gera um valor não-*Fuzzy*. (FRANCELINE, 2016).

➤ Defuzzificação

A defuzzificação é a última etapa do processo da lógica *Fuzzy*, nela é feita a tradução ou decodificação dos valores das variáveis linguísticas de saída (inferidas pelas regras *Fuzzy*) em valores reais discretos (não-*Fuzzy*), ou seja, para cada variável de saída é gerado um valor único que melhor representa a possível solução do sistema, levando em conta todas as possibilidades envolvidas. Os métodos de defuzzificação mais usados são *Centro-da-Área (C-o-A)*, *Centro-do-Máximo (C-o-M)* e *Média-do-Máximo (M-o-M)*.

- ***Centro-da-Área (C-o-A)***: Também conhecido como *centro-de-gravidade*, é um método que calcula o centroide da área gráfica resultante da saída do bloco de inferência *Fuzzy*. Ele divide a área gráfica em duas partes iguais e calcula seu centro, utilizando a equação (74):

$$u^* = \frac{\sum_{i=1}^N u_i A_i}{\sum_{i=1}^N A_i} \quad (74)$$

Onde  $A_i$  é área de uma função de pertinência  $i$ ,  $u_i$  é a posição do centroide da função de pertinência  $i$ , e  $N$  é o número de funções de pertinência da variável linguística.

- ***Centro-do-Máximo (C-o-M)***: Esse método também é chamado de *defuzzificação pelas alturas*, ele ignora a área gráfica resultante das funções de pertinência e utiliza apenas os picos (valor de pertinência máximo) de cada função. Esses picos representam os pesos de cada regra, dessa forma o valor de saída discreto é calculado através de uma média ponderada de todos os máximos afim de se encontrar um ponto de apoio onde todos os pesos fiquem equilibrados, através da equação (75):

$$u^* = \frac{\sum_{k=1}^n \sum_{i=1}^N u_i y_k(u_i)}{\sum_{k=1}^n \sum_{i=1}^N y_k(u_i)} \quad (75)$$

Onde  $u_i$  é o ponto onde ocorre o máximo da função de pertinência  $i$ , e o  $y_k(u_i)$  é o valor do grau de pertinência da função  $i$  da regra  $k$ . O  $N$  representa o número de funções de pertinência e o  $n$  é o número de regras ativadas pela função  $i$ . (NETTO, 2005).

- ***Média-do-Máximo (M-o-M)***: Esse método calcula a média de todos os máximos, através da equação (76):

$$u^* = \sum_{m=1}^M \frac{u_m}{M} \quad (76)$$

Onde  $u_m$  é o m-ésimo elemento do universo de discurso onde a função de pertinência tenha o máximo valor, sendo que M é o número total desses elementos.

No geral a lógica *Fuzzy* é bem flexível, ela permite o tratamento de expressões de grandezas não exatas, onde sua busca por tomada de decisão é feita por meio da utilização de um senso comum, permitindo trabalhar com valores intermediários e expressões linguísticas mal definidas, o que antes não era possível com a utilização da lógica clássica (MARRO *et al*, 2010). Dessa forma, a lógica *Fuzzy* é considerada por muitos pesquisadores, como sendo o método que mais assemelha ao raciocínio humano, sendo bastante usada em projetos de controladores que necessitam de tomadas de decisão, como é o caso da proposta de trabalho dessa dissertação.

### 2.3.3. Controlador Híbrido *Fuzzy*-PID

Em um sistema supervisório os parâmetros do sinal de saída do processo são constantemente supervisionados e precisam ser mantidos dentro das margens de tolerância do ponto de operação estabelecido pelo programador, o objetivo é controlar as eventuais perturbações que venham a surgir no sistema mantendo o mesmo sempre operando dentro dos padrões desejáveis. Esse tipo de controle é comumente encontrado dentro de processos industriais complexos, esses sistemas requerem controladores sofisticados que realizem com eficiência a ação de controle adequada de forma a garantir o bom desempenho da planta. Um dos controladores mais utilizados nesse tipo de processo é o controlador de três termos o PID.

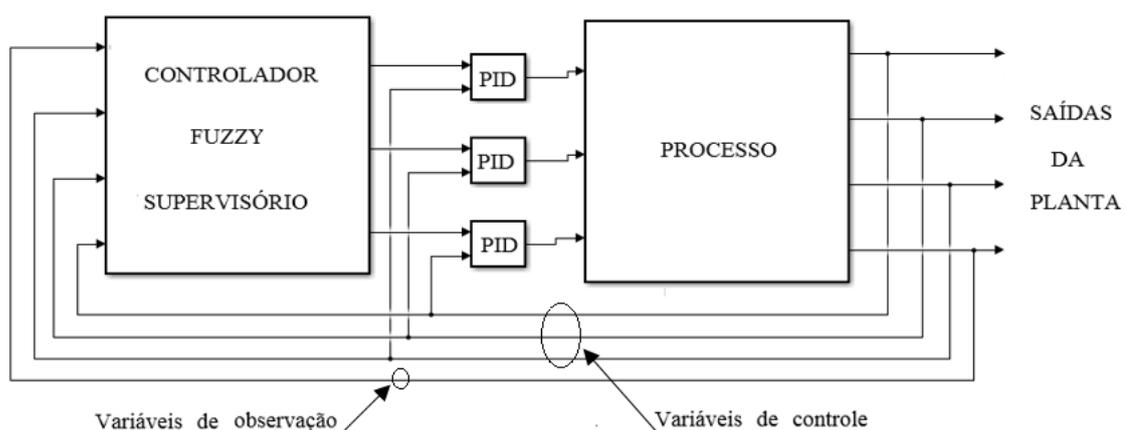
O problema em se usar controladores PID em sistemas de controle supervisório é que o PID controla apenas as variáveis do processo de forma que os ajustes de seus parâmetros são feitos por intermédio de um operador humano, que realiza a supervisão do sistema e faz a sintonia manual dos ganhos quando detectado uma curva fora do ponto de operação. Além disso, como já foi discutido, os controladores PID possuem desempenho pobre em sistemas não-lineares o que limita sua aplicação em alguns processos industriais. Com tudo, a lógica *Fuzzy* pode fornecer uma solução eficiente para esse problema, ela pode ser combinada ao controlador PID e realizar a etapa de supervisão do sistema, substituindo o operador humano. Além disso o *Fuzzy* atribui ao controlador a vantagem de poder trabalhar com sistemas não-lineares, ele permite ao programador dispensar o modelo matemático da planta e realizar as

ações de controle utilizando apenas a experiência profissional de um especialista sobre o sistema, esse profissional por meio das regras de inferência *Fuzzy* transfere seus conhecimentos aplicando todas as possíveis soluções para se controlar as saídas da planta, tornando o controle completamente autônomo, mais rápido e mais eficiente (SIMÕES, 2007, p. 71).

Vale lembrar ainda que os controladores PID são controladores de uma entrada e uma saída, logo, em aplicações de controle de sistemas multivariáveis, deve-se utilizar um controlador PID para cada variável com seu próprio ciclo de realimentação, isto é, independente uns dos outros (SIMÕES, 2007, p. 68). Entretanto o controlador *Fuzzy* supervisor pode ser comum a todos os controladores PID, desde que as variáveis possuam o mesmo comportamento sistemático ou possuam relação entre si com a mudança do comportamento do sistema, caso contrário, será necessário também um controlador *Fuzzy* supervisor para cada controlador PID.

A Figura 29 mostra um exemplo de um esquema de controle utilizando o controlador híbrido *Fuzzy*-PID, nele o *Fuzzy* fica responsável por supervisionar a saída do sistema fornecendo os ajustes necessários dos ganhos do controlador PID, enquanto que o PID recebe as variáveis de controle, normalmente o erro (dado pela diferença entre a saída da planta e o ponto de operação desejado), e realiza o controle do sistema conforme os ajustes enviados pelo controlador *Fuzzy* supervisor, que deve detectar e enviar os ajustes de ganho necessários para corrigir esse erro.

Figura 29 – Exemplo de um esquema de controle utilizando o controlador híbrido *Fuzzy*-PID



**Fonte:** Adaptado de SIMÕES, 2007, p. 72.

Diferentemente da utilização do controlador PID puro, em que os ajustes dos parâmetros de saída do sistema dependem da percepção e agilidade do operador humano em sintonizá-los, o incremento do controle *Fuzzy* supervisor permite que esses ajustes possam ser feitos durante

as modificações do *set-point*, antecipando o erro e suavizando as perturbações. Em outras palavras, no controle PID puro, o operador humano ao detectar o erro, teria que realizar a sintonia e esperar até que o sistema respondesse a esses ajustes, e em seguida realizar uma nova sintonia caso o erro não tivesse sido solucionado, já no controlador *Fuzzy*-PID, devido a velocidade de resposta do controle *Fuzzy* supervisor e sua capacidade de aprendizado (caso tenha sido bem programado, munido de todas as possíveis eventualidades e soluções do sistema), ele permite realizar uma sintonia simultaneamente com a variação do sinal de saída, tornando o controle do sistema muito mais rápido.

Ao se construir um controlador híbrido *Fuzzy*-PID devesse lembrar da expressão do controlador PID no domínio do tempo que também pode ser analisada da seguinte forma:

$$du_{pid} = K_p e + K_i de + K_d d^2 e \quad (77)$$

Onde  $e$  é o erro entre o ponto de referência e o sinal de saída do sistema,  $de$  é a variação desse erro,  $d^2e$  é a variação da variação desse erro e  $du_{pid}$  é a variação do sinal de controle do PID.

Dessa forma, segundo Simões (2007) é sugerido que o controlador *Fuzzy* supervisor possua três entradas:  $e$ ,  $de$  e  $d^2e$  e uma saída  $du_i$ , onde as regras devem ter o seguinte formato:

$$\mathbf{SE} \langle erro = E_i \rangle \mathbf{E} \langle variação do erro = dE_i \rangle \mathbf{E} \quad (78)$$

$$\langle variação da variação do erro = d^2E_i \rangle \mathbf{ENTÃO}$$

$$\langle variação do controle = du_i \rangle$$

Concluindo, o controle híbrido *Fuzzy*-PID reúne todas as vantagens de ambas as técnicas de controle, porém também requer o conhecimento aprofundado em ambos os assuntos, tais como a definição das funções de pertinência, as técnicas de fuzzificação, defuzzificação assim como a sintonia do controlador *Fuzzy*-PID através dos parâmetros ( $K_p$ ,  $K_i$  e  $K_d$ ). O desenvolvimento desse tipo de controlador exige um certo conhecimento de programação assim como um conhecimento de um especialista sobre o funcionamento da planta, dessa forma para a elaboração desse controlador foram necessários longos períodos de simulação e testes práticos com o Drone até se obter a experiência necessária.

### 3. MATERIAIS E MÉTODOS

Nesse capítulo será apresentado o método utilizado para o desenvolvimento do controlador híbrido *Fuzzy*-PID assim como a descrição dos materiais utilizados. O Capítulo foi dividido em três seções, na seção 3.1 é destinada ao conhecimento referente ao modelo do quadrimotor escolhido para realização das ações de controle propostas nesta dissertação, apresentando sua estrutura e os sensores aos quais este modelo dispõe. Na seção 3.2, é mostrado a estrutura da plataforma AuRoRA, que foi utilizada para realização dos testes e validação do controlador proposto, vale salientar ainda que tal plataforma foi disponibilizada para uso nesta dissertação apenas para fins didáticos. Por fim, a seção 3.3, descreve o passo a passo do desenvolvimento do controlador híbrido *Fuzzy*-PID proposto, mostrando como foi desenvolvido e quais os métodos e técnicas escolhidas para sua elaboração.

#### 3.1. AR.DRONE PARROT 2.0

O modelo do quadrimotor utilizado para validação por simulação do controlador proposto nessa dissertação foi o Ar.Drone Parrot 2.0 (veja Figura 3), cuja a modelagem dinâmica foi brevemente descrita no tópico 2.2, que segue com o formalismo de Euler-Lagrange.

O Ar.Drone Parrot 2.0 é munido dos seguintes sensores (PARROT, 2014):

- **Acelerômetro:** É um instrumento capaz de medir a aceleração em um dado eixo de movimento, o Ar.Drone possui 3 acelerômetros um em cada eixo ( $x, y, z$ ) com +/- 50mg de precisão, o que permite medir sua velocidade nas três direções.
- **Giroscópio:** É um instrumento utilizado para medir o ângulo de giro da aeronave, ele mensura as inclinações do veículo, no Ar.Drone esse instrumento possui a precisão de 2000°/segundo, o que permite saber a orientação angular do veículo em relação ao referencial inercial.
- **Magnetômetro:** é um sensor utilizado para medir tanto a direção quanto a magnitude dos campos magnéticos da Terra, no caso do Ar.Drone ele possui Magnetometro nos 3 eixos com 6° de precisão para o controle de direção da aeronave.
- **Sensor de pressão:** Como o próprio nome já diz, é um sensor utilizado para medir a pressão atmosférica, no AR.Drone esse sensor possui +/- 10 Pa de precisão (80

cm no nível do mar) e é utilizado para determinar a altura do veículo segundo o princípio de pressão estática do ar, ao qual a pressão cai quase linearmente com o aumento da altitude.

- **Ultra-som:** É um sensor utilizado para detectar objetos a uma determinada distância, no Ar.Drone este sensor é utilizado para medir a altura do veículo, no entanto seu alcance é limitado, possuindo uma faixa limite entorno de seis metros.

O Ar.Drone é um modelo comercial disponível para compra em seu site: [www.parrot.com](http://www.parrot.com), esse Drone possui uma estrutura fixa (*frame*) reforçada em forma de ‘×’, como discutido na seção 2.2.1, em cada extremidade do *frame* contém um motor de *brushless inrunner* sem escovas de 14.5 Wats e 28.500 RPM, possuindo um total de quatro motores, os motores possuem uma relação indireta com as hélices, ou seja, a rotação passa primeiro por um multiplicador de engrenagem de Nylatron de baixo ruído, que aumenta a rotação dos mesmos exigindo menos esforço dos motores, o eixo da hélice é de aço temperado. O Ar.Drone gera uma rede *wireless* ao qual é utilizada para a comunicação sem fio com outros dispositivos como *Smartphone*, *tablets* e computadores o que substituem a função do rádio controlador. Ele ainda possui duas câmeras, uma câmera em HD com 720 pixel a 30 FPS que está na parte frontal do veículo alinhada ao eixo  $^b x$ , é ela quem determina o sentido positivo da movimentação do Drone, e outra câmera QVGA de 60 FPS com lente panorâmica que está apontada para baixo, no sentido negativo do eixo  $^b z$ , essa câmera é usada para medição da velocidade terrestre, ambas também possuem a capacidade de captura de fotos em JPEG, podendo armazená-las durante o voo, para posteriormente retirada via USB ou transmitir diretamente ao dispositivo em tempo real via *Wi-Fi*. O centro do Ar.Drone é feito de espuma de alta tecnologia que isola o seu centro de inércia das vibrações do motor, nela está contido o processador ARM Cortex A8 de 1 GHz de 32 bits com vídeo de 800 MHz DSP TMS320DMC64x, a memória RAM de 1GB DDR2 a 200 MHz, e o sensores apresentados anteriormente, o Drone ainda possui duas armações de espuma, uma com proteção lateral das hélices (para utilização *indoor*) e outra sem, conforme visto na Figura 3, dessa forma o peso do Drone varia sendo mesmo pesando 380g com a armação sem proteção das hélices, e 420g com a armação com proteção das hélices. O controlador do motor é completamente reprogramável, é o que permite a utilização de uma plataforma para programação de uma nova estratégia de controle, além disso o Drone possui uma paragem de emergência controlada por *software*, no caso do travamento das hélices ou quando o veículo ultrapassa dos ângulos limites de rolagem, ou seja, caso ele fique emborcado.

A lista completa dos componentes do Ar.Drone Parrot 2.0 pode ser encontrado no site oficial da Parrot.

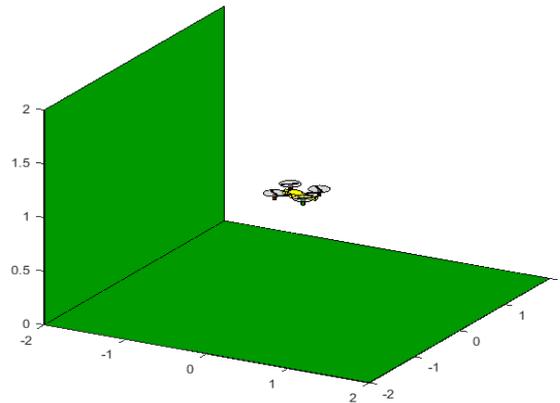
### 3.2. PLATAFORMA AuRoRA

Para que se possa validar uma estratégia de controle, seja ela aplicada a qualquer tipo de sistema, é necessário primeiro que haja a realização de inúmeros testes práticos que possa comprovar e garantir o bom funcionamento do mesmo (SUZUKI, 2009). No entanto, quando se trabalha no controle de sistemas dinâmicos, tais como o controle de VANTs (que são veículos inerentemente instáveis, não lineares, que possuem múltiplas variáveis com alto grau de acoplamento), a realização desses testes práticos pode se tornar bastante arriscado, à grande possibilidade de erros do controlador pode ocasionar acidentes, principalmente na realização de teste em ambiente fechados, podendo levar a prejuízos materiais ou até mesmo danos físicos a terceiros. Como solução, os programadores utilizam do artifício computacional, desenvolvendo simuladores capazes de reproduzir de forma mais aproximada possível o comportamento físico do sistema, e assim realizar os testes com o controlador de forma segura, validando-o e reduzindo consideravelmente o tempo de desenvolvimento do mesmo.

Dentro desse contexto, como o objetivo principal desta dissertação é projetar um controlador capaz de guiar um quadrimotor (modelo Ar.Drone Parrot 2.0) em um voo autônomo vertical (controle apenas de altitude), será necessário a utilização de uma plataforma que seja capaz tanto de simular o voo deste veículo como também consiga se comunicar com o mesmo para que se possa realizar a comprovação da eficiência do controlador através do teste prático. Dessa forma, optou-se por utilizar a plataforma AuRoRA (*Autonomous Robots for Research and Application*), essa plataforma é capaz de simular e experimentar VANTs com diferentes modelos matemáticos e diferentes estratégias de controle. Essa plataforma foi alvo de estudo e desenvolvimento em Brandão (2013) e em PIZETTA (2013), sendo a mesma disponibilizada para uso exclusivamente acadêmico desta dissertação, sendo a mesma aberta a introdução de uma nova estratégia de controle (neste caso, o controlador híbrido *Fuzzy-PID* proposto), a plataforma ainda permite realizar a comunicação sem fio com o Ar.Drone Parrot, através de um link de comunicação *wireless* criado pelo próprio Drone, com isso é possível enviar sinais de controle e receber dados de voos oriundos dos sensores da aeronave (para realizar essa troca de informação a plataforma utiliza o *Software Development Kit – SDK*, disponível no sítio de internet da própria Parrot, empresa que fabrica o Drone utilizado nessa dissertação), a plataforma ainda oferece uma visualização gráfica em 3D do plano de voo da aeronave

permitindo acompanhar toda a movimentação do Drone no plano tridimensional, como pode ser visto pela Figura 30 (BRANDÃO, 2013; PIZETTA, 2013).

Figura 30 – Exemplo da visualização gráfica em 3D do plano de voo da aeronave fornecido pela plataforma AuRoRA.



**Fonte:** Autoria própria

A seguir é descrito como ocorre o funcionamento estrutural da plataforma AuRoRA.

### 3.2.1. Estrutura e funcionamento da plataforma AuRoRA

A plataforma AuRoRA foi desenvolvida segundo o modelo *hardware-in-loop* (HIL), onde os dados correspondentes a uma parte dos componentes do sistema são obtidos via simulação através do modelo matemático dos mesmos, enquanto que os dados correspondentes aos demais componentes do sistema são obtidos diretamente da aeronave (no caso, sinais de posição captados pelos sensores do veículo, são enviados diretamente a plataforma que os utiliza em suas equações). É importante ressaltar que os simuladores HIL estão diretamente associados ao *hardware* ao qual ele simula, pois, sua malha de simulação lida com a representação do ambiente ao qual esse *hardware* (neste caso o quadrimotor) vai trabalhar, com isso a plataforma AuRoRA é munida da capacidade de simular diferentes condições de ambiente de trabalho, permitindo realizar uma avaliação geral do comportamento do veículo, o que ajuda bastante no projeto do controlador (BRANDÃO, 2013; PIZETTA *et al*, 2012).

A estrutura da plataforma AuRoRA ocorre conforme o algoritmo apresentado na Figura 31 desenvolvida no Matlab, ela segue uma programação sequencial, e é caracterizada como sendo um sistema em tempo real, onde cada ação e resposta do sistema obedecem a intervalos de tempo bem definidos, ou seja, após ocorrer uma ação, a próxima só ocorrerá após um intervalo fixo de tempo. Tal intervalo refere-se ao período de amostragem do VANT que está

sendo simulado, que neste caso é o Ar.Drone, esse tempo de amostragem é de  $t_{sq} = 0,0333 s$ . É possível observar que a plataforma é dividida em vários módulos, aos quais podem ser ativados ou não (exemplo, o programador pode optar por desativar a geração de gráficos, para reduzir o esforço computacional, ou ativar o controle manual pelo *joystick* que sobrepõem o controle automático, dentre outras opções disponíveis).

Figura 31 - Algoritmo da estrutura da plataforma AuRoRA

---

**Algoritmo 1:** Estrutura da Plataforma.

---

```

Inicialização;
while  $t < t_{max}$  do
  if Permissão de execução then
    Ler Sensores;
    Calcular posição desejada;
    if Existe joystick then
      | Ler comandos;
    else
      | Aplicar controlador;
    end
    Enviar sinal de controle;
    Armazenar variáveis;
  end
  if Permissão para gráfico then
    | Exibir parte gráfica;
  end
end
end

```

---

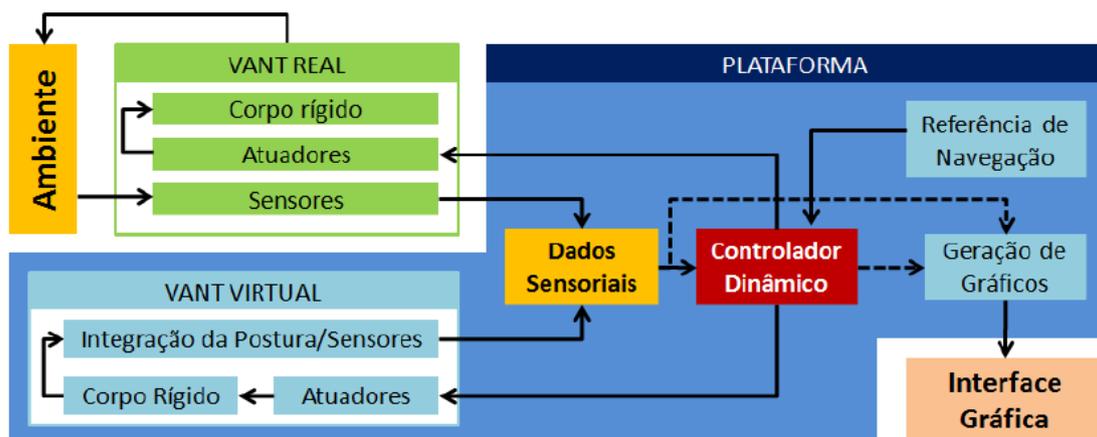
**Fonte:** BRANDÃO, 2013; PIZETTA, 2013.

A Figura 32 mostra o diagrama de bloco de como ocorre o funcionamento interno da plataforma AuRoRA. A plataforma oferece duas opções, o teste prático utilizando o Drone real ou a simulação através da integração numérica dos modelos matemáticos obtidos de cada componente. Na simulação, primeiramente é inserido os valores de referência de navegação que vão para o bloco do controlador dinâmico, os sinais de controle são enviados ao modelo dinâmico da aeronave que são os blocos dos atuadores e o do corpo rígido, onde é feito a integração da postura/sensores (tudo dentro do bloco VANT virtual, pois este se trata de uma simulação), a resposta da integração numérica fornece a postura futura da aeronave que é tomado como dados sensoriais e são enviados novamente ao controlador, que irá verificar a existência de erros e efetuar as correções necessária, todos os dados de voo são armazenados, e podem ser plotados através do bloco de geração de gráficos. (BRANDÃO, 2013)

Na aplicação real com o Drone, a plataforma precisa identificar a comunicação que foi estabelecida via *wi-fi*, para que se possa iniciar o processo de coleta de dados dos sensores e o envio dos sinais de controle para aeronave. A plataforma exige um *flag* de permissão de

execução para poder iniciar o processo, caso contrário nenhuma ação é executada e os dados recebidos e comandos serão descartados. Ativando o *flag* de permissão é iniciado o processo, primeiro os comandos de referência de navegação são enviados ao controlador dinâmico que envia os sinais de controle para os atuadores (neste caso os motores do Drone) que irão responder alterando sua velocidade de rotação conforme o programado, em resposta o Drone executará a manobra referente a ação dos atuadores, interagindo com as possíveis perturbações do ambiente, os sensores do Drone iram captar cada movimento executado pela aeronave e os envia como dados sensórias, que retornam novamente ao controlador dinâmico onde será verificado os erros de postura e executado o comando de correção, o ciclo se repete até o fim do plano de voo. Uma observação importante é que no teste real, a geração de gráfico retém o custo computacional bastante alto sendo colocada em segundo plano durante o processo, pois a prioridade da plataforma é em enviar os sinais de controle. (PIZETTA, 2013)

Figura 32 - Diagrama de bloco do funcionamento em torno da plataforma AuRoRA



Fonte: PIZETTA, *et al* 2012.

Na versão final da plataforma AuRaRA (que não será mostrada nesta dissertação) é possível utilizar o seu modelo descentralizado, ao qual permite dividir o processo em dois computadores, um sendo responsável por gerar a simulação gráfica, enquanto o outro é responsável pela integração numérica do controlador, reduzindo o esforço computacional, e solucionando o problema da geração de gráfico no teste real com o Drone, para mais detalhes sobre seu funcionamento veja em Brandão (2013) e Pizetta (2013).

### 3.3. DESENVOLVIMENTO DO CONTROLADOR HÍBRIDO *FUZZY*-PID

Como já explicado ao longo desta dissertação, a proposta deste trabalho é desenvolver um controlador que seja capaz de realizar controle de altitude de um Drone. Dentre as técnicas de controle disponíveis na literatura, optou-se por realizar uma junção de duas estratégias de controle, o controlador clássico PID e a lógica *Fuzzy* formando o controlador híbrido *Fuzzy*-PID.

O Drone utilizado foi o Ar.Drone Porret 2.0 descrito na seção 3.1, a comunicação e o controle desse veículo foram feitos por intermédio da plataforma AuRoRA descrita na seção 3.2. Como a plataforma foi desenvolvida no Matlab, logo todo o script do controlador proposto nesta dissertação foi desenvolvida em forma de algoritmo correspondente a essa linguagem de programação.

O método utilizado para desenvolver o controlador *Fuzzy*-PID ocorreu em duas etapas, a primeira foi o desenvolvimento do controle PID clássico, ao qual é realizada uma série de teste simulados e práticos com o mesmo aplicando uma sequência de ajustes manuais de ganhos  $K_p$ ,  $K_i$  e  $K_d$ , o objetivo é sintonizar o controlador para se obter a resposta desejada ao mesmo tempo em que se pretende conhecer o comportamento dinâmico do sistema. Além disso o desenvolvimento do controlador clássico PID será posteriormente utilizado como comparativo com o controlador híbrido proposto. A segunda etapa foi a introdução do controlador *Fuzzy* supervisor, para construção desse controle foi exigido todo conhecimento adquirido durante os testes do controlador PID, para que fosse possível desenvolver um conjunto de regras que executasse de forma eficiente os ajustes da sintonia automática do PID. Nos tópicos que se segue, serão descritos como foi feito a construção do controlador PID e o controlador *Fuzzy* supervisor e a junção dessas duas estratégias de controle, dando origem ao controle híbrido *Fuzzy*-PID.

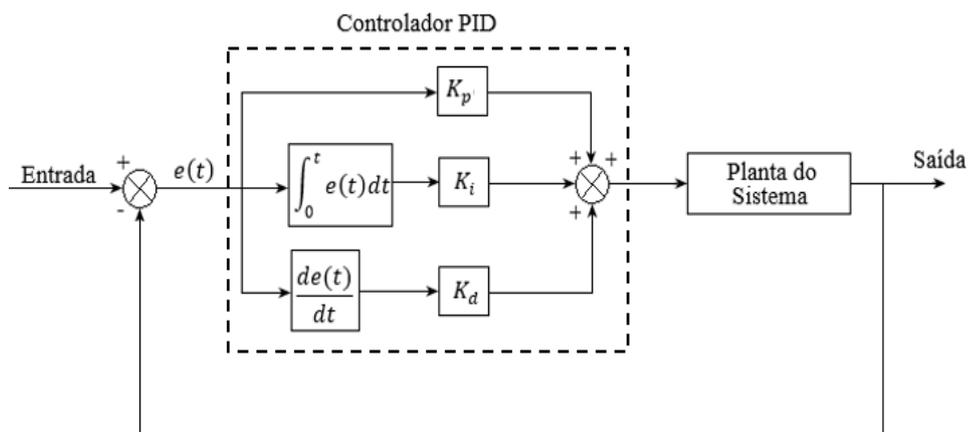
#### 3.3.1. Construção do controlador PID

Para a construção do controlador PID foi tomado como referência os trabalhos: de Sá (2012), ao qual é proposto a construção de um controlador PID e uma variação do PID para estabilidade de um quadrimotor; o de Vilas Boas e Honório (2013), em que se compara as técnicas de controle PID com outras técnicas de controle, visando a estabilização e segurança de quadrimotor; Guimarães (2012), onde se utiliza um controlador PID para estabilização de um quadrimotor e obtém bom desempenho do mesmo, até na presença de distúrbios; Bouabdallah

(2004) em que propõem um controlador PID para estabilizar a atitude do quadrimotor e também apresentada uma comparação de custo-benefício entre cinco tipos de veículos aéreos, dentre outros trabalhos complementares sobre o assunto.

Como já discutido na seção 3.3.1 os controladores PID são sistemas que controlam apenas uma variável por vez, dessa forma seria necessário um controlador PID para cada uma das seis variáveis que formam o sistema de referência inercial  $\langle g \rangle$  da posição do espaço tridimensional do quadrimotor ( ${}^g\mathbf{q} = [x \ y \ z \ \phi \ \theta \ \psi]^T \in \mathbb{R}^6$ ), caso o objetivo fosse desenvolver um controle de atitude, no entanto, neste trabalho o desenvolvimento está voltado para o controle de altitude, em que se precisa realizar o controle do veículo apenas em um eixo, neste caso, o movimento ao longo do eixo  ${}^g z$ , logo um controlador PID será utilizado, a Figura 33 representa o diagrama de bloco do controlador PID utilizado para o controle de altitude do quadrirrotor.

Figura 33 - Diagrama de bloco do controlador PID de altitude do quadrirrotor.



**Fonte:** Autoria própria

A primeira dificuldade encontrada para a elaboração do controlado PID, está relacionada a complexidade em se modelar esse tipo de veículo, ao qual não foi possível obter a função de transferência da dinâmica total do sistema, sem a função de transferência do sistema as técnicas convencionais que facilitam a sintonia dos ganhos do controlador PID, como a análise pelo lugar geométrico das raízes, análise pelo critério de estabilidade de Routh-Hurwitz, dentre outras, não poderão ser aplicadas, requerendo uma sintonia manual dos ganhos.

Uma solução para facilitar os ajustes manuais dos ganhos seria optar por utilizar um dos dois métodos de sintonia proposto por Ziegler-Nichol, uma vez que por meio da plataforma AuRoRA é possível simular e analisar o comportamento da saída da planta do sistema, que é um dos pré-requisitos essenciais para aplicação dessas técnicas. Entretanto os métodos propostos por Ziegler-Nichol requerem ainda que o sistema apresente, ou uma resposta no

formato de “S” para uma entrada ao degrau (que é o pré-requisito para aplicação do primeiro método), ou encontrar um valor de ganho que resultasse em uma estabilidade marginal do sistema (ou seja, a saída seria um sinal oscilando indefinidamente, que é o pré-requisito para aplicação do segundo método), ambas as tentativas não obtiveram êxito, de forma que a sintonia teve que ser realizada através de tentativa e erro, aumentando ainda mais o tempo de desenvolvimento do controlador, lembrando ainda que o controlador PID precisa realizar o ciclo completo de controle para que se possa exibir uma resposta, ou seja, ele não permite a sintonia manual durante o processo, obrigando ao operador realizar um ajuste por vez até que se alcance os parâmetros desejados.

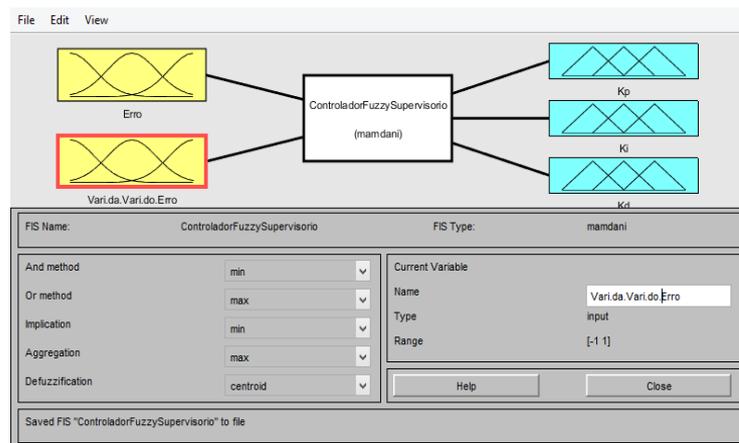
Dessa forma o diagrama de bloco do controlador PID apresentado Figura 33 possui a seguinte lógica de funcionamento, primeiramente é introduzido os valores iniciais dos ajustes dos ganhos  $K_p$ ,  $K_i$  e  $K_d$  (que requerem todo o conhecimento descrito pela seção 2.3.1), o bloco do controlador PID é formado pela equação (72) que resulta no sinal de controle indireto (neste caso o controle de altitude  $u_{\dot{z}}$ , que vai controlar a taxa de elevação vertical  $\dot{z}$ ) o sinal é enviado para a planta do sistema, onde está contido todo o modelo matemático da dinâmica do quadrimotor, resultando nos valores futuros da posição, orientação e velocidade em cada eixo, esses dados retornam para o início do controlador através da realimentação da malha, que serão comparados com os valores de postura desejados, gerando o sinal de erro atuante  $e(t)$ . O erro é enviado ao controlador PID que irá gerar a ação de controle, afim de se eliminar o erro, o ciclo continua até atingir o fim da operação.

### 3.3.2. Desenvolvimento do controlador *Fuzzy* supervisório

Como solução para evitar a sintonia manual dos ganhos, foi construído um controlador *Fuzzy* supervisório que será responsável por realizar o ajuste automático dos ganhos, a construção desse controlador teve como referência os trabalhos de Suzuki (2009), onde obteve bom desempenho em uma proposta de controle híbrido *Fuzzy*-PID para estabilização de VANT de asas fixas, o de Abbasi, *et al* (2013), em que é proposto um controlador *Fuzzy* supervisório para os ajustes de um controlador PID para estabilização de um VANT de asas rotativas, o trabalho de Netto (2005) em que é feito um estudo comparativo da influência das modificações dos parâmetros *Fuzzy* na resposta de um processo industrial (fornecendo conhecimento sobre a influência da modificação dos formatos das funções de pertinência sob a resposta do sistema), e o de Marro, *et al* (2010) onde é apresentado os conceitos e aplicações da lógica *Fuzzy*.

Para o desenvolvimento do controlador *Fuzzy* supervisor foi preciso utilizar a *toolbox* da lógica *Fuzzy* do Matlab, apresentado na Figura 34, essa ferramenta permite selecionar os parâmetros de entrada e saída do controlador *Fuzzy*, a elaboração das regras no formato SE-ENTÃO, dando a opção de escolher o modelo de Mamdani ou Takagi-Sugeno, além de escolher a técnica de defuzzificação.

Figura 34 - Toolbox da lógica *Fuzzy* no Matlab



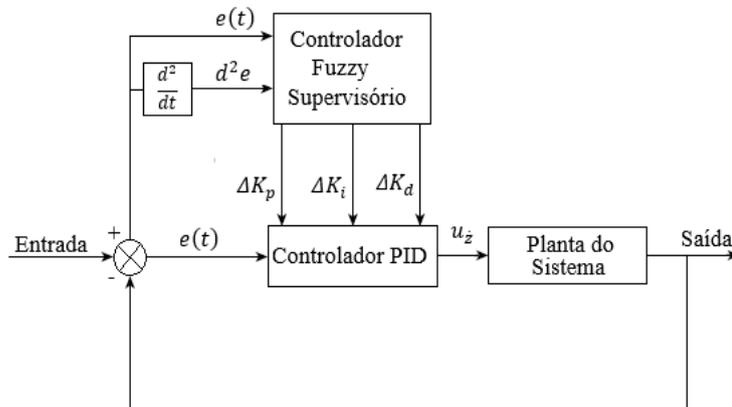
**Fonte:** Autoria própria.

Dessa forma, para a definição dos métodos de fuzzificação e defuzzificação foi seguindo a referência de Suzuki (2009) e Abbasi, *et al* (2013), onde se tem bons resultados através da utilização do modelo de Mandani. A composição da inferência *Fuzzy* foi usado o *máx-mín*, para a defuzzificação foi utilizado o método do centroide conforme explicado na seção 2.3.2. A criação das regras de sintonia do controlado *Fuzzy* foram feitas através do conhecimento adquirido ao longo do período de desenvolvimento do controlador PID, ao qual permitiu adquirir experiência sobre a dinâmica do sistema, por meio dos testes e do processo de sintonia do controlado por tentativa e erro. As funções de pertinência do controlador *Fuzzy* foram normalizadas para facilitar os ajustes dos ganhos PID, sendo utilizadas as funções de pertinência com formas triangulares e trapezoidais, assim como proposto em Suzuki (2009).

Para as entradas do controlador optou-se por utilizar o erro e a variação da variação do erro, e como saída, utilizou-se a variação dos ganhos  $\Delta K_p$ ,  $\Delta K_i$  e  $\Delta K$ , que são enviados para o controlador PID. A Figura 35 mostra o diagrama de bloco do controlador *Fuzzy* supervisor junto ao controlador PID formando o controle híbrido *Fuzzy*-PID proposto nesta dissertação, por meio desse bloco é possível ver que o controlador *Fuzzy* supervisor só vai começar a

funcionar a partir do surgimento do primeiro erro, pois sua função, como o próprio nome sugere, é supervisionar a saída do sistema e executar uma ação de ajuste dos ganhos quando necessário.

Figura 35- Diagrama de bloco do controlador *Fuzzy* supervisório



Fonte: Adaptado de ABBASI, *et all* (2013)

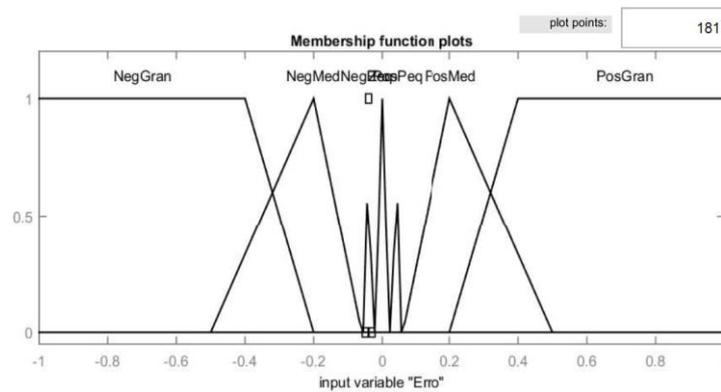
O funcionamento do diagrama de bloco ocorre da seguinte forma, primeiramente é introduzido o *set-point* da altura desejada junto com os valores iniciais dos ganhos, diferente do controlador PID puro, a utilização do *Fuzzy* supervisório dispensa a sintonia manual, de forma que se pode introduzir um valor inicial de ajuste aleatório, pois o mesmo irá reajustar esses valores de forma que o sistema atinja os parâmetros desejados. No primeiro ciclo de operação os valores de entradas são enviados para o controlador PID em que é realizado o processo de controle utilizando os valores iniciais sugerido, o sinal de controle é enviado a planta do sistema, a resposta de saída retorna para o início do controle através da malha de realimentação, proporcionando o possível erro do sistema, entretanto, esse erro agora é levado para o controlador *Fuzzy* supervisório, sendo antes dividido em duas entradas, o erro e a variação da variação do erro. Através das funções de pertinências e regras do controlador é realizado um ajuste para cada ganho, que posteriormente é enviado para o controlador PID, a utilização da variação dos ganhos ao invés de um range de ganho pré-definido, permite que o controlador *Fuzzy* atribua uma série de incrementos ou decrementos sucessivos até se obter os ajustes necessários para eliminar o erro (ou seja, ele é atribuído da capacidade de aprender sobre o sistema, o que permite, teoricamente, utilizar qualquer valor inicial de ganhos), o ciclo continua até o fim da operação de controle.

Para se chegar nos valores dos parâmetros desejados de controle, foram necessários o desenvolvimento de 28 (vinte e oito) controladores *Fuzzy* supervisório, alterando várias características do projeto até obter o controle mais adequado. Dentre os modelos construídos,

dois obtiveram os melhores resultados, o *Fuzzy 23* e o *Fuzzy 28* (que é uma adaptação do *Fuzzy 23*). A seguir serão mostradas a construção do controlador *Fuzzy 23* e em seguida as modificações aplicadas que resultaram no *Fuzzy 28*.

A função de pertinência de entrada do erro do controlador *Fuzzy 23* é demonstrada na Figura 36, em que são utilizadas sete funções de pertinência, duas trapezoidais e cinco triangulares, as variáveis linguísticas e a escala adotadas para cada um dos conjuntos *Fuzzy* foram: Erro negativo grande (NegGran) representado por uma função trapezoidal variando de -1 a -0.2; Erro negativo médio (NegMed) representado por uma função triangular variando de -0.5 a -0.06; Erro negativo pequeno (NegPeq) representado por uma função triangular variando de -0.06 a -0.03; Erro zero (Zero) representado por uma função triangular variando de -0.03 a 0.03; Erro positivo pequeno (PosPeq) representado por uma função triangular variando de 0.03 a -0.06; Erro positivo médio (PosMed) representado por uma função triangular variando de 0.06 a -0.5; E o erro positivo grande (PosGran) representado por uma função trapezoidal variando de 0.2 a 1. O universo de discurso escolhido foi os valores normalizados variando entre -1 e 1, que está relacionado com *set-point* (que é valor da altura desejada), o valor do erro é dado pela diferença entre o *set-point* e a altura atual do VANT, logo, se o erro for negativo indica que o Drone está acima do valor de *set-point*, se o valor de erro for positivo indica que o Drone está a baixo do valor de *set-point* (com o limite máximo inferior do erro de altura adotado para zero metros, ou seja, pousado no solo). Como o objetivo do controlador é estabilizar altura do Drone em um metro, foi considerado como sendo um erro grande, valores de erro superiores a 20% do *set-point* (referindo-se as funções de pertinência PosGran e NegGran), para as funções de pertinência NegMed e PosMed foi considerado como sendo um erro médio os valores de erro que estejam compreendidos entre 5 a 50% do *set-point*, para as funções de pertinência NegPeq e PosPeq foi adotado com sendo um erro pequeno os valores de erro que estejam entre 3 a 5% do *set-point*, e como sendo um erro aceitável ou erro zero, os valores de erros compreendidos entre 0 a 3% do *set-point*. Nota-se que a medida em que se aproxima do ponto central de equilíbrio (função de pertinência zero) as funções de pertinência ficam mais densa (juntas) e seu range se torna mais seletivo, não havendo intersecções entres os conjuntos *Fuzzy*, essa tática permite uma maior sensibilidade e uma melhor precisão nos ajustes dos ganhos, em contrapondo, a medida em que se afasta do ponto central o inverso ocorre, ou seja, as funções se afastam umas das outras, o range se torna maior e existe a presença de intersecções, essa tática permite um ajuste mais grosseiro melhorando o tempo de reposta do sistema para erros mais acentuados.

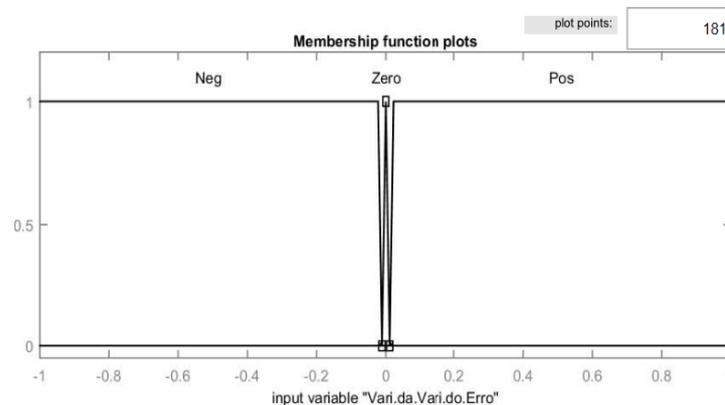
Figura 36 - Função de pertinência da primeira entrada do controlador *Fuzzy*: Erro



**Fonte:** Autoria Própria

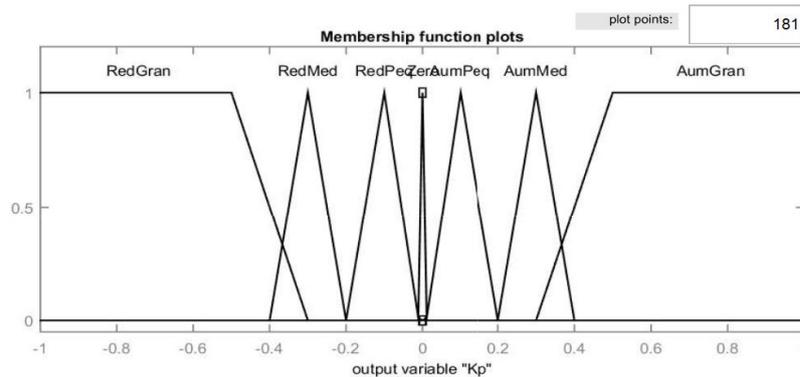
Para a segunda entrada do controlador *Fuzzy* 23, primeiramente optou-se por utilizar a derivada do erro ou variação do erro, entretanto, os testes com essa entrada não apresentaram o êxito esperado, devido à dificuldade da escolha para as funções de pertinência que descrevesse de forma eficiente o comportamento dessa variação do erro, como solução foi utilizado a variação da variação do erro, dando origem ao denominado *Fuzzy* 28 (mencionando anteriormente), com essa alteração o objetivo ficou em observação apenas do momento em que o erro muda de sinal, utilizado para identificar quando o erro está se afastando ou aproximando (positivamente ou negativamente) do *set-point*. A função de pertinência de entrada da variação da variação do erro do controlador *Fuzzy* 28 é demonstrada na Figura 37, ao qual utiliza três funções de pertinência, duas trapezoidais e uma triangular, as variáveis linguísticas e a escala adotadas para cada um dos conjuntos *Fuzzy* foram: Variação da variação do erro negativo (Neg) representado por uma função trapezoidal variando de -1 a -0.001; Variação da variação do erro zero (Zero) representado por uma função triangular variando de -0.001 a 0.001; E a variação da variação do erro positivo (Pos) representado por uma função triangular variando de 0.001 a 1. Dessa forma o universo de discurso escolhido foi os valores variando entre -1 e 1, uma vez que o interesse está no sinal e não no valor em si da variável, além do que esses valores foram tomados mediante os resultados obtidos de inúmeros testes realizados com quadrimotor, em que essa variação da variação do erro é sempre um valor pequeno compreendido dentro desse range. Para a função de pertinência zero, foi escolhido um range bastante pequeno de forma a detectar a mínima mudança do sinal, além disso a variação da variação do erro zero só ocorre quando a medição da postura do quadrimotor repete os valores de medida, isso ocorre em um intervalo de tempo bastante curto onde os sensores da aeronave captam valores de medição iguais, não havendo mudança de sinal da variação da variação do erro.

Figura 37- Função de pertinência da segunda entrada do controlador *Fuzzy*: Variação da Variação do Erro.



**Fonte:** Autoria Própria

Para a saída do controlador *Fuzzy* foram utilizadas três variáveis, que correspondem as variações dos ganhos  $\Delta K_p$ ,  $\Delta K_i$  e  $\Delta K$ , aos quais proporcionam a sintonia do controlador PID. As três variáveis possuem o mesmo formato, logo será tomado como referência para ilustração das saídas do controlador *Fuzzy* a variável  $\Delta K_p$ , demonstrada na Figura 38. Da mesma forma que nas variáveis de entrada, o universo de discurso para as saídas foram normalizadas nos valores entre -1 e 1, isso permite que o controlador *Fuzzy* realize o incremento ou o decremento dos ganhos de acordo com sua tomada de decisão (baseadas nas regras *Fuzzy*). Foram utilizadas também sete funções de pertinência, duas trapezoidais e cinco triangulares, as variáveis linguísticas e o range adotadas para cada um dos conjuntos *Fuzzy* foram: Redução grande do ganho (RedGran) representado por uma função trapezoidal variando de -1 a -0.3; Redução média do ganho (RedMed) representado por uma função triangular variando de -0.4 a -0.2; Redução pequena do ganho (RedPeq) representado por uma função triangular variando de -0.2 a -0.001; Sem incremento de ganho (Zero) representado por uma função triangular variando de -0.001 a 0.001; Aumento pequeno do ganho (AumPeq) representado por uma função triangular variando de 0.001 a 0.2; Aumento médio do ganho (AumMed) representado por uma função triangular variando de 0.2 a 0.4; E o aumento grande do ganho (AumGran) representado por uma função trapezoidal variando de 0.3 a 1. A normalização do universo de discurso permite uma sintonia gradativa do ganho, possibilitando o controlador optar por diminuir ou aumenta o ganho conforme a apresentação do erro.

Figura 38- Função de pertinência da saída variação do ganho  $\Delta K_p$  do controlador Fuzzy

Fonte: Autoria Própria

Por fim, foram elaboradas as regras do controlador *Fuzzy* supervisor, as quais expressam todas as experiências adquiridas durante a sintonia manual do controlador PID. Como a primeira variável de entrada possui sete funções de pertinência, e a segunda possui três, logo foram necessárias a criação 21 regras, utilizando o formato SE-ENTÃO, que está mostrado na Tabela 3.

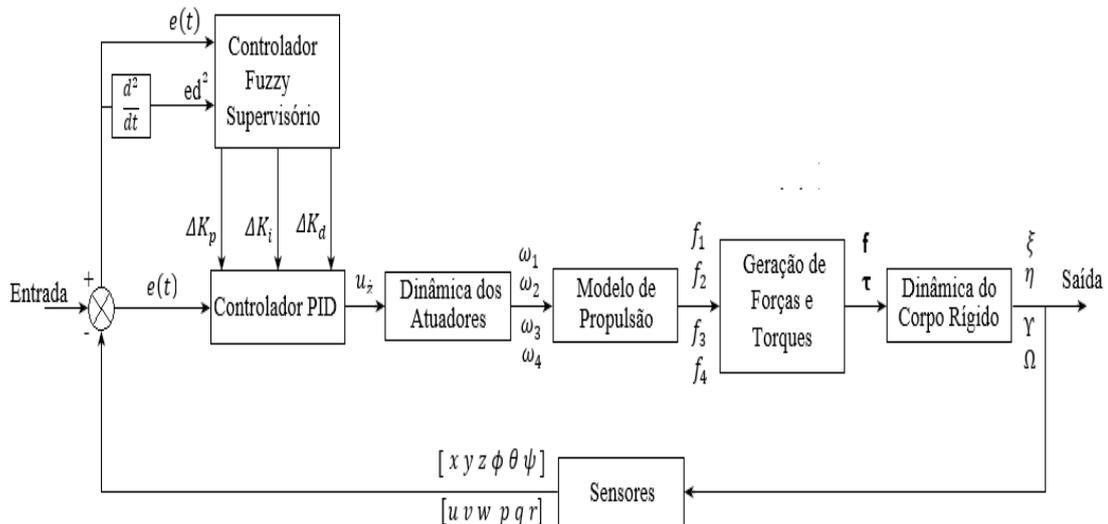
Tabela 3 - Regras desenvolvidas para o controlador *Fuzzy* supervisor

Variação da variação do erro				
Regras	Neg	Zero	Pos	
<b>Erro</b>	<b>NegGran</b>	(Kp = RedMed) <b>E</b> (Ki = AumPeq) <b>E</b> (Kd = AumGran)	(Kp = RedGran) <b>E</b> (Ki = RedGran) <b>E</b> (Kd = AumGran)	(Kp = RedGran) <b>E</b> (Ki = RedGran) <b>E</b> (Kd = AumGran)
	<b>NegMed</b>	(Kp = RedMed) <b>E</b> (Ki = RedMed) <b>E</b> (Kd = AumMed)	(Kp = RedMed) <b>E</b> (Ki = AumPeq) <b>E</b> (Kd = AumMed)	(Kp = RedGran) <b>E</b> (Ki = RedMed) <b>E</b> (Kd = AumMed)
	<b>NegPeq</b>	(Kp = AumPeq) <b>E</b> (Ki = RedPeq) <b>E</b> (Kd = AumPeq)	(Kp = Zero) <b>E</b> (Ki = Zero) <b>E</b> (Kd = AumPeq)	(Kp = RedPeq) <b>E</b> (Ki = RedPeq) <b>E</b> (Kd = AumPeq)
	<b>Zero</b>	(Kp = Zero) <b>E</b> (Ki = Zero) <b>E</b> (Kd = Zero)	(Kp = Zero) <b>E</b> (Ki = Zero) <b>E</b> (Kd = Zero)	(Kp = Zero) <b>E</b> (Ki = Zero) <b>E</b> (Kd = Zero)
	<b>PosPeq</b>	(Kp = AumPeq) <b>E</b> (Ki = RedPeq) <b>E</b> (Kd = AumPeq)	(Kp = AumPeq) <b>E</b> (Ki = Zero) <b>E</b> (Kd = Zero)	(Kp = AumPeq) <b>E</b> (Ki = RedPeq) <b>E</b> (Kd = AumPeq)
	<b>PosMed</b>	(Kp = RedMed) <b>E</b> (Ki = RedMed) <b>E</b> (Kd = AumMed)	(Kp = AumMed) <b>E</b> (Ki = AumMed) <b>E</b> (Kd = RedMed)	(Kp = AumPeq) <b>E</b> (Ki = RedPeq) <b>E</b> (Kd = AumMed)
	<b>PosGran</b>	(Kp = RedGran) <b>E</b> (Ki = RedGran) <b>E</b> (Kd = AumGran)	(Kp = AumPeq) <b>E</b> (Ki = AumMed) <b>E</b> (Kd = RedMed)	(Kp = AumGran) <b>E</b> (Ki = AumPeq) <b>E</b> (Kd = AumMed)

Fonte: Autoria Própria

Com a conclusão do controlador PID e o *Fuzzy* supervisor, a junção de ambas as técnicas formam o controlador híbrido *Fuzzy*-PID. A Figura 39 mostra o controlador híbrido *Fuzzy*-PID aplicado ao controle de altura do quadrimotor proposto nesta dissertação.

Figura 39 - Representação em diagrama de bloco do controlador híbrido *Fuzzy*-PID de altitude integrado ao modelo dinâmico do quadrimotor



**Fonte:** Adaptado de ABBASI *et all*, 2013; BRANDÃO, 2013

Dessa forma fica explicitado o diagrama de bloco completo do controlador híbrido *Fuzzy*-PID proposto nesta dissertação. Vale ainda ressaltar, com base na Figura 39, que o escopo deste trabalho está restrito apenas ao desenvolvimento do bloco do controlador *Fuzzy* supervisor e do bloco do controlador PID, sendo que os demais blocos (dinâmica dos atuadores, modelo de propulsão, geração de forças e torques, dinâmica do corpo rígido e sensores) já possuem sua estrutura pronta, estando os mesmos contemplados dentro da plataforma AuRoRA utilizada, para mais detalhes sobre o funcionamento desses blocos e os parâmetros do Drone utilizado (como peso, tamanho das hélices, torque dos motores, dentre outros) para o seu desenvolvimento, consulte Brandão (2013).

A seguir será dado início ao Capítulo 4, ao qual apresenta todos os resultados obtidos ao longo do desenvolvimento do projeto, assim como algumas discussões a respeito das dificuldades, problemas e soluções encontradas para o andamento/conclusão dos controladores.

#### 4. RESULTADOS E DISCUSSÃO

Este capítulo apresenta os resultados obtidos com o controlador PID clássico e o controlador híbrido *Fuzzy*-PID desenvolvido, com o objetivo de comparar e validar a sua eficiência.

A apresentação dos resultados obtidos foram divididos em três seções, primeiros os resultados simulados do controlador PID clássico e do *Fuzzy*-PID, em seguida a validação por meio dos resultados com os testes práticos com Ar.Drone, aplicando cada técnica de controle individualmente, e por fim, foi realizada a comparação entre as duas estratégias de controle com o objetivo de evidenciar possíveis melhorias proporcionada pela união desses dois controladores.

Ambas as estratégias de controle foram desenvolvidas com o objetivo de realizar o controle de um grau de liberdade, neste caso a altura do Ar.Drone, restringindo seu movimento apenas ao longo do eixo  $^g z$ , dessa forma pode-se dizer que o controle dos demais graus de liberdade são relaxados.

A plataforma AuRoRA é utilizada em ambos os testes, tanto o simulado quanto o prático, a linguagem de programação foi desenvolvida através do *software* Matlab. A plataforma é munida de uma matriz de perturbação o que torna as simulações mais próximas da realidade, entretanto os testes práticos foram realizados em ambiente fechado (*indoor*), reduzindo a ação de forças externas e possíveis perturbações, já que o controle ainda não está completamente aprimorado sendo o mesmo constituído apenas do controle de atitude do veículo.

Para cada controlador (o PID e o *Fuzzy*-PID) foram aplicados dois testes individuais com objetivo de verificar dois fatores importantes no desempenho das duas técnicas de controle. O primeiro teste tem como o objetivo saber se o controlador é capaz de estabilizar o Drone em uma certa altura e manter essa posição estável por um período longo de voo, o segundo teste tem como objetivo verificar se o controlador apresenta um bom desempenho para mudanças consecutivas do *set-point* (mudanças repentinas de altura), tendo em vista que a mudança de altura gera uma perturbação no sistema.

Para cada teste que será apresentado nos tópicos em que se seguem serão exibidos uma sequência de gráficos que mostrará as seguintes variáveis analisadas: O gráfico (a) mostra o controle de altitude do Drone (movimento apenas ao longo do eixo  $^g z$ ), ou seja, a altura em metros (m) que o Drone atinge a cada instante de tempo (segundos); O gráfico (b) mostra as entradas de controle indiretas, que são a saída do controlador PID, neste caso ela representa as forças geradas pelos propulsores na direção normal ao plano da aeronave, ou seja, a força de

empuxo necessário para sustentar o Drone, a sigla que denota essa variável é o  $u$  dada em Newton (N), o que é importante ressaltar sobre este gráfico é que ele mostra as contribuições positivas de força (idênticas para todos os motores) que resulta no movimento de subida do Drone, assim como as contribuições negativas que resultam no movimento de descida. Os gráficos (c) e (d) mostram os deslocamentos laterais temporal da aeronave com relação ao eixo  $^g x$  e  $^g y$ , respectivamente; Os gráficos (e), (f) e (g) mostram a evolução temporal da orientação da aeronave com relação aos ângulos: Phi –  $\phi$  (Rolagem ou Roll), Psi –  $\theta$  (arfagem ou Pitch) e Theta –  $\psi$  (guinada ou Yall), respectivamente, apesar de não haver, para esta dissertação, um controle específico sob essas variáveis as mesmas são evidenciadas com o propósito de justificar as limitações desses controladores além de servirem como parâmetros para trabalhos posteriores que venham tomar este projeto como referência. Vale mencionar ainda que nesses gráficos de apresentação de resultados, a linha tracejada representa os valores desejados ou *set-point*, enquanto que a linha contínua em azul mostram os valores efetivamente obtidos pelos sensores do Drone (para o caso experimental) ou obtidos por integração numérica (para o caso dos testes simulados) contidos na programação da plataforma AuRoRA.

#### 4.1. RESULTADOS COMPUTACIONAIS OU SIMULADOS

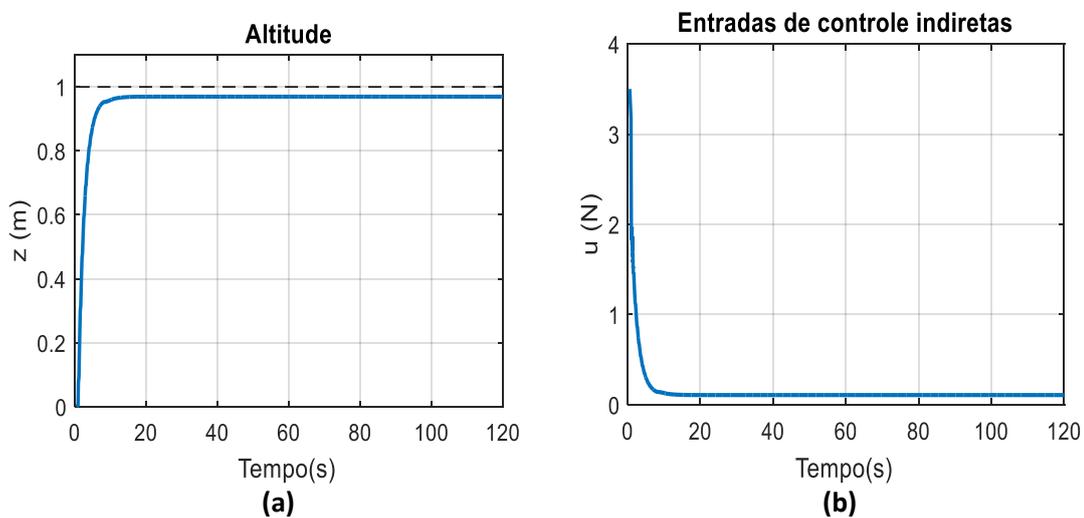
##### ➤ Resultados das simulação para o primeiro teste com ambos os controladores:

Para a aplicação do primeiro teste, foi simulado uma missão de voo onde o Drone deve atingir uma certa altura de valor fixado em 1 metro (*set-point* adotado para este trabalho) e após atingir essa altura o mesmo deve se manter o mais estável possível (é importante salientar que essa estabilização está relacionada somente a altura e não a eventuais movimentos nos demais graus de liberdade) durante todo o período de 120 segundos (2 minutos), que foi um tempo escolhido para os testes com base em outros trabalhos do gênero, como em Brandão (2013).

A Figura 40 mostra os resultados simulados do primeiro teste com o controlador PID clássico, apresentando a evolução temporal da altura do Drone e dos ganhos de força atribuídos as entradas de controle indiretas. Para esse teste foi utilizado os ganhos iniciais encontrados por meio de sintonia manual (necessários para esse tipo de controle) aos quais foram estabelecidos os seguintes valores:  $K_p = 1.5$ ,  $K_i = 1.5$  e  $K_d = 2$ . A Figura 40(a) apresenta o gráfico da evolução temporal da altitude (em metros) do veículo em relação ao tempo (segundos), com ele, é possível observar que há uma convergência do voo do Drone com relação ao *set-point*, é possível ver ainda uma subida rápida sem oscilações ao qual é mantido uma postura estável até o fim da simulação. Na Figura 40(b) é possível ver o comportamento temporal da variável de

controle indireta, que mostra a saída do controlador PID que atua sobre todos os motores simultaneamente (o objetivo de gerar a força de empuxo necessária para sustentação da aeronave), nesse gráfico é possível notar que o ganho inicial é alto em virtude do Drone está inicialmente em repouso no solo, as contribuições de força diminuem conforme a aeronave se aproxima do *set-point*, como este gráfico se trata de um simulação, após atingir o valor desejado os incrementos de força são estabilizados, sendo apenas inserido o suficiente para manter o Drone planando na altura desejada (que é o objetivo do primeiro teste).

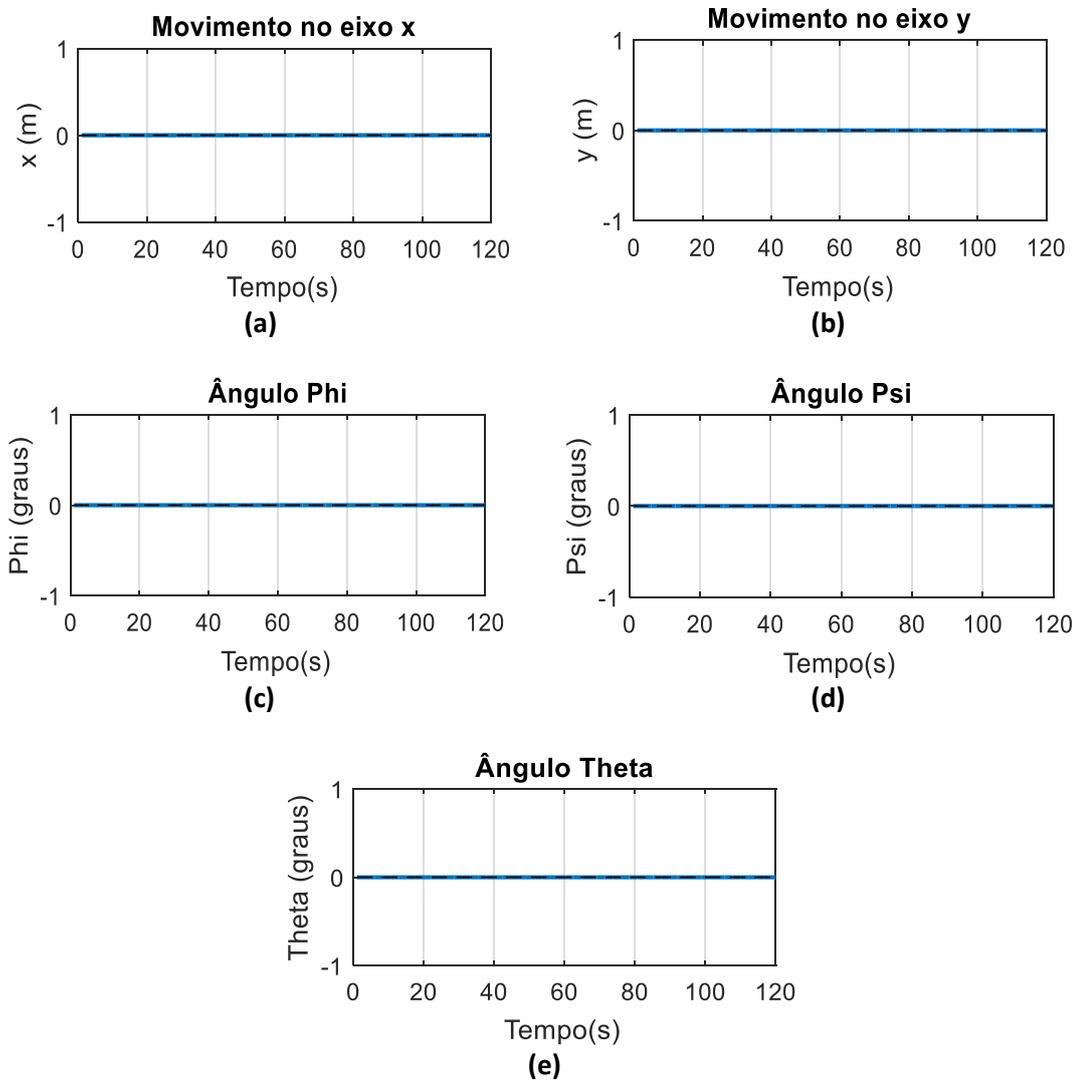
Figura 40 – Resultados simulados do primeiro teste com o controlador PID clássico: (a) Altitude; (b) Entradas de controle indiretas.



**Fonte:** Autoria Própria.

A Figura 41 mostra os resultados de outras variáveis ainda analisadas nesse primeiro teste simulado com o controlador PID clássico. Dentre elas, tem-se a Figura 41(a) e 41(b) que mostram o deslocamento do Drone em relação aos eixos  $^g x$  e  $^g y$ , enquanto que as Figura 41(c), 41(d) e 41(e) mostram a evolução temporal dos ângulos  $\phi$  (Phi), responsável pelo movimento de rolagem, do  $\theta$  (Psi), responsável pelo movimento de arfagem e do  $\psi$  (Theta), responsável pelo movimento de guinada. O que é importante destacar sobre esses gráficos é que apesar de não haver, para esta dissertação, um controle específico sob essas variáveis, as mesma não apresentaram variações acentuadas se mantendo constante em torno de zero, o que está de acordo por se tratar de uma simulação (apesar da existência de uma matriz de perturbação dentro das interações numéricas da plataforma, não foi suficiente para causar alteração nesse teste).

Figura 41 – Resultados simulados do primeiro teste com o controlador PID clássico: (a) Movimento no eixo x; (b) Movimento no eixo y; (c) Ângulo Phi; (d) Ângulo Psi; (e) Ângulo Theta

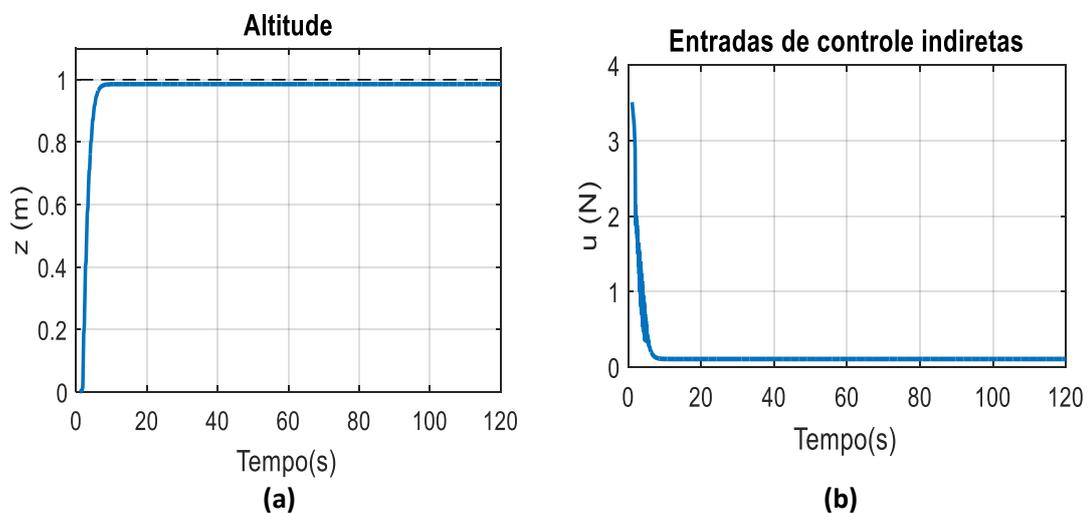


**Fonte:** Autoria Própria.

Da mesma forma como foi realizado no controlador PID puro, também foi aplicado o primeiro teste para o controlador *Fuzzy*-PID proposto. Para esse teste foi utilizado os mesmos ganhos iniciais que foram inferidos no PID puro,  $K_p = 1.5$ ,  $K_i = 1.5$  e  $K_d = 2$ , isso se deve ao fato de que o *Fuzzy* supervisor do controlador híbrido precisa de valores iniciais de ganho diferentes de zero, para que não faça um incremento exagerado necessitando assim de um referencial, foi escolhido os mesmo valores do controlador PID puro para que posteriormente pudesse compará-los. O resultado da simulação do primeiro teste com controlador híbrido *Fuzzy*-PID pode ser visto através da Figura 42(a) ao qual mostra a evolução temporal da altura do Drone, nesse gráfico é possível notar um subida suave sem oscilações apresentando um bom

resultado na estabilização da aeronave em um valor próximo ao *set-point*, assim também como Figura 42(b) em que mostra os ganho decorrentes das entradas de controle indiretas no tempo, da mesma forma como foi mostrado no controlador PID clássico, os detalhes da análise comparativa entre os dois controladores será realizado no tópico específico mais adiante, ao qual é destinado a comparação entre as duas estruturas de controle construídas, o PID clássico e o *Fuzzy*-PID.

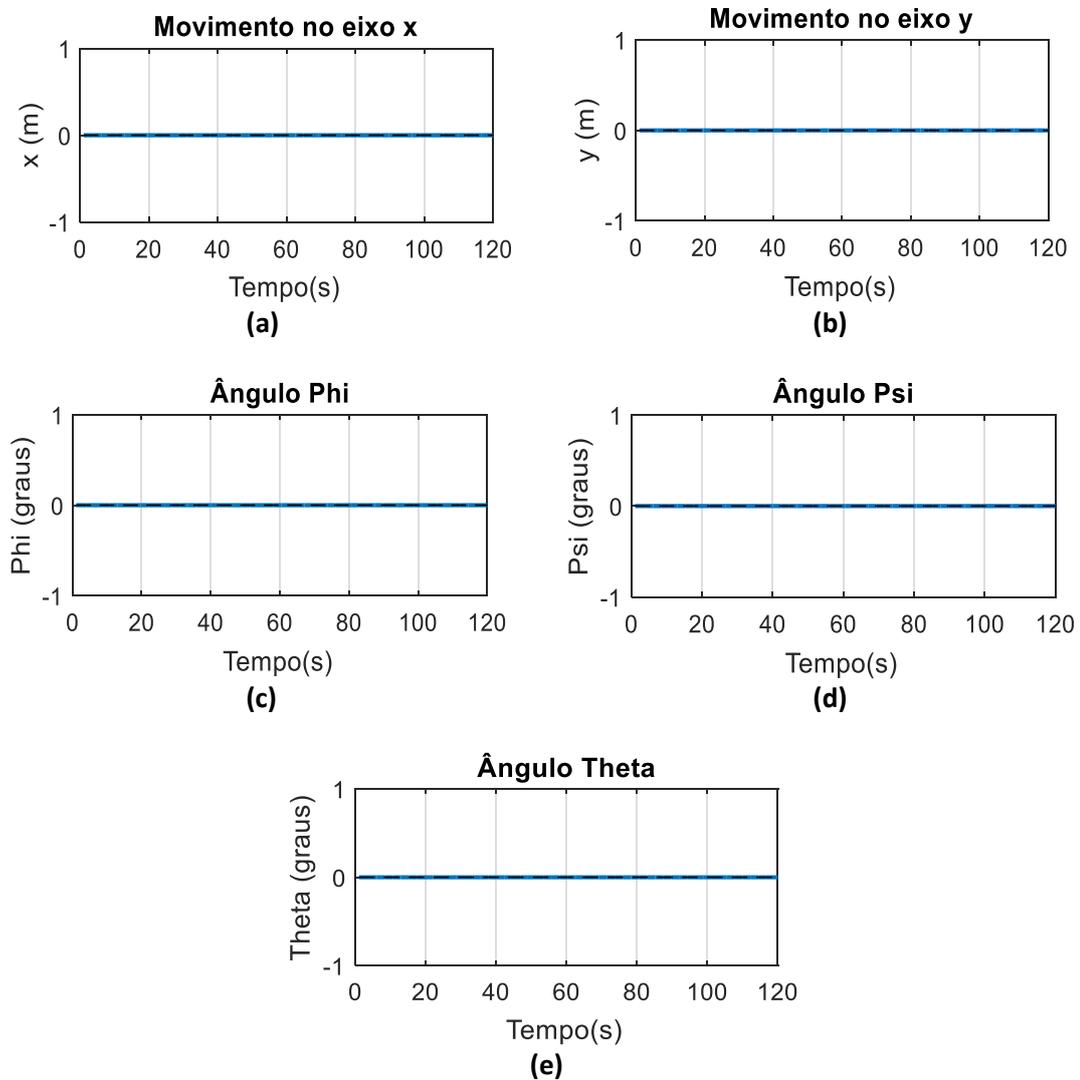
Figura 42 – Resultados simulados do primeiro teste com o controlador *Fuzzy*-PID: (a) Altitude; (b) Entradas de controle indiretas.



**Fonte:** Autoria Própria.

Da mesma forma como foi demonstrado no PID clássico, a Figura 43 mostra os gráfico das demais variáveis analisadas durante a realização da simulação do primeiro teste com o controlador *Fuzzy*-PID. A apresentação ocorre na mesma ordem, onde a Figura 43(a) e 43(b) ilustram o deslocamento do Drone em relação aos eixos  $^g x$  e  $^g y$ , respectivamente, enquanto que as Figura 43(c), 43(d) e 43(e) mostram a evolução temporal dos ângulos  $\phi$  (Phi),  $\theta$  (Psi), e do  $\psi$  (Theta), respectivamente. Sendo assim, é possível perceber que todos os gráficos apresentaram valores constantes em zero, oriundos dos mesmos motivos discutidos anteriormente no controlador PID clássico, contudo vale sempre ressaltar, que tais gráficos se tratam de uma simulação cujo os valores dessas variáveis no tempo são geradas a partir de integrações numéricas decorrentes das equações contidas no bloco do modelo de corpo rígido, apresentada em tópicos anteriores, que apesar da existência de uma matriz de perturbação não foi suficiente para promover um ruído significativo nessas variáveis, porém isso não implica dizer que nos testes reais esse ruído não venha a existir.

Figura 43 – Resultados simulados do primeiro teste com o controlador *Fuzzy*-PID: (a) Movimento no eixo x; (b) Movimento no eixo y; (c) Ângulo Phi; (d) Ângulo Psi; (e) Ângulo Theta.



**Fonte:** Autoria Própria.

➤ Resultados das simulação para o segundo teste com ambos os controladores:

Para o segundo teste foi programado uma missão de voo para o Drone, assim como realizado em Brandão (2013), onde para cada intervalo de tempo de 20s é estabelecido um novo valor para o *set-point*, ou seja, a aeronave irá realizar uma tarefa de posicionamento executando manobras de subida e descida em relação ao eixo  $z$ , conforme pode ser visto na Tabela 4.

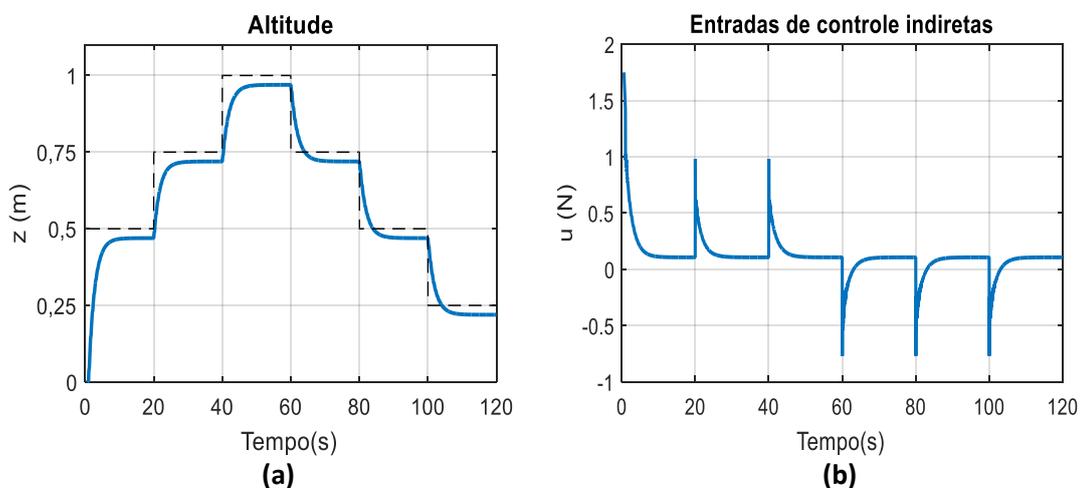
Tabela 4 - Valores desejados da missão de voo para o segundo teste com os controladores

Tempos (s)	0-20	20-40	40-60	60-80	80-100	100-120
Altura z (m)	0.5	0.75	1	0.75	0.5	0.25

**Fonte:** Adaptado de Brandão (2013).

A Figura 44 apresenta os resultados da simulação para o segundo teste com o controlador PID clássico, é possível observar através do gráfico de altitude da Figura 44(a) que os valores desejados dos *set-point* estão representados por linhas tracejadas, enquanto que a linha contínua representa a altura do Drone obtida por integrações numéricas. Assim como no primeiro teste com o controlador PID, o segundo teste também apresentou bons resultados na simulação, tendo uma boa resposta durante as transições de altura, apresentando apenas um pequeno erro de *Offset* (diferença entre a altura desejada e a altura obtida pelo Drone, esse erro será melhor explicado no tópico que trata da comparação entre os dois controladores). A Figura 44(b), como previsto, apresenta três saltos positivo e três negativos em intervalos de 20s que são decorrentes das transições dos movimentos de subida e de descida do Drone, programados pela missão de voo. Uma vez atingido o valor de *set-point*, as forças aplicadas ao Drone têm como objetivo apenas compensar a força gravitacional, de forma a manter a aeronave planando na altura desejada.

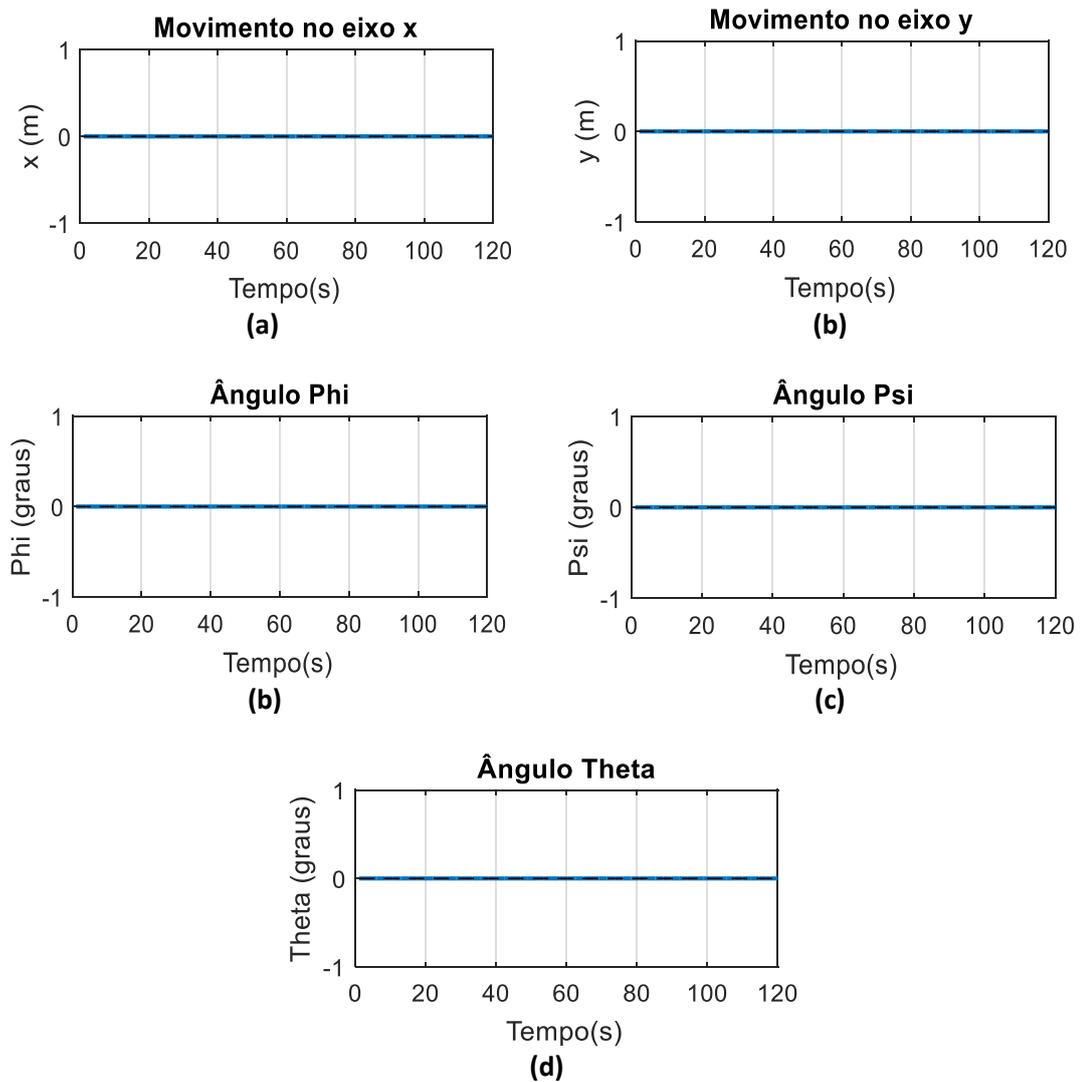
Figura 44 – Resultados simulados do segundo teste com o controlador PID clássico: (a) Altitude; (b) Entradas de controle indiretas.



**Fonte:** Autoria Própria.

Os demais gráficos apresentados na Figura 45, como de se esperar de uma simulação, mantiveram sua posição estável em zero já que o movimento é apenas no eixo z.

Figura 45 – Resultados simulados do segundo teste com o controlador PID clássico: (a) Movimento no eixo x; (b) Movimento no eixo y; (c) Ângulo Phi; (d) Ângulo Psi; (e) Ângulo Theta.

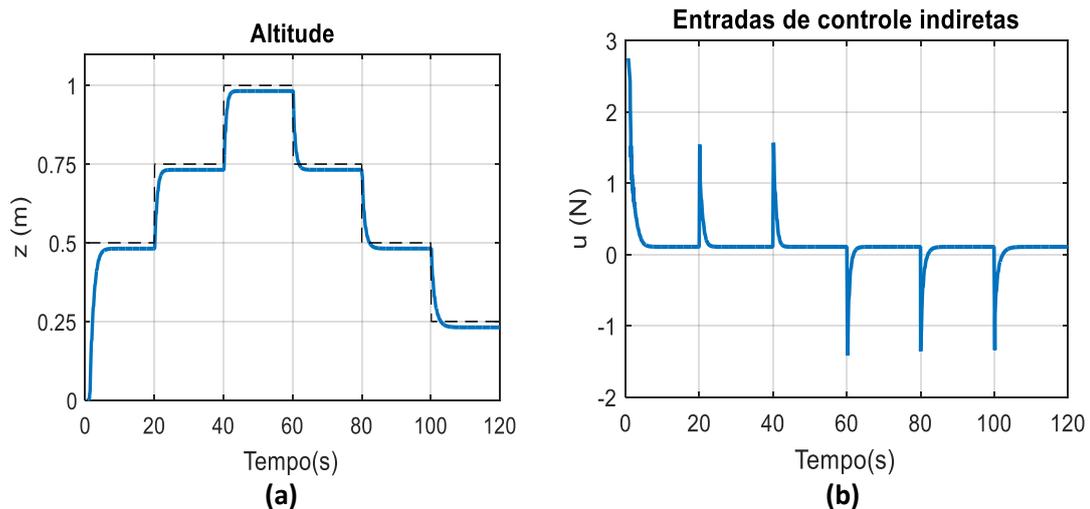


**Fonte:** Autoria Própria.

Aplicando o mesmo plano de voo apresentado na Tabela 4 ao controlador *Fuzzy*-PID é obtido os resultados desse controlador (Figura 46), onde é visualmente notória a melhoria apresentada pela Figura 46(a), com ela é possível ver uma convergência assintótica da altura do Drone em torno dos vários *set-points* estabelecidos pela missão de voo, logo os resultados simulados do controle híbrido mostrou ser bem promissor. A Figura 46(b) apresenta os mesmos picos mostrados pelo PID, esses picos evidencia os instantes em que ocorre o incremento e

decremento de força para as várias mudanças de altura. Devido a permanência inalterada dos valores dos demais gráficos complementares desse teste, optou-se por não ilustrá-los, mais fica a afirmativa de seus valores constantes em zero, sem perturbações.

Figura 46 – Resultados simulados do segundo teste com o controlador *Fuzzy*-PID: (a)Altitude; (b) Entradas de controle indiretas



**Fonte:** Autoria Própria.

#### 4.2. RESULTADOS DOS TESTES PRÁTICOS.

Após os testes computacionais é dado início aos testes práticos, onde é destinado a aplicação dos dois controladores apresentados neste capítulo (o controlador PID e o *Fuzzy*-PID) no sistema de controle do VANT, modelo Ar.Drone Parrot 2.0, afim de se validar ambos as estruturas de controle.

Antes de ilustrar os resultados práticos obtidos neste trabalho é importante frisar novamente que devido ao projeto está restrito somente ao controle de apenas um grau de liberdade, logo todos os movimentos referentes aos demais graus de liberdade foram relaxados. O plano de voo foi delimitado pelo interior de uma sala, afim de evitar perturbações externas, tais como rajadas de vento além de garantir uma margem de segurança (o teto da sala) caso algo de errado durante os testes (falhas ou perda do sinal de controle por exemplo) e o Drone venha a subir indefinidamente. O uso *indoor* do VANT pode limitar sua capacidade de localização através do sistema de posicionamento global (GPS), dessa forma para a detecção da altura foi utilizado o sensor ultrassônico (já incluso no Drone). Outro ponto que irão perceber com os

gráficos que se seguem é que o registro da altura não inicia no zero (como ocorria durante as simulações), esse fato é decorrente da calibragem do Drone que ao ser ligado o mesmo inicia levantando voo e atingindo a altura de cerca 0.75m onde os seus sensores são calibrados, caso contrário o mesmo entra em oscilação e fica buscando uma orientação por meio do GPS, após a calibragem o Drone passa a ser manipulado pelo controlador proposto, dessa forma, para todos os testes práticos com o Drone a altura inicial será contabilizada a partir dos 0.75m, como pode ser visto na Figura 47.

Figura 47 –Fotos do momento de calibragem dos sensores do Drone



**Fonte:** Autoria Própria.

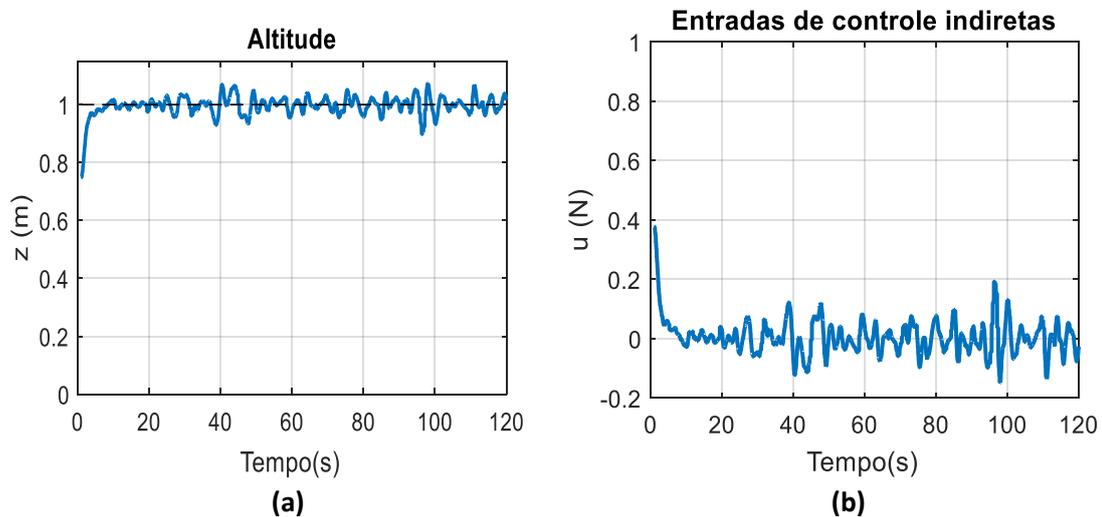
Da mesma forma que na seção anterior, será mostrado os gráficos obtidos para cada teste com o controlador individualmente. É importante frisar que as melhorias apresentadas pelos gráficos que se segue serão deixadas para ser discutidas no próximo tópico.

➤ Resultados experimentais do primeiro teste com ambos os controladores:

Para fins de validação do primeiro teste com controlador PID clássico, foram utilizados os mesmos valores de ganhos atribuídos durante as simulações ( $K_p = 1.5$ ,  $K_i = 1.5$  e  $K_d = 2$ ) e verificado o comportamento do Drone que são mostrados pelos gráfico da Figura 48. A Figura 48(a) mostra a evolução temporal da altitude do Drone no espaço cartesiano, como era previsto existe pequenas oscilações entorno do valor desejado, decorrente das incertezas nas medições de altitude com o sensor ultrassom do Drone, contudo o mesmo permaneceu planando em torno de 1m, como era planejado. As oscilações mostradas pelo gráfico da Figura 48(b) mostram a

tentativa do controlador em manter a estabilidade da altura em torno do *set-point* realizando os incrementos e decrementos de força.

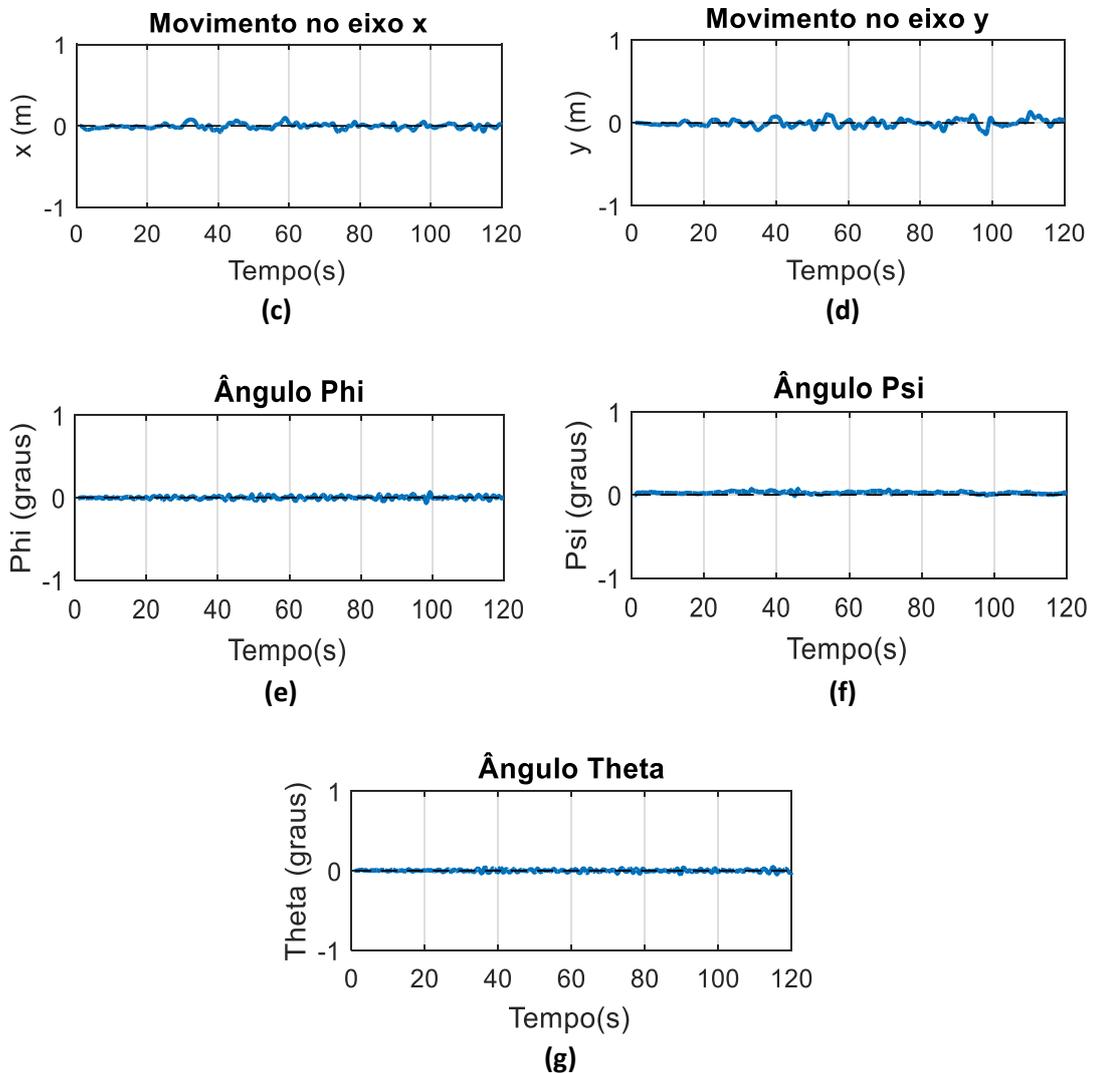
Figura 48 – Resultados experimentais do primeiro teste com o controlador PID clássico:  
(a)Altitude; (b) Entradas de controle indiretas.



**Fonte:** Autoria Própria.

A Figura 49 mostra as demais variáveis analisadas durante o teste, no que diz respeito aos deslocamentos laterais indesejados mostrados pelos gráficos das 49(a) e 49(b) são consequências, segundo Brandão (2013), dos efeitos de *drifting* dos sensores inerciais, que afeta diretamente no sistema de posicionamento do veículo, além do forte acoplamento dinâmico presente nesse tipo de aeronave, que aplica a condição de que todos os motores devem estar com a mesma rotação (caso contrário qualquer alteração na velocidade de um dos rotores pode gerar um movimento indesejado), dessa forma, como não foi aplicado nenhum controle sobre esses eixos, qualquer mudança na rotação dos motores pode provocar um torque maior em um dado eixo, fazendo o Drone realizar movimentos indesejados. O mesmo pode ser observado pelos gráficos das Figura 55(c), 49(d) e 49(e). É importante sempre frisar que o controle que está sendo feito é apenas referente ao eixo z, logo, mesmo que o Drone venha a realizar um deslocamento indesejado no eixo x e/ou y, desde que ele mantenha a altura estabelecida pelo *set-point*, esse deslocamento não possui influência direta na eficiência do controlador proposto, dessa forma os gráficos da Figura 49, são apenas complementares e poderão ser utilizados para critério de comparação em trabalhos futuros.

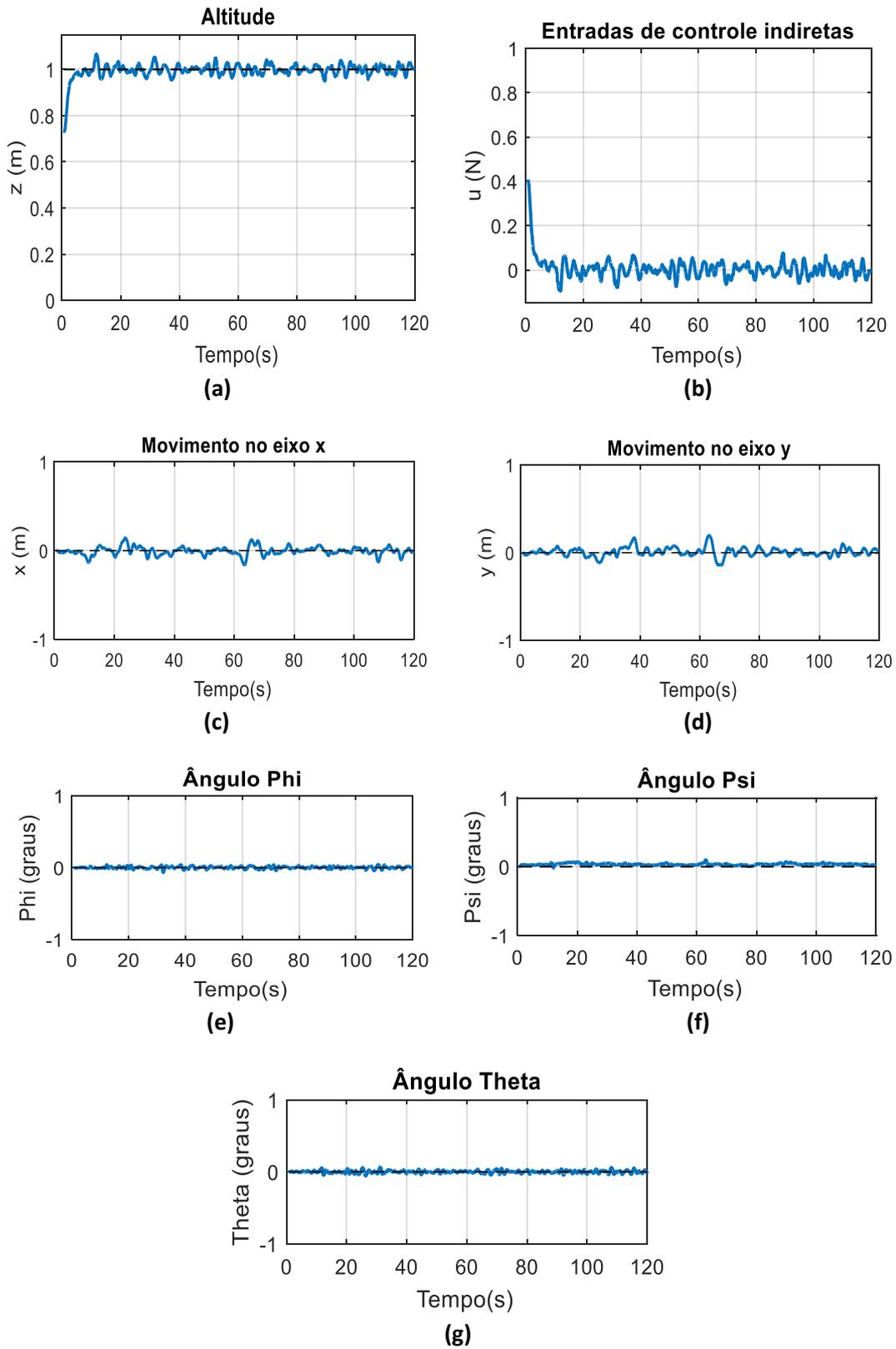
Figura 49 – Resultados experimentais do primeiro teste com o controlador PID clássico: (a) Movimento no eixo x; (b) Movimento no eixo y; (c) Ângulo Phi; (d) Ângulo Psi; (e) Ângulo Theta.



**Fonte:** Autoria Própria.

A Figura 50 mostra todas as variáveis analisadas referentes ao resultado experimental do primeiro teste com o controlador híbrido *Fuzzy*-PID. Na análise do gráfico referente a Figura 50(a), o mesmo também apresentou várias oscilações em torno do valor do *set-point*, essas oscilações ocorrem pelos mesmos motivos já explanados anteriormente sendo o erro de medição do sensor ultrassônico o principal deles. No entanto essas variações já eram esperadas, contudo as mesmas apresentaram baixa magnitude o que está em conformidade com os limites do projeto. A argumentação que respalda as vantagens apresentadas por esses gráficos serão melhor detalhados no tópico referente a comparação entre os controladores, dessa forma a análise técnica sobre eles serão deixadas para o referido tópico em questão.

Figura 50 – Resultados experimentais do primeiro teste com o controlador *Fuzzy*-PID:  
 (a)Altitude; (b) Entradas de controle indiretas.

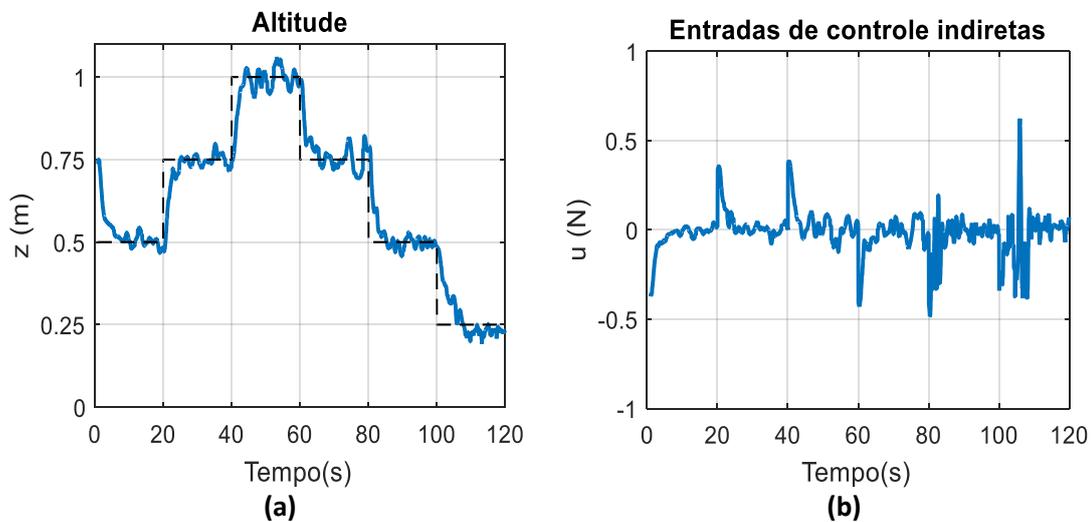


Fonte: Autoria Própria.

➤ Resultados experimentais do segundo teste com ambos os controladores:

A Figura 51 mostra resultados práticos do segundo teste com o controlador PID puro. É possível perceber através da Figura 51(a) que assim como no resultado do primeiro teste o gráfico de resposta da altitude o mesmo mostra pequenas oscilações em torno dos vários *set-point* estabelecidos pela missão de voo, como era de se esperar em virtude de ser um resultado prático, contudo é possível perceber também que o Drone responde aos comandos do controlador de forma eficiente, executando todas as manobras de subida e descidas estabelecidas pelo plano de voo, ao qual era o objetivo desse segundo teste. Assim como na simulação, o gráfico da Figura 51(b) mostra quatro sobre saltos negativos e dois positivos, o primeiro salto negativo deve-se ao fato de que o Drone inicia a altura em 0.75m e passa para 0.2m (*set-point* inicial da missão de voo), logo ele deve realizar, como primeira ação, o movimento de descida, para depois retomar a sequência de voo planejada.

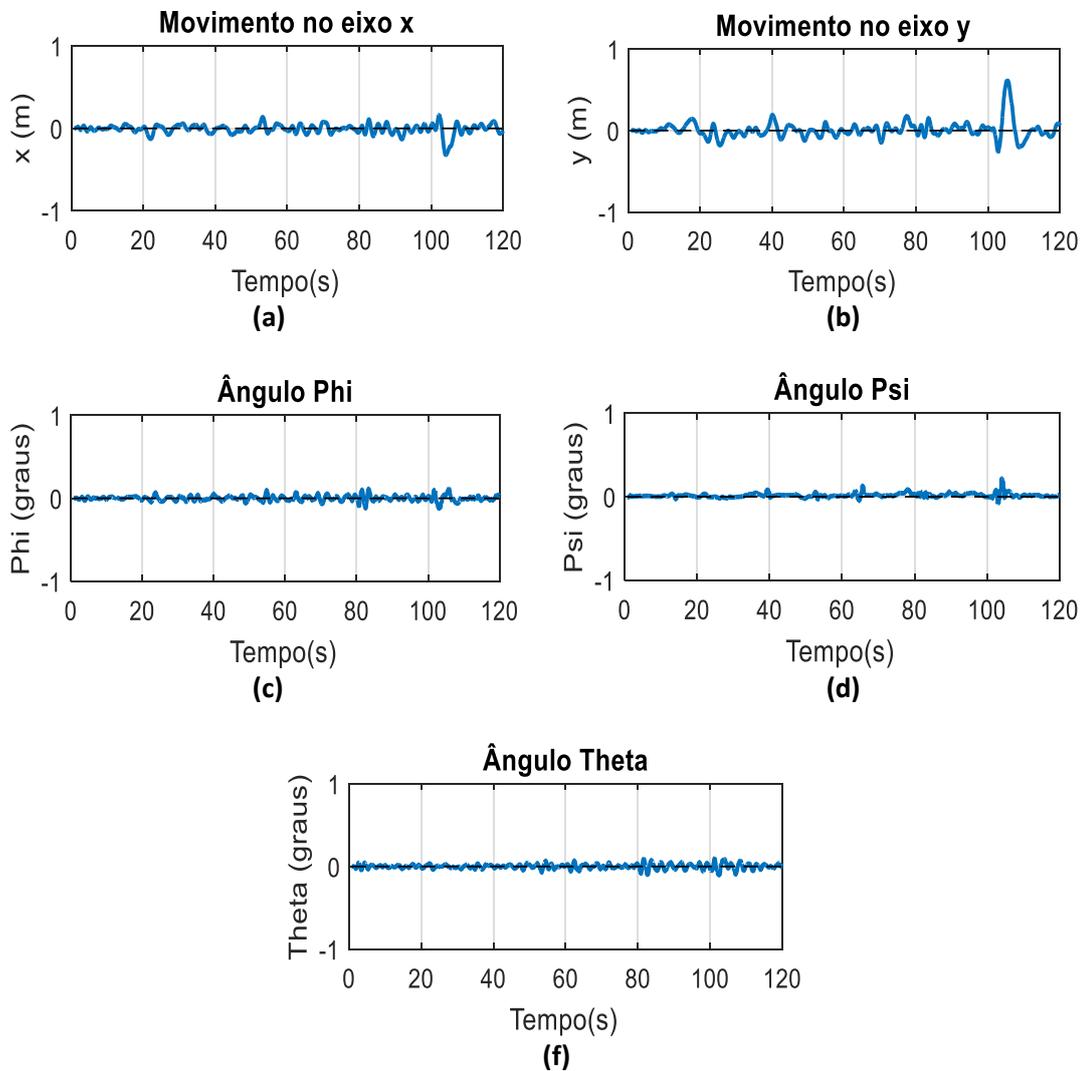
Figura 51– Resultados experimentais do segundo teste com o controlador PID clássico:  
(a)Altitude; (b) Entradas de controle indiretas.



**Fonte:** Autoria Própria.

Os demais gráficos mostrados na Figura 58 apresentaram oscilações maiores, consequência das várias mudanças de altura que geram perturbações no sistema, dessa forma, como não há de fato um controle sob essas variáveis as mesma são influenciadas pela correlação entre os eixo, característica do seu sistema subatuado, como já discutido anteriormente.

Figura 52 – Resultados experimentais do segundo teste com o controlador PID clássico: (a) Movimento no eixo x; (b) Movimento no eixo y; (c) Ângulo Phi; (d) Ângulo Psi; (e) Ângulo Theta.

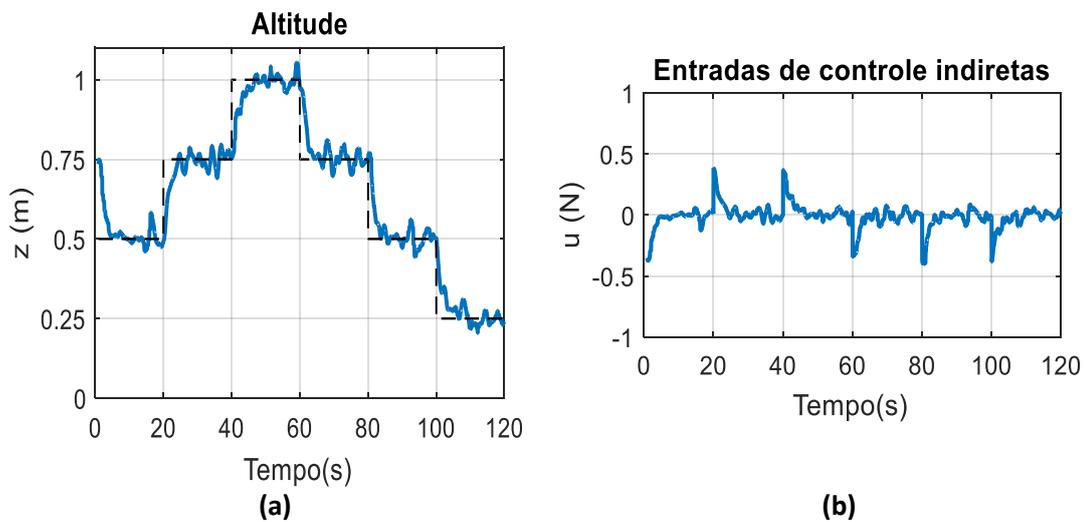


Fonte: Autoria Própria.

Como se segue, a Figura 53 mostra os resultados práticos do segundo teste com o controlador híbrido *Fuzzy*-PID. O gráfico da Figura 53(a) mostra a resposta do controlador mediante a consecutivas mudanças de altura, com ela é possível notar que o mesmo se assemelha a resposta apresentada pelo controlador PID, em termos práticos, ambas as estruturas de controle executaram de maneira satisfatória o objetivo do segundo teste, a acentuação das oscilações são naturais, elas são intensificadas devido as perturbações geradas pelas mudanças de altura, além disso o tempo de repouso de cada salto (os 20s estabelecidos pelo plano de voo) é relativamente curto sendo insuficiente para que ambos os controladores consigam estabilizá-los. Além disso, o controlador *Fuzzy*-PID apresentou uma desvantagem com relação ao PID,

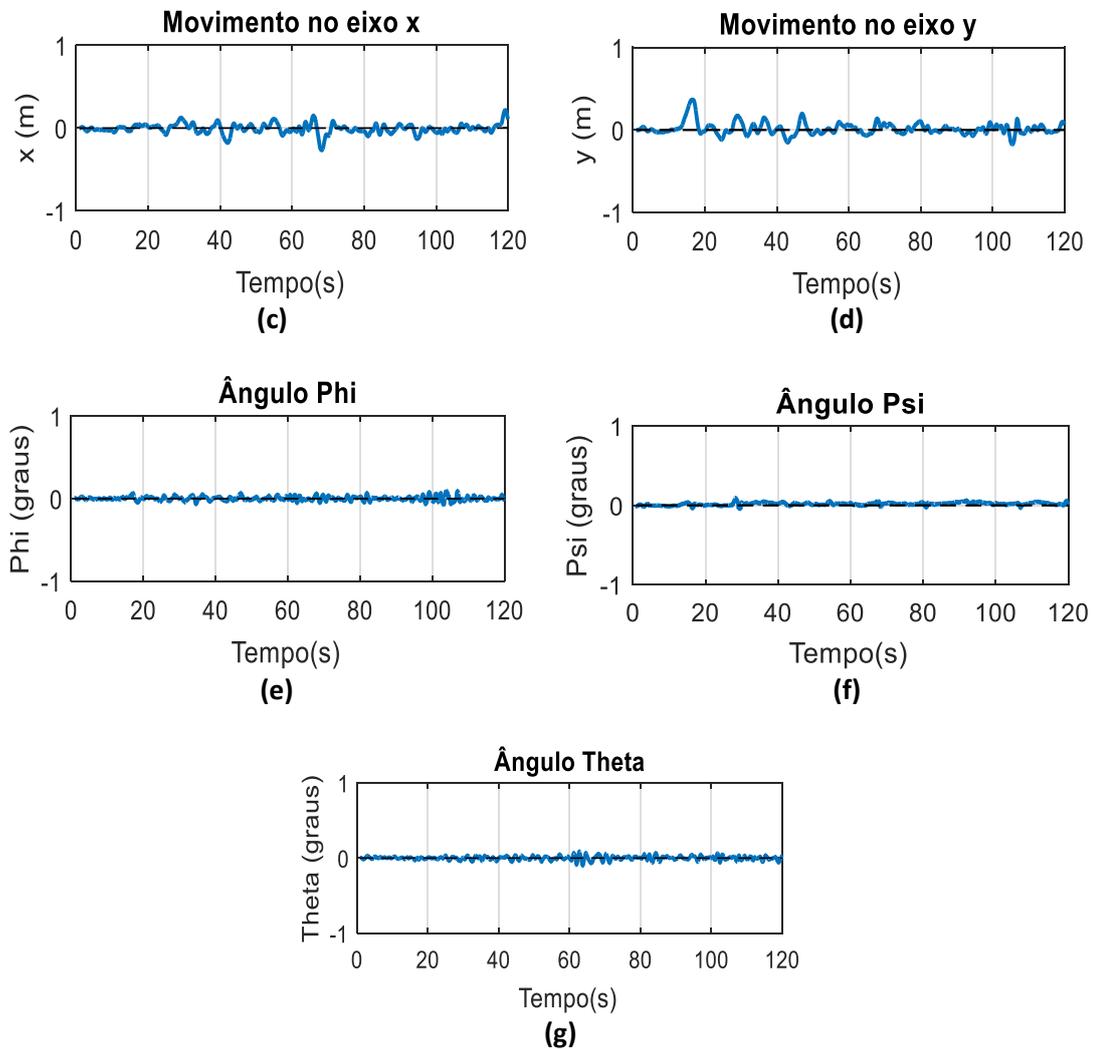
notada durante a realização desse segundo teste, que foi o incremento de ganhos oscilantes, ao qual devido ao tempo curto o mesmo não consegue ajustar o controle antes da próxima mudança de *set-point*, apresentando em alguns pontos, oscilações mais acentuadas. Esses incrementos desordenados podem ser vistos pelo gráfico da Figura 53(b) onde mostra os picos da inserção e redução de forças.

Figura 53 – Resultados experimentais do segundo teste com o controlador *Fuzzy*-PID:  
(a)Altitude; (b) Entradas de controle indiretas.



As demais variáveis analisadas, mostradas pela Figura 54, apresentaram os mesmo resultados discutidos no controlado PID clássico, além disso é importante ressaltar que devido ao espaço confinado ao qual os testes práticos foram realizados (interior de uma sala fechada com objetos), tais movimentos indesejados dificultaram a realização desses testes práticos, sendo os mesmo interrompidos diversas vezes, afim de se evitar uma colisão do Drone com alguns objetos no interior da sala.

Figura 54 – Resultados experimentais do segundo teste com o controlador *Fuzzy*-PID: (a) Movimento no eixo x; (b) Movimento no eixo y; (c) Ângulo Phi; (d) Ângulo Psi; (e) Ângulo Theta.



**Fonte:** Autoria Própria.

#### 4.3. COMPARAÇÃO COM CONTROLADOR CLÁSSICO PID

Este tópico será destinado a comparação entre as duas estruturas de controle realizadas nesta dissertação, a primeira se tratando de um controlador clássico proporcional, integrativo e derivativo o PID e a segunda trata-se de uma implementação desse controlador com a inserção de um controle *Fuzzy* supervisor dando origem a uma estrutura de controle híbrido *Fuzzy*-PID, sendo esta é a principal proposta desse trabalho.

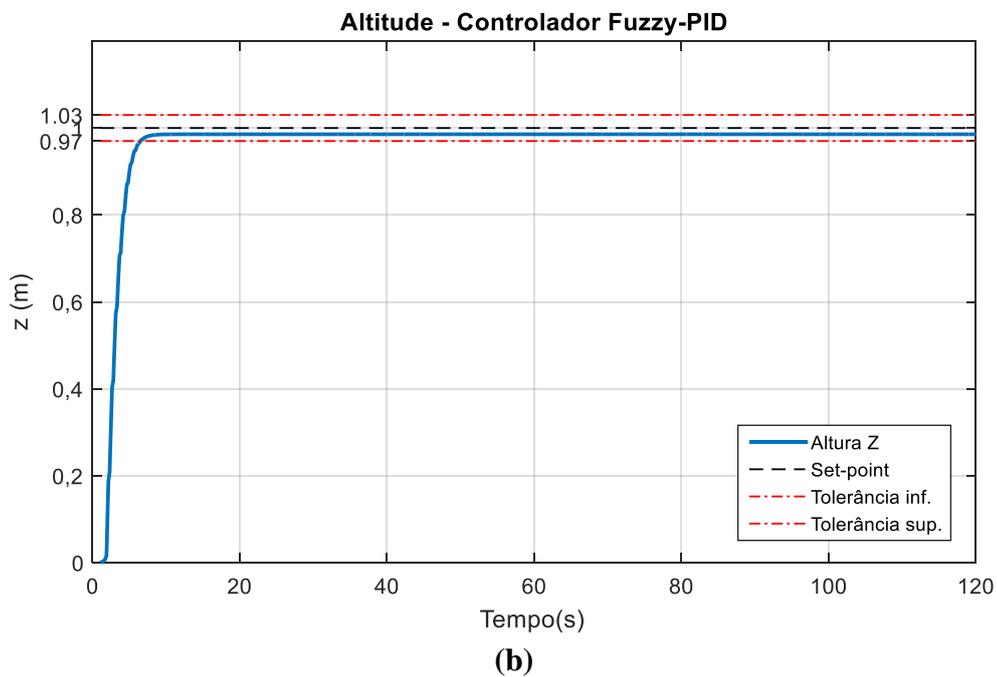
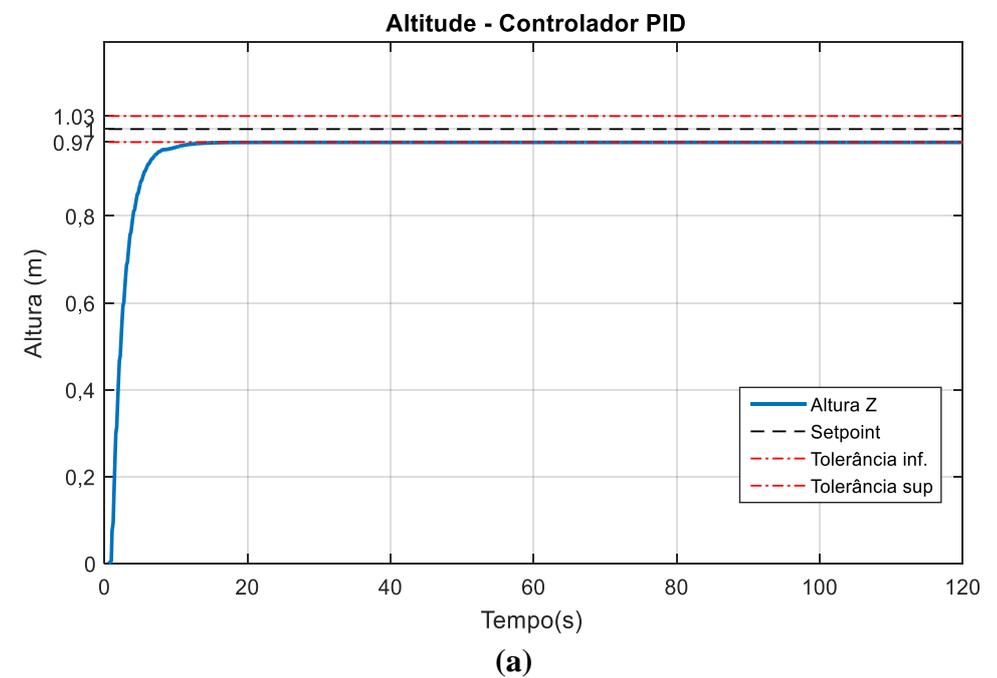
Focando no objetivo dos controladores, será feito uma análise apenas sob os gráficos de altitude apresentados ao longo desse capítulo. Sobre elas serão realizadas as comparação dos

resultados simulados e dos experimentais. Vale ainda ressaltar que para melhor análise dos gráficos os mesmo poderão ser ampliados e customizados para que se possa ver as melhorias e defeitos de cada controlador.

Dando início as análises, na Figura 55 mostra a comparação entre os resultados simulados do primeiro teste com controlador PID e o *Fuzzy*-PID. Nesse teste foi dado uma margem de tolerância de  $\pm 0.03\text{m}$  que é possível ver pela Figura 55(a) onde o controlador PID atingem o limite inferior da tolerância do projeto estando o mesmo coerente com o objetivo do controlador, além disso o mesmo apresentou uma subida rápida (aproximadamente 9s) sem oscilações, ao qual é mantido uma postura estável até o fim da simulação. Contudo, quando comparado a Figura 55(b) é possível ver que há uma pequena melhoria proporcionada pelo controlador *Fuzzy*-PID, cuja resposta se aproxima mais do *set-point*, onde o erro de *Offset* diminui de 0.03m para 0.008m e o tempo de subida reduzido para 8s, esse efeito é decorrente do fato de que no controlador PID o ajuste de ganho é realizado manualmente enquanto que o controlador *Fuzzy*-PID possui o *Fuzzy* supervisor se encarrega de fazer esse ajustes finos durante a simulação levando a reposta mais próxima do resultado desejado.

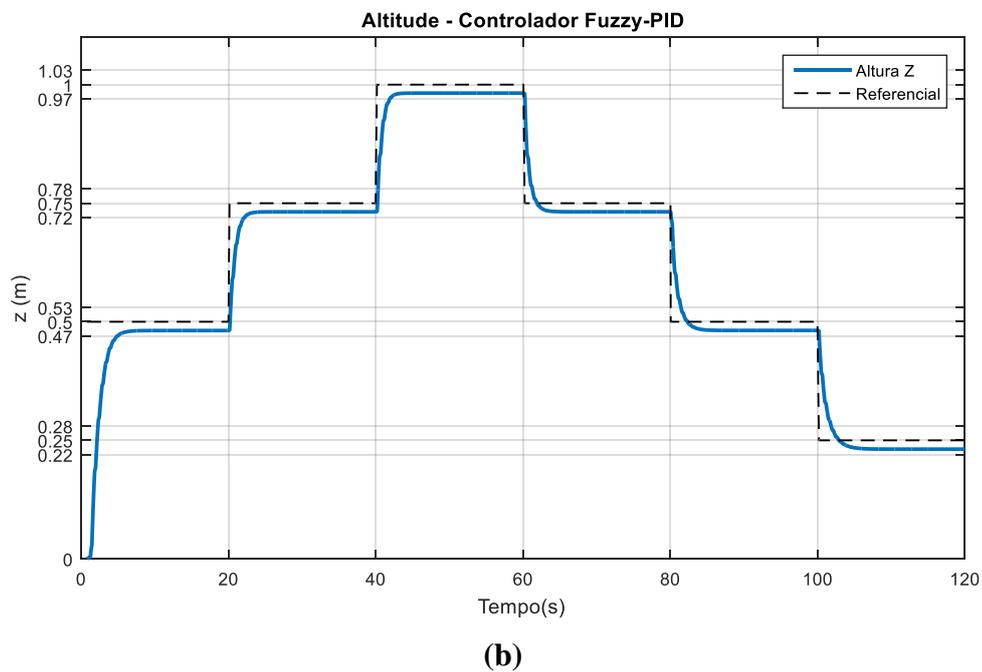
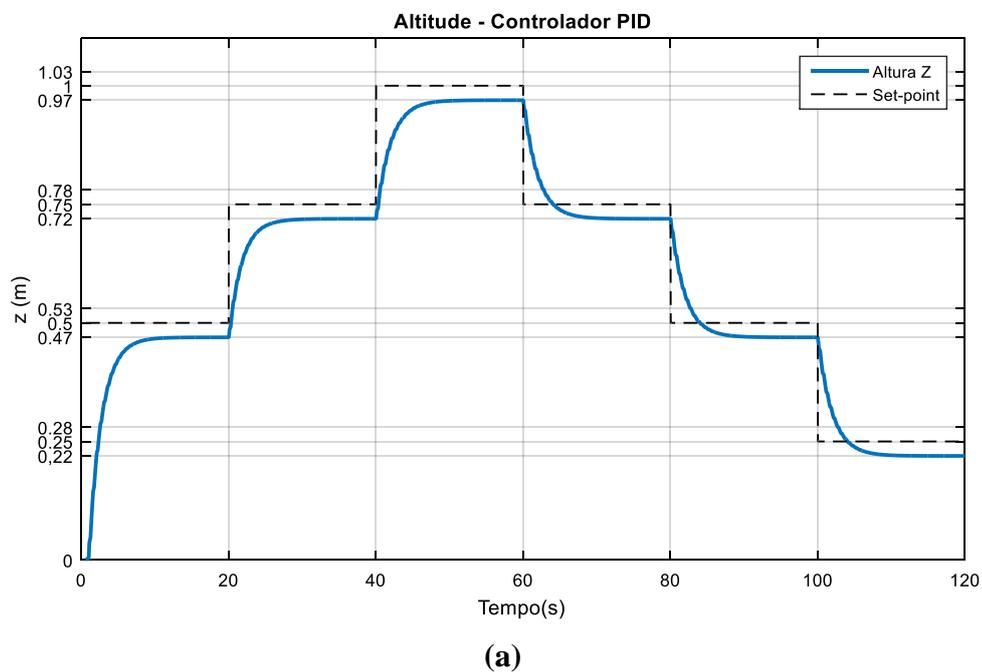
Através da Figura 56 é possível fazer uma comparação entre os resultados simulados do segundo teste com controlador PID e o *Fuzzy*-PID, nela o possível notar que ambas as simulações apresentaram bons resultados sendo que pelo gráfico Figura 56(a) é possível ver que o controlador PID atinge a tolerância pré-estabelecido no projeto com uma margem de erro do *set-point* de 0.03m abaixo do desejado, enquanto que pelo gráfico da Figura 56(b) é possível ver que com a implementação do *Fuzzy*-PID, o controlador reajusta as variáveis de entrada fazendo uma melhor aproximação com valor do referencial, levando ao erro de aproximadamente 0.01m abaixo do desejado, ou seja, uma redução de 2mm, apesar de pequena essa melhoria é significativa para o desenvolvimento do projeto, além disso é possível notar uma melhoria na velocidade de resposta do controlador *Fuzzy*-PID, que responde bem a mudança consecutivas do *Set-point*.

Figura 55 – Comparação entre os resultados simulados do primeiro teste com controlador PID e o *Fuzzy*-PID.



Fonte: Autoria Própria.

Figura 56 – Comparação entre os resultados simulados do segundo teste com controlador PID e o *Fuzzy*-PID.

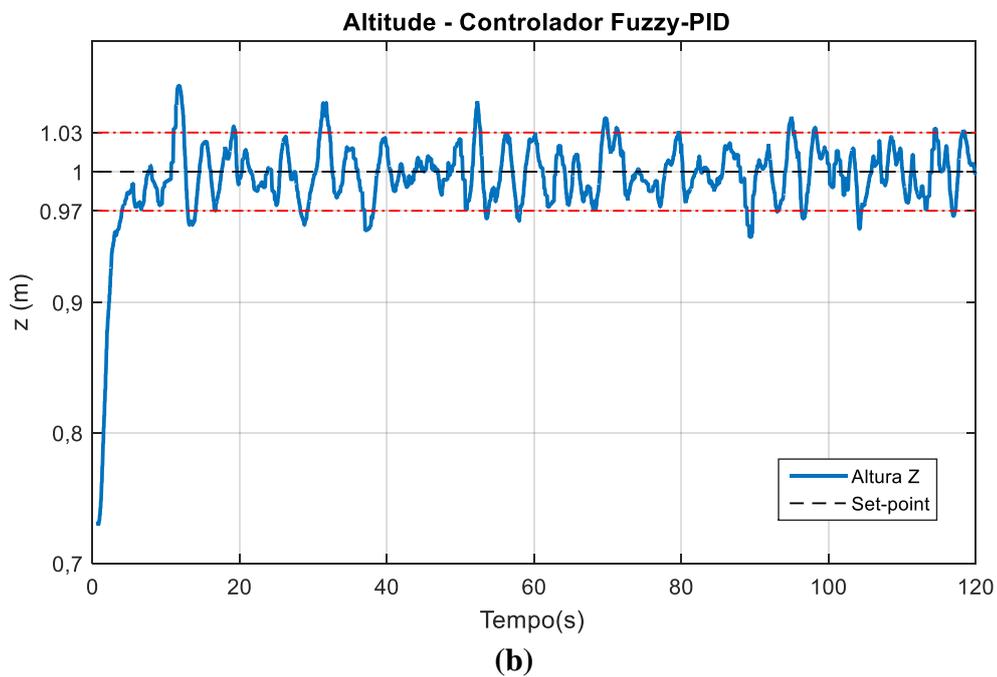
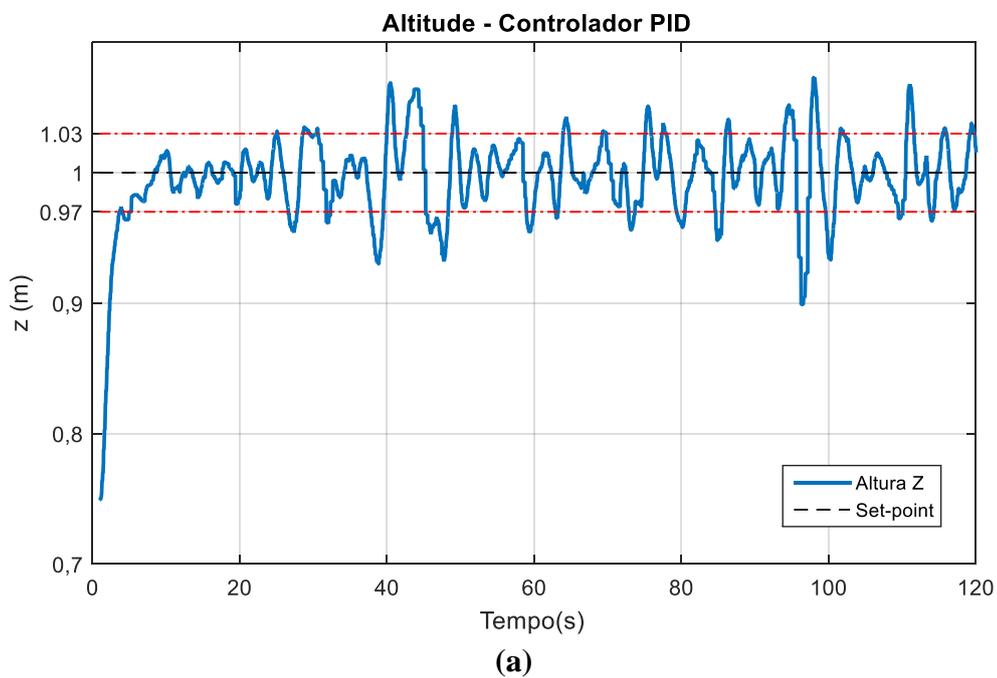


**Fonte:** Autoria Própria.

Dando continuidade à análise dos controladores, agora será feita a comparação entre os resultados obtidos experimentalmente com o Drone, começando pelos gráficos da Figura 57 onde é possível ver os resultados práticos do primeiro teste de ambos os controladores aplicados ao Drone real. Pelo gráfico da Figura 57(a) pode-se perceber que o controlador PID consegue elevar o Drone até a altura desejada e tenta manter a postura estável dentro dos limites estabelecidos pelo projeto (linhas tracejadas de vermelho), contudo apresenta alguns sobre saltos acentuados que ultrapassam as margens estabelecidas, quando comparado ao gráfico do controlado *Fuzzy*-PID denotados pela Figura 57(b) é possível ver uma melhoria na estabilidade da altura do Drone, tendo um melhor controle em torno das margens de tolerância, provenientes da introdução do controlador *Fuzzy* supervisor, que reajusta os valores do ganho conforme o surgimento de perturbações, ou seja, ao perceber um sobre salto ele imediatamente ajusta os ganhos afim de prevenir um sobressalto posterior, o que não ocorre com o PID já que o mesmo só permite o reajuste dos ganhos após realizar o clique completo da compilação.

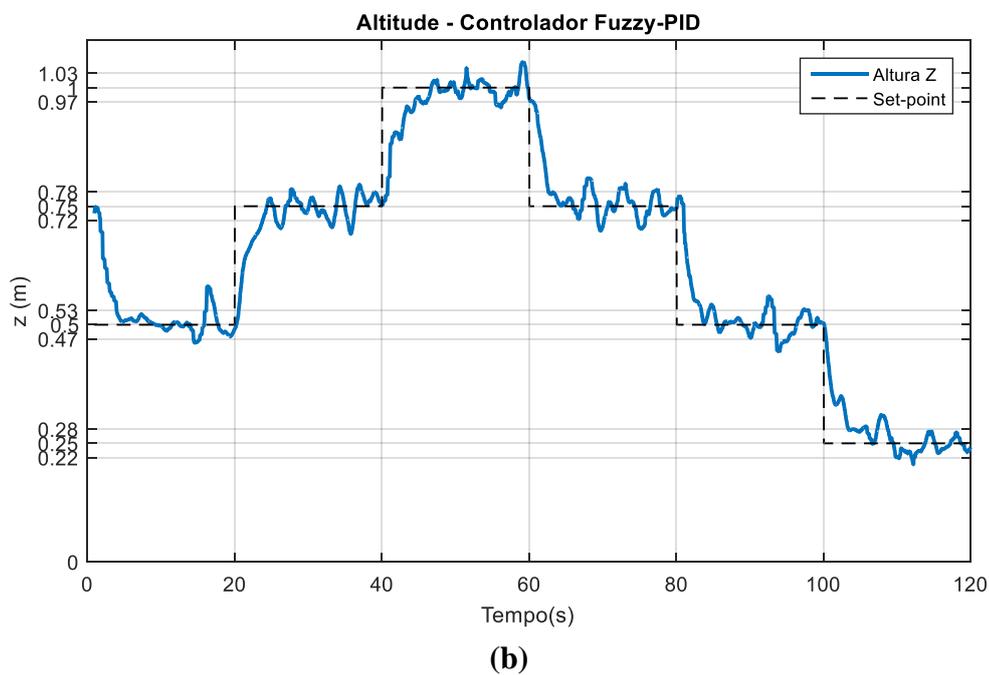
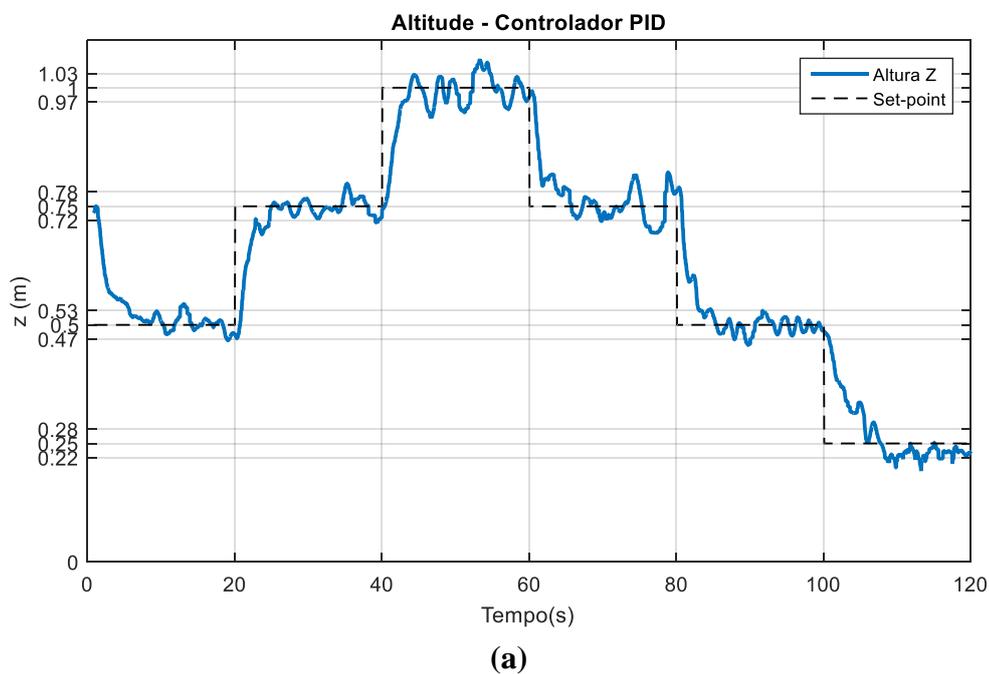
Através da Figura 58 é possível fazer uma melhor análises entres os resultados obtidos pelo segundo teste com o controlador PID e o *Fuzzy*-PID, nela é possível notar que ambos os controladores conseguem realizar as manobras de subida e descida do Drone conforme descrito pelo plano de voo, apesar de ambos os controladores apresentarem oscilações durante as transições de altura é possível notar uma pequena melhoria na estabilidade por parte do controlador *Fuzzy*-PID no que se refere a quantidade de sobressaltos que ultrapassam as margens de tolerância de  $\pm 0.03\text{m}$ , podendo ser visto pelo gráfico da Figura 58(b).

Figura 57 – Comparação entre os resultados práticos do primeiro teste com controlador PID e o *Fuzzy*-PID.



**Fonte:** Autoria Própria.

Figura 58 – Comparação entre os resultados práticos do segundo teste com controlador PID e o *Fuzzy*-PID.



**Fonte:** Autoria Própria.

## 5. CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi apresentado o projeto de desenvolvimento de uma estrutura híbrida envolvendo a junção de duas estruturas de controle, o controlador PID (*Proporcional-Integral-Derivativo*) e o controlador *Fuzzy* supervisor. Esse controlador tem como objetivo o controle de estabilidade de altura de um Drone do tipo quadrimotor, mais especificamente o modelo Ar.Drone Porrot 2.0. Conforme o decorrer do trabalho foi apresentado o processo de elaboração do controlador proposto incluído os resultados simulados e a validação por meio da comparação com resultados práticos, também foi possível obter uma análise das melhorias apresentadas pela junção híbrida desses controladores, fazendo uma comparação entre o controlador PID puro com o controlador *Fuzzy*-PID proposto.

No geral as duas estruturas de controle (PID clássico e *Fuzzy*-PID) apresentaram resultados satisfatórios tanto no critério de estabilidade de altura para um único valor de *set-point* ao longo de um período de 120s (apresentado pelo primeiro teste), como também pela resposta para mudanças consecutivas de altura, proposta pelo segundo teste. O controlador PID apresentou resultados satisfatórios, a grande dificuldade desse controlador está contido no ajuste adequado dos ganhos, que para este caso, devido à complexidade de se obter o modelo matemático da planta, não foi possível aplicar nenhum método matemático ou gráfico para se obter os parâmetro de ajuste dos valores dos ganhos (como o caso do método Ziegler-Nichols, que proporciona uma orientação para melhorar a sintonia dos ganhos) dessa forma foi necessário realizar a sintonia manual através de tentativa e erro o que denotou uma quantidade de tempo considerável para o desenvolvimento dos controladores, visto que o controlador PID puro não permite a sintonia durante a simulação, ou seja, é necessário que o controlador realize uma simulação para cada ajuste de ganho diferente. Já com a implantação do *Fuzzy* supervisor em conjunto com o controlador PID é possível construir uma estrutura híbrida capaz de realizar ajustes automáticos dos ganhos com base em um conjunto de regras pré estabelecidas pelo programador. No projeto proposto nesta dissertação o controlador *Fuzzy*-PID apresentou resultados satisfatórios tanto para o primeiro como para o segundo teste aplicado, além disso ele apresentou algumas melhorias de estabilidade evidenciadas pelos resultados simulados e os resultados práticos com o Drone.

A grande vantagem do controlador *Fuzzy*-PID encontrada neste trabalho devesse ao fato do auto ajuste dos ganhos, concebidos pelo *Fuzzy* supervisor, que monitora o erro e a variação da variação do erro, dando ao controlador a capacidade de antecipar um erro futuro e corrigi-lo antes que aconteça, isso tudo durante a simulação, ou seja, não é necessário esperar que a

simulação acabe para poder reajustar os ganhos, pois o *Fuzzy* permite fazer isso durante a simulação. Caso o conjunto de regras contemplem todas possibilidades de ajuste o controle tornasse ainda mais completo e mais eficaz estando preparado para conter qualquer perturbação que possa surgir. Com base nessa vantagem é possível destacar o maior problema encontrado neste trabalho durante o desenvolvimento do controle *Fuzzy*, que é a necessidade do conhecimento heurístico de um especialista no comportamento do sistema. Como solução foi realizada diversas simulações tanto experimentais como computacionais, afim de se conhecer a dinâmica e o comportamento dessa aeronave testando várias possibilidades de ajustes e regras até se chegar no controle apresentado nesta dissertação. O controle ainda não está perfeito, no entanto na medida do tempo disponível o mesmo apresentou resultados satisfatórios para o intuito do projeto, sendo deixado as melhorias a serem feitas para trabalhos posteriores.

Com respeito a execução dos testes experimentais, foram evidenciadas várias dificuldades a mais pertinente está refletida no fato do pouco espaço disponível para execução dos testes, sendo estes executados nas imediações de uma sala de estudo repleta de móveis, aumentando consideravelmente o tempo de projeto, já que o Drone precisa de espaço para executar suas manobras de voo, e a aplicação desses testes em uma área externa iria incrementar várias outras intempéries que só aumentariam o tempo de desenvolvimento do projeto, tais com rajada de ventos, possíveis acidentes (materiais ou/e físicos) caso o mesmo venha a perder o controle e colida com alguém ou alguma estrutura (fios elétricos, árvores, telhados ou pessoas) dentre outros.

No que se refere ao material utilizado, o Drone disponibilizado pela instituição foi o modelo Ar.Drone Porrot 2.0, o mesmo apresentava já algumas avarias de uso, como por exemplo uma das engrenagens que faz a transmissão de força para as hélices estava com dificuldades de giro, sendo necessário o uso de lubrificante. Outro ponto ainda com respeito ao Drone se deve ao fato de que o mesmo dispunha de apenas uma bateria, que devido ao tempo de uso já estava no seu limiar de carga, reduzindo consideravelmente o tempo de voo dos testes práticos. Contudo o Drone foi bastante útil já que o mesmo é compatível com a plataforma AuRoRA que foi a peça chave que possibilitou o desenvolvimento e os testes com controlador proposto.

Apesar de sua estrutura simples foi requerido um custo de tempo considerável para compreensão do funcionamento da plataforma AuRaRA de forma a permitir a inserção de uma nova estrutura de controle além de conseguir extrair os gráficos para validação dos resultados. A mesma foi disponibilizada para uso dessa distração e já contemplava todos os preceitos matemáticos da dinâmica do VANT (apresentados ao longo desta dissertação) assim como os

parâmetros do Drone utilizado (para mais detalhes sobre esses valores consulte Brandão, 2013), permitindo a inserção apenas de uma nova estrutura de controle. A plataforma é toda desenvolvida em algoritmo do Matlab o que obriga a construção do controlador na mesma linguagem de programação, esse script não é disponibilizado pois se pretende dar continuidade ao projeto, a vantagem de se usar o Matlab é que o mesmo dispõe de uma *toolbox* específico para construção do *Fuzzy* supervisor, isso facilitou a implantação desse projeto.

O uso de um simulador foi essencial para integridade da estrutura do Drone, pois a atuação incorreta do controle ou até mesmo uma falha na troca de informação com os sensores da aeronave pode comprometer todo o projeto (sabendo que o mesmo dispunha desse único Drone).

A continuação deste trabalho poderá ser dado com inclusão do restante dos controladores para os demais graus de liberdade, transformando esse projeto de controle de altitude em um projeto de controle de atitude. Com isso é possível fazer um comparativo com os gráficos disponíveis nesta dissertação referentes aos eixos  $^g x$ ,  $^g y$ ,  $^g z$ , além dos ângulos  $\phi$  (Phi),  $\theta$  (Psi) e do  $\psi$  (Theta). Ainda como trabalhos futuros pede-se realizar a aplicação dessa estrutura de controle em tarefa mais complexa como, missões de reconhecimento de terreno, fazendo o Drone voar com controlador ao redor de uma aérea, ou até mesmo simular um missão de vigília que a realização de um voo vertical em conjunto com uma variação consecutiva no ângulo de guinda do Drone, já que o mesmo é munido de uma câmera frontal que através desse movimento pode contemplar uma área maior de vigília.

## 6. REFERÊNCIAS

ABBASI, E.; MAHJOOB, M. Controlling of Quadrotor UAV Using a Fuzzy System for Tuning the PID Gains in Hovering Mode. **Center for Mechatronics and Automation, School of Mechanical Engineering College of engineering, University of Tehran Tehran, Iran, 2013.**

ALMEIDA, J. F., SOBRINHO, C. L.S. S., **Influência do Posicionamento de uma Estrutura PBG sobre a Largura de Banda de uma Antena de Microfita**, IEEE Latin America Transactions, Vol. 2, No. 1, March, 2004.

BENIGNO, T. C. P., “**Modelagem matemática e controle de altitude e posição do quadrotor**”. Dissertação de mestrado, UFERSA, Mossoró, RN. 2015.

BOUABDALLAH, S. “**Desing and control of quadrotors with application to autonomous flying**”. Tese de doutorado, Échole Polytechnique Fédérale de Lausanne, 2007.

BOUABDALLAH, S.; MURRIERI, P.; SIEGWART, R. “**Design and Control of an Indoor Micro Quadrotor**”. IEEE International Conference on Robotics and Automation, New Orleans, p. 4393 – 4398, 2004.

BOUABDALLAH, S.; NOTH, A.; SIEGWART, R. “**Pid vs lq control techniques applied to an indoor micro quadrotor**”. IEEE. Intelligent Robots and Systems, Proceedings IEEE/RSJ International Conference on, v. 3, p. 2451-2456, 2004.

BRANDÃO, A. S. et al. **Decentralized control of leader-follower formations of mobile robots with obstacle avoidance**. In: Proceedings of the 5th IEEE International Conference on Mechatronics. Málaga, Spain: IEEE, 2009.

BRANDÃO, A. S. et al. “**Modelagem e controle não linear subatuado de um quad-rotor: Parte 1**”. In: Anais do XIX Congresso Brasileiro de Automática, Campina Grande, PB: CBA, 2012a. p. 449-454.

BRANDÃO, A. S. et al. “**Modelagem e controle não linear subatuado de um quad-rotor: Parte 2**”. In: Anais do XIX Congresso Brasileiro de Automática, Campina Grande, PB: CBA, 2012b. p. 3102-3109.

BRANDÃO, A. S. et al. “**Projeto de controladores não lineares para voo autônomo de veículos aéreos de pás rotativas**”. Tese de Doutorado em Engenharia Elétrica, Universidade federal do Espírito Santo, UFES, Brasil. p. 01-231, 2013.

CAMPOS, C. M. M.; TEIXEIRA, H. C. G. **Controles Típicos de equipamentos processos industriais**. 2. ed. São Paulo: Blucher, 2010.

CASTILLO, P.; LOZANO, R.; DZUL, A. **Stabilization of a mini rotorcraft with four rotors: experimental implementation of linear and nonlinear control laws**. IEEE ControlSystem Magazine, v. 25, n. 6, p. 45–55, 2005.

DORF, R. C.; BISHOP, R. H. “**Sistemas de controle modernos**”. Tradução de Jackson Paul Matsuura. 11ª. ed. Rio de Janeiro: LTC Editora, 2011.

FRANCELIN, R. A. R. “**Sistemas Inteligentes**”. Disponível em: <<http://wiki.icmc.usp.br/images/b/b6/Fuzzy.pdf>>. Acessado em 23 de junho de 2016.

GUIMARÃES, J. P F. **Controle de Atitude e Altitude Para Um Veículo Aéreo Não Tripulado Do Tipo Quadrirotor**. Dissertação de Mestrado em Engenharia Elétrica e de Computação. Universidade Federal do Rio Grande do Norte, Natal, 2012.

GUIMARÃES, J. P F.; SANCA, A., DE DEUS, M. S., LAURA, T. L., SCHILDT, A. N., MEDEIROS, A. A. D., ALISINA, P. J. **CEstratégia de estabilização para um helicóptero quadri-rotor autônomo**. XIX Congresso Brasileiro de Automática, CBA 2012.

GOMES, R. C.; GEORGE A. P. T. "**Estratégia de navegação tolerante a falha de motores em quadrirotoros com controle PID.**" Anais do EATI. Departamento de Engenharia de Teleinformática. Universidade Federal do Ceará. Frederico Westphalen – RS, 2015.

GOMIDE, F. A. C., & GUDWIN, R. R. “**Modelagem, controle, sistemas e lógica fuzzy.**” SBA controle & Automação, 4(3), 97-115. (1994).

JENKINS, D. R.; LANDIS, T.; MILLER, J. “**American X-Vehicles: An Inventory-X-1 to X-50**”. 2005. Disponível em:< <http://history.nasa.gov/monograph31.pdf>>. Acessado em 05 de Dezembro 2016.

VERBEKE, J.; HULENS, D.; RAMON, H.; GOEDEMÉ, T.; SCHUTTER, J. “**The design and construction of a high endurance hexacopter suited for narrow corridors**”. In: Unmanned Aircraft Systems (ICUAS), 2014 International Conference on. IEEE, 2014. p. 543-551.

LOPES, L. L. et al. “**Modelagem e validação de um quadrimotor ardrone parrot**”. In: XIX Congresso Brasileiro de Automática. Anais Campina Grande, PB: CBA, 2012. p. 01-06.

MARRO, A. A., SOUZA, A., CAVALCANTE, E., BEZERRA, G. S., & NUNES, R. “**Lógica fuzzy: conceitos e aplicações**”. Natal: Universidade Federal do Rio Grande do Norte (UFRN), 2010.

MATTIELLO, C. D. “**Comparativo entre controlador pid e fuzzy no controle de atitude em um quadricóptero**”. Trabalho de conclusão de curso. Pato Branco, 2014.

NETTO, J. C. “**Controladores Nebulosos Aplicados a Processos Industriais: Estudo Comparativo de Métodos de Sintonia**”. Tese de Doutorado. Universidade Federal de Juiz de Fora, 2005.

NISE, N. S. “**Engenharia de sistemas de controle**”. Tradução de Fernando Ribeiro da Silva. 5ª. ed. Rio de Janeiro: LTC Editora, 2011.

OGATA, K. “**Engenharia de controle moderno**”. Tradução de Heloísa Coimbra de Souza. 5ª. ed. São Paulo: Editora PEARSON, 2011.

OSCARSON, O. “**Design, Modeling and Control of an Octocopter**”. Programa de Mestrado em Matemática Aplicada e Computacional. Instituto Real de Tecnologia. 2015.

PARROT, **Technical Specifications Ar-Drone**, Disponível em: <<https://www.parrot.com/uk/drones/parrot-ardrone-20-elite-edition#technicals>>. Acessado em 10 de junho 2016.

POUNDS, P. E. I. **Design, construction and control of a large quadrotor micro air vehicle.** Australian National University, 2007.

POUNDS, P.; MAHONY, R.; CORKE, P. “**Modelling and Control of a Large Quadrotor Robot**”. *Control Engineering Practice*, [s.l.], v. 18, n. 7, p. 691-699, July 2010.

POUNDS, P. et al. “**Towards dynamically-favourable quad-rotor aerial robots**”. In: *Proceedings of the Australian Conference on Robotics and Automation*. Canberra, Australia: ACRA, 2004.

POUNDS, P.; MAHONY, R.; CORKE, P. “**Modelling and control of a quad-rotor robot**”. In: *Proceedings of the Australian Conference on Robotics and Automation*. Auckland, New Zealand: ACRA, 2006.

PIZETTA, I. H. B. “**Uma plataforma hardware-in-loop para vants de asas rotativas**”. Dissertação de Mestrado em Engenharia Elétrica, Centro Tecnológico da Universidade Federal do Espírito Santo, UFES, Brasil. p. 01-121, 2013.

PIZETTA, I. H. B. et al. “**Uma plataforma hardware-in-loop para vants de asas rotativas**”. In: *Anais do XIX Congresso Brasileiro de Automática*, Campina Grande, PB: CBA, p. 3565-3570. 9, 2012.

SÁ, R. C. “**Construção, modelagem dinâmica e controle pid para estabilidade de um veículo aéreo não tripulado do tipo quadrirotor**”. Dissertação de Mestrado em Engenharia de Teleinformática, Universidade Federal do Ceará, UFC, Brasil. p. 01-74, 2012.

SÁ, R. C. *et al.* “**Construction and PID Control for Stability of an Unmanned Aerial Vehicle of the Type Quadrotor**”. In: *IEEE LATIN AMERICAN ROBOTICS SYMPOSIUM (LARS/LAC)*, 2013, Arequipa. *Proceedings of the 2013 IEEE Latin American Robotics Symposium*. Arequipa: IEEE. p. 95-99, 2013.

SAMPAIO, R.P. **Sistema de controle de atitude embarcado para voo autônomo de aviões em escala.** Dissertação de Mestrado em Mecatrônica. Escola Politécnica e Instituto de Matemática, Universidade Federal da Bahia, Salvador, 2006.

SANTANA, P. H. R. Q. A.; BORGES, G.A. “**Modelagem e Controle de Quadrirotos. IX Simpósio Brasileiro de Automação Inteligente**”. SBAI, 2009.

SANTANA, P. H. R. Q. A. *et al.* **Filtragem estocástica para sistemas híbridos e suas aplicações em robótica aérea.** 21st International Congress of Mechanical Engineering October 24-28, Natal, RN, 2011.

SANTANA, L. V. et al. **Uma estrutura sensorial e de comunicação para o controle de altitude e guinada de um mini-helicóptero autônomo.** In: *Anais do XVIII Congresso Brasileiro de Automática*. Bonito - MS, Brasil: SBA, 2010. p. 1840–1845.

SANTANA, L. V. et al. **Hovering control of a miniature helicopter attached to a platform.** In: *Proceedings of the 20th IEEE International Symposium on Industrial Electronics*. Gdansk, Poland: IEEE, 2011. p. 2231–2236.

SILVA, M. F. “**Modelagem dinâmica, identificação de parâmetros e controle de um veículo aéreo não tripulado do tipo quadricóptero**”. Dissertação de mestrado, UFERSA, Mossoró, RN. 2015.

SIMÕES, M. G.; SHAW, I. S. “**Controle e modelagem *Fuzzy***”. Blucher: FAPESP, São Paulo, BRA, 2007.

SOUZA, A. C. Z.; PINHEIRO, C. A. M. “**Introdução à Modelagem, Análise e Simulação de Sistemas Dinâmicos**”. Rio de Janeiro: Interciência, 2008. 173 p.

SUZUKI, N. K. G., “**Proposta de uma arquitetura de controle híbrida fuzzy-PID para a realização de manobras em VANTs**”. Dissertação de Mestrado em Computação, Universidade do Vale do Itajaí, UNIVALI, Brasil. v. 1. p. 01-99, 2009.

VILAS BOAS, A. S. C. A.; VILAS BOAS, E. HONÓRIO, L. M. “**Análise e Comparação das Técnicas de Controle PID, LQR e Backstepping para Estabilização de Voo de Quadricópteros**”. Em: SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, 11., 2013, Fortaleza. Anais do XI Simpósio Brasileiro de Automação Inteligente. Fortaleza: UFC, 2013. Não paginado. Artigo 5139.

WIEREMA, M. B. SC. “**Design, implementation and flight test of indoor navigation and control system for a quadrotor UAV**”, Dissertação de mestrado, Delft University of Technology, Holanda, 2008.