

ALEX PEREIRA BRAGA

**SIMULAÇÃO E CONTROLE DE UM *QUADCOPTER* UTILIZANDO LÓGICA
*FUZZY***

Guaratinguetá - SP

2013

Alex Pereira Braga

**SIMULAÇÃO E CONTROLE DE UM *QUADCOPTER* UTILIZANDO LÓGICA
*FUZZY***

Trabalho de Graduação apresentado ao Conselho de Curso de Graduação em Engenharia Elétrica da Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, como parte dos requisitos para obtenção do diploma de Graduação em Engenharia Elétrica.

Orientadora: Profª. Dra. Paloma M. S. Rocha
Rizol

Guaratinguetá
2013

B813s Braga, Alex Pereira
Simulação e controle de um quadcopter utilizando lógica fuzzy / Alex Pereira Braga. – Guaratinguetá : [s.n], 2013
89 f. : il.
Bibliografia : f. 85-87

Trabalho de Graduação em Engenharia Elétrica –
Universidade Estadual Paulista, Faculdade de Engenharia de
Guaratinguetá, 2013
Orientadora: Profa. Dra. Paloma Maria S. Rocha Rizol

1.Lógica difusa 2. SIMULINK(Software) I.Título.
CDU 510.6

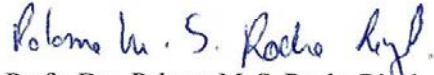
Alex Pereira Braga

ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO PARTE
DO REQUISITO PARA A OBTENÇÃO DO DIPLOMA DE **"GRADUADO EM
ENGENHARIA ELÉTRICA"**

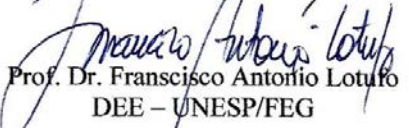
APROVADO EM SUA FORMA FINAL PELO CONSELHO DE CURSO DE
GRADUAÇÃO EM ENGENHARIA ELÉTRICA


Prof. Dr. LEONARDO MESQUITA
Coordenador

BANCA EXAMINADORA:


Profa. Dra. Paloma M. S. Rocha Rizol
Orientadora/ DEE – UNESP/FEG


Prof. Dr. Leonardo Mesquita
DEE – UNESP/FEG


Prof. Dr. Francisco Antonio Lotufo
DEE – UNESP/FEG

Dezembro de 2013

de modo especial, aos meus pais Wilson e Paulete e à minha irmã Camila, pelo apoio e confiança incondicional.

AGRADECIMENTOS

Gostaria de agradecer, primeiramente, aos meu pais pela orientação e conselhos, e dedicação durante todo a minha vida,

à minha orientadora, *Prof^a. Dra. Paloma Maria Silva Rocha Rizol*, pela paciência, disponibilidade e confiança que me passou ao desenvolvimento deste trabalho, sempre pronta a ajudar e dar sua opinião.

à minha namorada Paula, pela sensatez e companheirismo todos os dias,
aos atuais e ex-moradores da República Mau-Mau, lugar de grandes amigos para a vida toda,

às amizades que fiz durante a graduação, em especial aos meus amigos Henrique, Pedro, Fernando e Daniel.

BRAGA, A. P. **Simulação e controle de um *quadcopter* utilizando lógica *fuzzy***. 2013. 89f. Trabalho de Graduação (Graduação em Engenharia Elétrica) – Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2013.

RESUMO

O presente trabalho trata do desenvolvimento do modelo virtual; de um *quadcopter* utilizando controladores *fuzzy*, implementado no *simulink* do *matlab*. Primeiramente é apresentado um breve histórico sobre *quadcopters* e uma introdução a lógica *fuzzy*. Seguido de uma visão geral da dinâmica do sistema a ser implementado e de sua modelagem matemática utilizando o formalismo Newton-Euler. A partir das equações obtidas na modelagem implementa-se um modelo virtual utilizando o software *simulink*, e então cada subsistema é descrito detalhadamente e também o controlador *fuzzy*. Propõe-se também um possível protótipo descrevendo cada componente necessário. Por fim, apresenta-se os resultados obtidos do simulador e discute-se os resultados entre um sistema com sensores ideais e com sensores não ideais.

PALAVRAS CHAVE: *Quadcopter*. Controle *Fuzzy*. *Simulink*. Lógica *Fuzzy*. Newton-Euler. VANT.

BRAGA, A. P. **Simulation and control of a quadcopter model applying fuzzy logic**. 2013. 89f. Graduate Work (Graduate in Electrical Engineering) – Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2013.

ABSTRACT

The present work develops a model to simulate the dynamics of a quadcopter being controlled by a PD fuzzy controller. Initially is presented a brief history of quadcopters an introduction to fuzzy logic and fuzzy control systems. Afterwards is presented an overview of the quadcopter dynamics and the mathematical modelling development applying Newton-Euler method. Then the modelling are implemented in a Simulink model in addition to a PD fuzzy controller. A prototype proposition is made, by describing each necessary component to build up a quadcopter. In the end the results from the simulators are discussed and compared due to the discrepancy between the model using ideal sensor and the model using non-ideal sensors.

Keywords: Quadcopter. Fuzzy Control. Simulink. Fuzzy Logic. Newton-Euler. UAV. VTOL.

LISTA DE FIGURAS

Figura 1 – Oemichen 2.....	15
Figura 2 – ConvertaWings.	16
Figura 3 – Curtiss-Wright X-19.....	17
Figura 4 – BELL X-22.....	18
Figura 5 – WGM 21.....	19
Figura 6 – <i>Quadcopter</i> do laboratório GRASP.	20
Figura 7 – Abordagem da Universidade de Minnesota.....	21
Figura 8 – <i>Quadcopter</i> com controle totalmente visual utilizando câmera.	22
Figura 9 – <i>Quadcopter</i> feito em série pela DJI	23
Figura 10 – a)Conjunto clássico b) Conjunto <i>fuzzy</i>	25
Figura 11 – Função de pertinência usando a) conjunto clássico b)Conjunto <i>fuzzy</i>	26
Figura 12- Função pertinência triangular	27
Figura 13- Função pertinência trapezoidal.....	28
Figura 14 - Operador união aplicado a conjuntos <i>fuzzy</i>	28
Figura 15 – Operador interseção aplicado a conjuntos <i>fuzzy</i>	29
Figura 16 – Operador complemento aplicado a um conjunto <i>fuzzy</i>	29
Figura 17 – Sistema em malha fechada com controlador <i>fuzzy</i>	30
Figura 18 – Método de inferência Mamdani para um controlador <i>fuzzy</i> de duas entradas.....	31
Figura 19 – Método de defuzzificação centro de área	32
Figura 20 – Método de defuzzificação da altura	33
Figura 21 - Movimentos angulares do <i>quadcopter</i>	34
Figura 22 – <i>Quadcopter</i> em posição de estabilidade.....	36
Figura 23 – <i>Quadcopter</i> em aceleração vertical a)positiva b) negativa.....	37
Figura 24 – Atuação do <i>quadcopter</i> para movimento <i>roll</i> a)postivo b)negativo	38
Figura 25 – Atuação do <i>quadcopter</i> para movimento <i>pitch</i> a)postivo b)negativo	39
Figura 26– Atuação do <i>quadcopter</i> para movimento <i>yaw</i> a)positivo b)negativo.....	40
Figura 27 – Representação da posição do <i>quadcopter</i> em relação a terra	42
Figura 28 – Desenho técnico do <i>frame</i> DJI450	52
Figura 29 –STM32F3DISCOVERY	53
Figura 30 – L3GD20.....	54
Figura 31 - Acelerômetro LSM303DLHC	54
Figura 32 – LSM303DLHC	55

Figura 33 - HC-SR04	55
Figura 34 – Modelo NTM Prop Drive Series 28-30S 900kv	56
Figura 35 – Hélices 10x4.5.....	56
Figura 36 - Hobby King 30A ESC 3A UBEC.....	57
Figura 37 - Turnigy 2200mAh 3S 40C Lipo Pack	58
Figura 38 - FPV 433Mhz Radio Telemetry Kit 100mW V1.1	59
Figura 39 – Visão geral da comunicação dos sistemas.....	59
Figura 40 – Visão geral do simulador.....	61
Figura 41 – Subsistema entrada.....	62
Figura 42 – Subsistema Controlador <i>Fuzzy</i>	63
Figura 43 – Função pertinência erro proporcional	64
Figura 44 – Superfície do controlador <i>fuzzy</i>	65
Figura 45 – Subsistema atuadores	67
Figura 46 – Subsistema dinâmica	69
Figura 47 – Subsistema Sensores	70
Figura 48 – Subsistema Gráficos.....	71
Figura 49 – Resposta da altura em relação a uma entrada igual a posição inicial.....	72
Figura 50 – Entrada para avaliação de ganho de altura para o <i>quadcopter</i>	73
Figura 51 – Resposta do sistema ao ganho de altura utilizando sensores ideais.....	74
Figura 52 – Entrada para a avaliação do ângulo <i>Roll</i>	74
Figura 53 – Resposta do ângulo <i>roll</i> de um sistema com sensores ideais	75
Figura 54 – Entrada do simulador para avaliação do ângulo <i>pitch</i>	75
Figura 55 – Resposta do ângulo <i>pitch</i> de um sistema com sensores ideais	76
Figura 56 – Entrada do simulador para avaliação do ângulo <i>yaw</i>	76
Figura 57 - Resposta do ângulo <i>yaw</i> de um sistema com sensores ideais	77
Figura 58 - Resposta do sistema ao ganho de altura utilizando sensores não ideais	78
Figura 59 - Resposta do ângulo <i>roll</i> de um sistema com sensores não ideais.....	79
Figura 60 - Resposta do ângulo <i>pitch</i> de um sistema com sensores não ideais.....	80
Figura 61 - Resposta do ângulo <i>yaw</i> de um sistema com sensores não ideais	81

LISTA DE TABELAS

Tabela 1 – Avaliação das funções pertinência	26
Tabela 2 – Associação dos movimentos fundamentais a variáveis de força e torque	35
Tabela 3 – Visão geral da dinâmica do <i>quadcopter</i>	40
Tabela 4 – Constantes adotadas.....	60
Tabela 5 – Ganho e saturação dos controladores.....	63
Tabela 6 – Parâmetros da função pertinência erro proporcional	64
Tabela 7 – Base de regras do controlador <i>fuzzy</i>	65

SUMÁRIO

1	INTRODUÇÃO	13
1.1	CONSIDERAÇÕES GERAIS.....	13
1.2	OBJETIVOS	13
1.3	MOTIVAÇÃO	13
1.4	PLANO DE TRABALHO	13
2	EMBASAMENTO TEÓRICO.....	15
2.1	HISTÓRIA DO <i>QUADCOPTER</i>	15
2.2	APLICAÇÕES.....	19
2.3	ESTADO DA ARTE.....	20
2.3.1	Universidade da Pennsylvania.....	20
2.3.2	Universidade de Minnesota.....	20
2.3.3	Universidade técnica de Viena.....	21
2.3.4	Universidade técnica de Munique	22
2.3.5	Aeroquad.....	22
2.3.6	DJI	23
2.4	LÓGICA <i>FUZZY</i>	23
2.4.1	Aplicações.....	24
2.4.2	Conjuntos <i>Fuzzy</i>	24
2.4.3	Funções pertinência	25
2.4.4	Operadores <i>Fuzzy</i>	28
2.4.5	Controlador <i>Fuzzy</i>	30
2.4.6	Fuzzificador.....	30
2.4.7	Base de regras.....	30
2.4.8	Inferência <i>fuzzy</i>	30
2.4.9	Defuzzificador	31
3	MODELO DO QUADCOPTER	34
3.1	CONCEITOS DE FUNCIONAMENTO	34
3.1.1	Hovering	35
3.1.2	<i>Altura</i> – $U1[N]$	36
3.1.3	<i>Roll</i> – $U2[N.m]$	37
3.1.4	<i>Pitch</i> – $U3[N.m]$	38

3.1.5	Yaw – U4[N.m].	39
3.2	MODELO DE NEWTON EULER	40
3.3	ARQUITETURA DO PROTÓTIPO	52
3.3.1	Frame.	52
3.3.2	Microcontrolador	52
3.3.3	Giroscópio.	53
3.3.4	Acelerômetro	54
3.3.5	Magnetômetro - LSM303DLHC	54
3.3.6	Sonar.	55
3.3.7	Motor.	55
3.3.8	Hélice	56
3.3.9	ESC	56
3.3.10	Bateria	57
3.3.11	Sistema de Telemetria	58
3.3.12	Estação Base	59
3.4	MODELO VIRTUAL NO SIMULINK	60
3.4.1	Entrada.	61
3.4.2	Controlador <i>Fuzzy</i>.	62
3.4.3	Atuadores	66
3.4.4	Dinâmica do <i>quadcopter</i>	67
3.4.5	Sensores	69
3.4.6	Gráficos	70
4	ÁNALISE DE RESULTADOS	72
4.1	SIMULAÇÃO USANDO SENSORES IDEAIS	72
4.1.1	Hovering	72
4.1.2	Altura.	72
4.1.3	<i>Roll</i>.	74
4.1.4	<i>Pitch</i>	75
4.1.5	<i>Yaw</i>.	76
4.2	LIMITAÇÃO DOS SENSORES	77
4.2.1	Altura.	77
4.2.2	<i>Roll</i>.	78
4.2.3	<i>Pitch</i>	79

4.2.4	<i>Yaw</i>.....	80
5	CONCLUSÃO	82
6	SUGESTÕES E TRABALHOS FUTUROS	83
	REFERÊNCIAS.....	84
	APÊNDICE A – ESTRUTURA FIS DO CONTROLADOR	87

1 INTRODUÇÃO

1.1 CONSIDERAÇÕES GERAIS

Quadcopter é um veículo capaz de aterrissar e decolar na vertical, caracterizando-o como um veículo VTOL (*Vertical Take Off and Landing*). Possuem 4 motores dispostos numa estrutura, chamada *frame*, em X, sendo que há um motor em cada extremidade de estrutura. O controle da dinâmica do *quadcopter* é feito através da variação da velocidade angular dos motores, diferentemente da abordagem do helicóptero onde controla-se modificando o ângulo de ataque da hélice (ALTUG, 2002).

Outro importante ramo de aplicação de *quadcopters* é na pesquisa de métodos de controle no meio acadêmico, comumente utilizado em pesquisas na área de engenharia de controle nas mais diversas universidades globalmente.

1.2 OBJETIVOS

O objetivo do presente trabalho é a aplicação do método *fuzzy* de controle a um modelo matemático de um *quadcopter* utilizando o *software simulink*. A altura e os ângulos de *roll*, *pitch* e *yaw* devem ser estabilizados por meio do controle aplicado, avaliando-se também a diferença nos resultados entre o uso de sensores ideais, ou seja sem ruído e erros agregados, com sensores os quais suas limitações técnicas são modeladas como o intervalo de atuação, frequência de operação e erros relativos a ruídos e quantização.

1.3 MOTIVAÇÃO

Os veículos aéreos não tripulados (VANT) que são também VTOL, caso do *quadcopter*, são capazes de voar em velocidades reduzidas e alcançar lugares de difícil acesso, sendo por isso aplicado para resgate em casos de desastres naturais, desabamentos ou locais de difícil acesso (PITMAN, 2010).

A dinâmica do *quadcopter*, possuindo seis graus de liberdade, sendo três translacionais e três rotacionais, e somente 4 variáveis de entrada, as velocidades angulares dos motores, torna-se um desafio mantê-lo dentro da estabilidade. Sendo então o principal fator motivador a verificação do desempenho de um controlador *fuzzy* aplicado a um modelo matemático de um *quadcopter*, desenvolvendo-se um simulador para avalia-lo.

1.4 PLANO DE TRABALHO

Nessa seção é apresentado uma visão geral do trabalho como um todo, apresentando uma breve descrição dos temas abordados em cada capítulo.

No capítulo dois é apresentado alguns pontos da história do *quadcopter*, suas aplicações e estado da arte assim como uma breve introdução à lógica *fuzzy*.

O capítulo três consiste de três partes, a primeira é a apresentação do princípio de funcionamento do *quadcopter*, a segunda parte refere-se a modelagem matemática utilizando o método *Newton-Euler* e por fim o desenvolvimento do modelo no *simulink* descrendo em detalhes os seus subsistemas.

No capítulo quatro é apresentado os resultados referentes as simulações feitas usando o modelo do *quadcopter* desenvolvido no capítulo três, sendo então separado em duas partes, uma parte consiste dos resultados relativos ao modelo utilizando sensores ideais e a segunda parte utilizando sensores com capacidades limitadas e com erro introduzido.

No capítulo cinco discute-se sobre os resultados do capítulo quatro, concluindo como o modelo dos sensores afetam o sistema e também se o controle *fuzzy* é válido para a aplicação.

Por fim, são apresentados possíveis temas para trabalhos futuros, no capítulo 6.

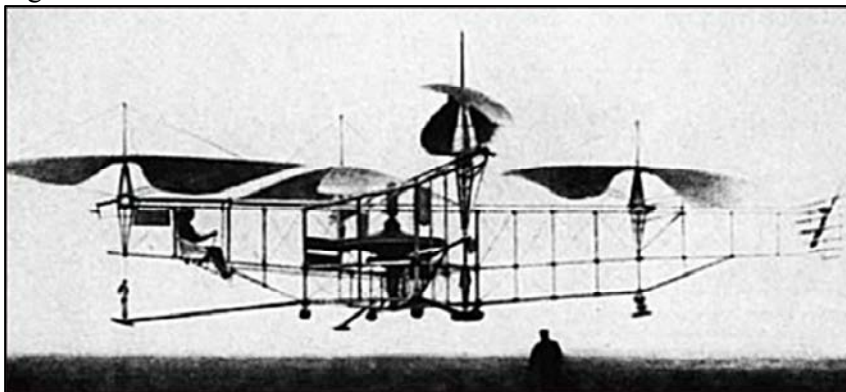
2 EMBASAMENTO TEÓRICO

2.1 HISTÓRIA DO *QUADCOPTER*

Étienne Oehmichen, engenheiro francês nascido em 1884 foi um dos pioneiros no ramo de máquinas com asas rotativas durante a década 20, no total ele projetou seis modelos sendo o segundo deles projetado com quatro rotores e oito hélices todas ligadas a um único motor. O projeto consistia em uma estrutura tubular de metal com rotores de duas pás em cada extremidade, sendo as pás das hélices moveis, podendo ser controlado o ângulo em que se posicionam, gerando mais ou menos empuxo. Cinco das hélices giravam no plano horizontal estabilizando a máquina lateralmente, uma hélice era montada verticalmente para conduzir a máquina na direção desejada e as últimas duas hélices eram responsáveis pela propulsão da máquina no sentido desejado (OURSLAND, 2010).

O Oemichen 2, a segunda máquina projetada por Étienne Oehmichen apresentou bons resultados em estabilidade e controle para a época e durante alguns anos após a construção houveram aproximadamente mil testes, obtendo como resultados a possibilidade de alguns minutos de voo. Na época de tal construção os testes alcançaram o recorde de distância percorrida para helicópteros, com a distância de 360 metros, anos depois conseguiu percorrer a distância de um quilometro completando o voo pousando no mesmo local de decolagem. A Figura 1 ilustra a máquina Oemichen 2 (SKIBA, 2013).

Figura 1 – Oemichen 2

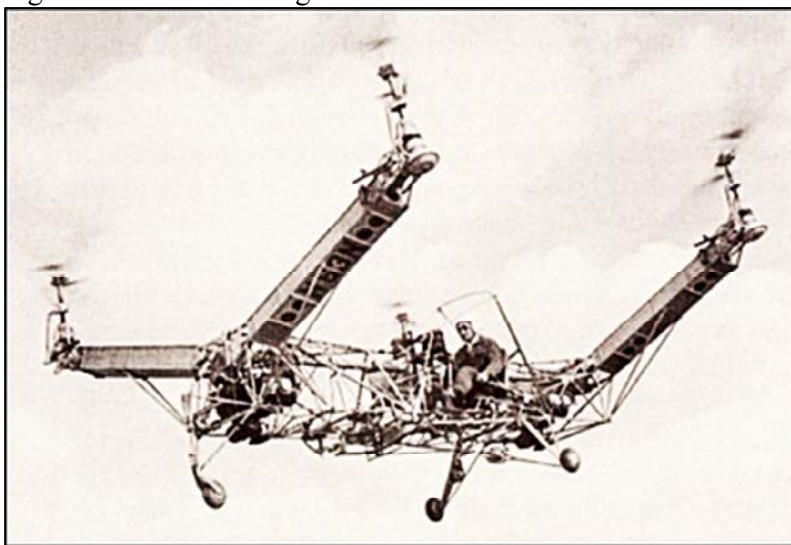


Fonte: (DOMINGUES, 2009)

O projeto Convertawings foi uma revisão da máquina de Oehmichen de autoria de G. de Bothezat, consistindo de 4 rotores arranjados dois a dois montados em H. Foi o primeiro helicóptero a ser encomendado pelo exército dos Estados Unidos. O controle foi muito simplificado em relação a seus antecessores e foi introduzido também a mudança de impulso entre os motores, a estrutura era de ferro em formato de tubo e rotores em alumínio, a potência

era fornecida por dois motores conectados aos sistemas de rotação por meio de diversas correias. O primeiro voo foi em março de 1956, devido ao seu formato em H a complexidade do controle foi diminuída pelo fato de eliminar alguns fatores que o complicam como o movimento de *roll* e *pitch*, apresentado na Figura 2 (DE SOUSA, 2011).

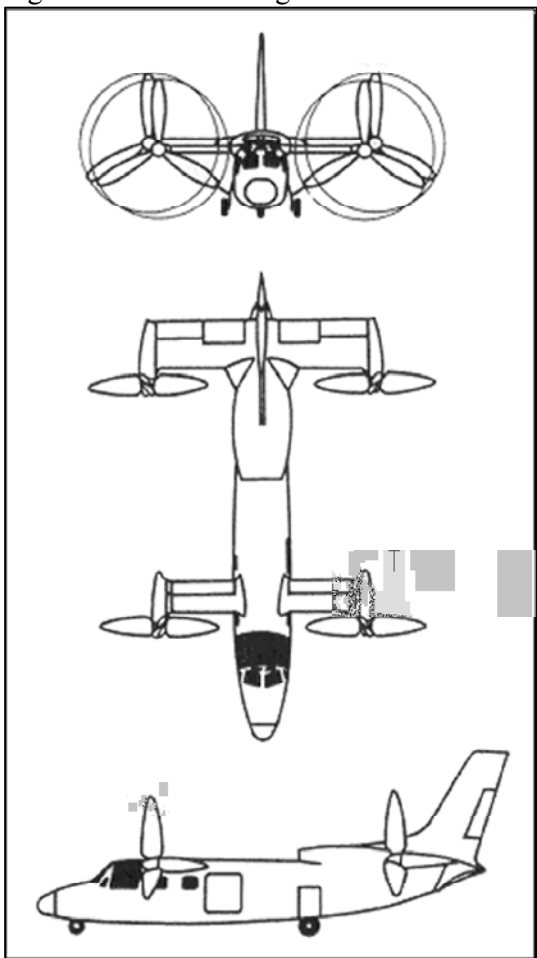
Figura 2 – ConvertaWings.



Fonte: (DOMINGUES, 2009)

Outro projeto de veículo de pouso e aterrissagem vertical (VTOL) foi o Curtiss-Wright X-19, mostrado na Figura 3, um avião experimental americano que tinha a capacidade de virar seus rotores da posição vertical para horizontal e vice-versa, possibilitando a decolagem vertical do avião, o Curtiss era um veículo de transporte de passageiros, seu primeiro voo foi em 1963 e após um acidente em 1965 o projeto foi descontinuado restando apenas um segundo protótipo guardado no museu nacional do estados unidos (DE SOUSA, 2011).

Figura 3 – Curtiss-Wright X-19.



Fonte: (DE SOUSA, 2011) .

O BELL X-22, ilustrado na Figura 4, foi desenvolvido em 1966 a pedido do exército americano como um veículo de decolagem e pouso vertical ou com aplicação em transporte. A implementação foi feita por quatro asas de três hélices cada uma, cada asa conectada a uma turbina à gás montada em pares, duas frontais e duas traseiras, o controle era feito alterando o ângulos das pás das hélices em combinação com outras peças aerodinâmicas (DE SOUSA, 2011).

Figura 4 – BELL X-22.



Fonte: (DE SOUSA, 2011)

Os alemães também tiveram sua participação no desenvolvimento de veículos de VTOL com o modelo WGM 21 datado de 1970, foi implementado com um motor de uma motocicleta BMW numa estrutura em X com uma hélice em cada extremidade, com o *pitch* sendo controlado por pedais dentro de um *cockpit* aberto, conforme a Figura 5 (OURSLAND,2010).

Figura 5 – WGM 21.



Fonte: (DE SOUSA, 2011)

2.2 APLICAÇÕES

Nos últimos anos houve um crescente interesse em *quadcopters* porém diferente dos apresentados no início deste capítulo, atualmente exclusivamente com característica de veículos não tripulados (UAV ou VANT), e de tamanho reduzido, podendo ser autônomos ou remotamente controláveis, podendo assim ser aplicados em situações onde a presença de um humano é inviável ou substituível por um UAV (ENGEL, 2012).

Quadcopters estão cada vez mais presentes substituindo ou auxiliando humanos em desastres naturais ou acidentes, como por exemplo: inundações, terremotos, desabamentos, tornados, entre outros. Onde o acesso é difícil e as vezes inviável. Cabendo o uso de *quadcopter* para fazer uma primeira avaliação da situação, protegendo assim a vida do profissional da área. A aplicação em vigilância também é explorada por esses veículos, muitas vezes desenvolvidos por governos para setores estratégicos de defesa, a fim de proteger a vida do passageiro pode-se enviar um UAV que muitas vezes consegue realizar o objetivo com mais rapidez e precisão (JADHAV, 2013).

2.3 ESTADO DA ARTE

Atualmente muitos projetos estão sendo desenvolvidos no âmbito acadêmico e também comercial, ao longo dos próximos subcapítulos será apresentado alguns dos projetos desenvolvidos em universidades por todo o mundo e também produtos disponíveis a consumidores com ou sem fundo lucrativo.

2.3.1 Universidade da Pennsylvania

No laboratório GRASP da universidade da Pennsylvania foi desenvolvido um projeto com UAVs para uso militar. Foi obtido um nível de controle tal que os *quadcopters* conseguiram voar por espaços extremamente pequenos e também trabalhar em conjunto com outros *quadcopters* para levantar objetos pesados, assim, podendo ser aplicado em campo de batalha auxiliando a tropa. A Figura 6 apresenta o modelo criado no Laboratório GRASP (VIEIRA, 2011).

Figura 6 – *Quadcopter* do laboratório GRASP.



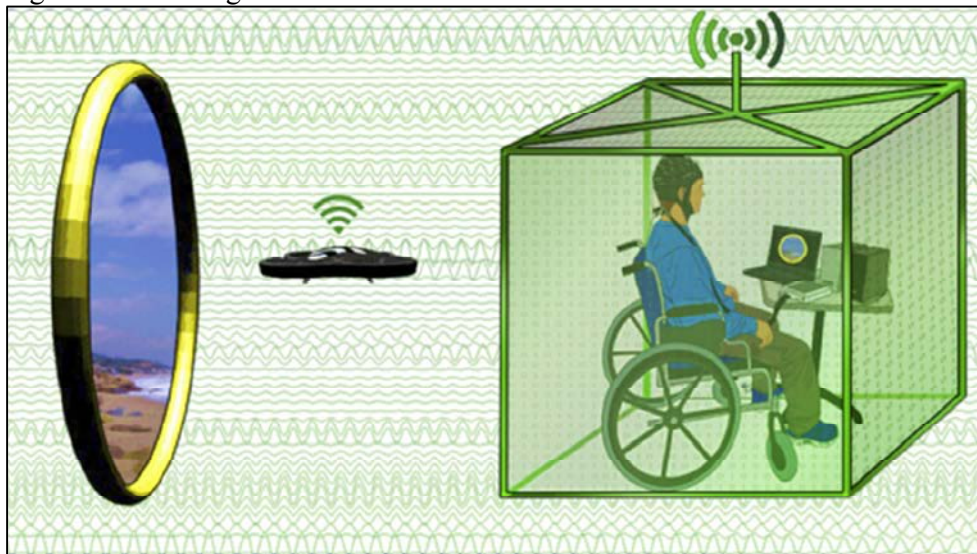
Fonte: (SKIBA, 2013).

2.3.2 Universidade de Minnesota

A escola de ciência e engenharia da universidade de Minnesota desenvolveu um sistema que possibilita o controle de *quadcopters* por meio da mente, 5 pessoas foram treinadas para controlá-lo em um ambiente 3d, uma câmera no *quadcopter* auxiliava as pessoas a saberem

como o *quadcopters* estava reagindo, servindo assim como *feedback* ao seu controle, foi possível alcançar 90,5% dos alvos voando a uma velocidade de 0,69 metros por segundo. A estrutura básica é apresentada na Figura 7 (LAFLEUR, 2013).

Figura 7 – Abordagem da Universidade de Minnesota.

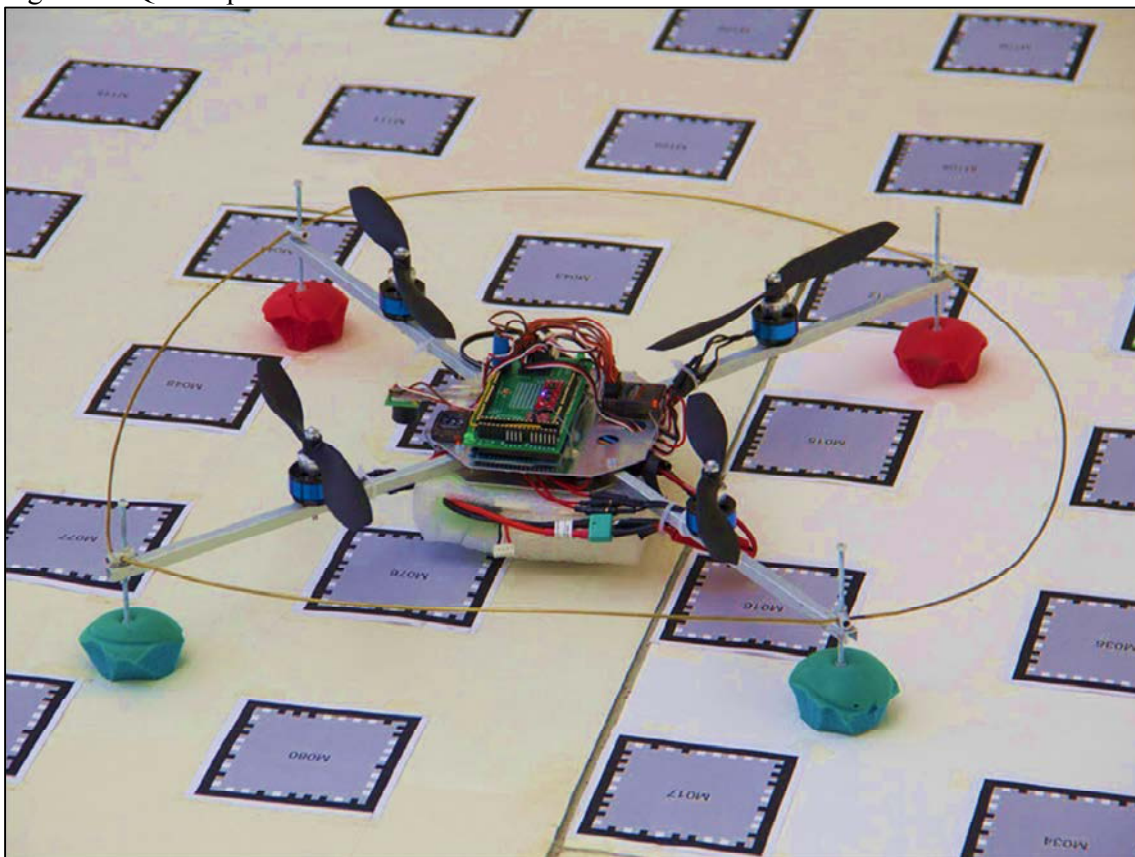


Fonte: (LAFLEUR, 2013)

2.3.3 Universidade técnica de Viena

Pesquisadores da universidade técnica de Viena projetaram um *quadcopter* totalmente autônomo, como o apresentado na Figura 8, sendo capaz de voar em ambientes fechados, com todo o poder computacional necessário já embutido ao veículo, sem necessidade de um computador externo. A central de processamento é um smartphone responsável pelo processamento de imagem e também controle do veículo implementado usando a imagem da câmera e um aplicativo desenvolvido para o smartphone, responsável então pela estabilidade e autonomia (GELLERSEN, 2009).

Figura 8 – Quadcopter com controle totalmente visual utilizando câmera.



Fonte: (GELLERSEN, 2009)

2.3.4 Universidade técnica de Munique

O departamento de pesquisa em visão computacional da universidade técnica de Munique também tem um projeto utilizando *quadcopters* para aplicações que o utilizam como câmera voadora em vídeos esportivos, para fiscalização de ponte e vigilância em agricultura. O objetivo deste projeto é resolver alguns pontos ainda não totalmente resolvidos nesse ramo de pesquisa, como capacidade de carga, localização e navegação em um ambiente tridimensional. Neste projeto é utilizado como sensor principal câmeras monoculares e RGB-D (ENGEL, 2012).

2.3.5 Aeroquad

É uma base de dados compartilhada entre amadores que desejam criar seus próprios *quadcopters* utilizando como controlador o arduino. Milhares de usuários compartilham seus resultados e discutem problemas que ocorrem durante o desenvolvimento, visando o compartilhamento de códigos e esquemas elétricos, sendo também disponibilizado guias e manuais para auxiliar os interessados (OURSLAND, 2010).

2.3.6 DJI

A empresa DJI foi uma das pioneiras a desenvolver *quadcopters* comerciais, como o modelo apresentado na Figura 9, atualmente comercializa desde o veículo completo, assim como suas peças e acessórios. Em seu portfólio há além dos convencionais UAV de quatro rotores, veículos com 6 rotores, suas aplicações indicadas são recreação e filmagem contando com diversos acessórios para tais atividades. Contam também com gps, e podem ser pré-programados para sobrevoar um caminho determinado na estação de controle terrestre, um tablet, smartphone ou computador (ENOMOTO, 2010).

Figura 9 – *Quadcopter* feito em série pela DJI



Fonte: (DJI, 2013)

Como o objetivo deste trabalho é realizar o controle *fuzzy* do *quadcopter*. A seguir, será apresentado uma visão geral sobre este assunto.

2.4 LÓGICA FUZZY

A lógica *fuzzy* é uma generalização da lógica booleana, a diferença é que ao invés de se trabalhar com variáveis binárias, trabalha-se com grau de pertinência entre zero e um. A lógica *fuzzy* foi desenvolvida para aproximar as informações da linguagem humana. Esta lógica foi desenvolvido por Lofti Zadeh da Universidade da Califórnia em 1965, utilizando variáveis

linguísticas, anos depois foi publicado um estudo sobre teoria das possibilidades, no qual discutiu-se a incerteza da informação (CHEN, 2000).

2.4.1 Aplicações

A teoria da lógica *fuzzy* não foi aceita rapidamente e por consequência, demorou-se para haver uma aplicação em grande escala deste assunto. Uma das grandes aplicações usando lógica *fuzzy* foi um metrô completamente automatizado em Sandai no Japão, onde toda a regulação foi implementada utilizando lógica *fuzzy* nos anos 80, projetado e desenvolvido pela empresa Hitachi também de origem japonesa (RODRIGUEZ, 2010).

Na Europa as aplicações da lógica *fuzzy* aumentaram bastante após a diminuição das discussões sobre a capacidade e confiabilidade da teoria em meados dos anos noventa.

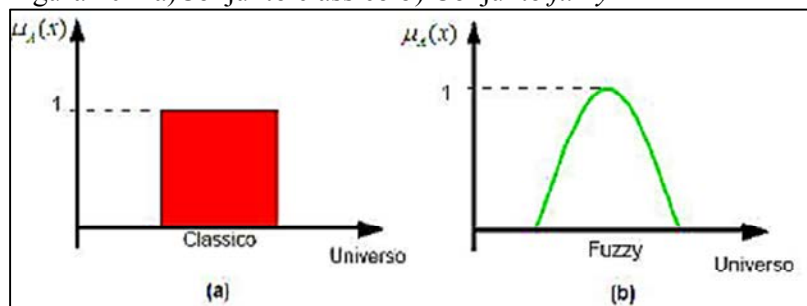
Os campos de aplicação da lógica *fuzzy* são dos mais diversos possíveis, por exemplo: controle de sistemas eletromecânicos, economia, processamento de imagem, inteligência artificial entre outros. Assim fica-se claro de que se trata de uma teoria com ampla aplicação desde foco automático de câmeras a predição e modelagem de terremotos.

2.4.2 Conjuntos Fuzzy

Como dito anteriormente, na teoria clássica os elementos assumem valores binários, ou seja, são limitados a 2 valores, sejam linguísticos (bom/mal, alto/baixo) ou numéricos (0/1). Na lógica *fuzzy*, os elementos possuem um grau de pertinência que pode variar entre zero e um, ficando assim mais próxima da forma do pensamento humano (CASTILLO, 2008).

Na Figura 10 expressa-se em (a) um conjunto clássico e em (b) um conjunto *fuzzy*, demonstrando que conjuntos *fuzzy* não tem limites bem definidos como no clássico onde os elementos tem valores que ou são zeros ou uns e sim valores intermediários. O valor a ser atribuído é o da função pertinência $\mu_A(x)$, que será explicado posteriormente, verifica-se que no conjunto *fuzzy* a função pertinência varia gradualmente possibilitando a existência de valores que não são zeros nem uns (CASTILLO, 2008).

Figura 10 – a) Conjunto clássico b) Conjunto *fuzzy*



Fonte: (RIZOL, 2008)

Conjuntos *fuzzy* são definidos por variáveis linguísticas, as quais são definidas por três elementos principais (x , $T(x)$, U), sendo x o nome da variável, por exemplo: rotação, ângulo, altura, etc. $T(x)$ é um grupo de valores linguísticos que definem x . No caso de $T(\text{altura})$, tem-se:

$$T(\text{altura}) = \{ \text{baixo, médio, alto} \}$$

Cada termo em $T(\text{altura})$ é caracterizado por um conjunto *fuzzy* em um universo de discurso dado por $U = [1.50, 2.00]$.

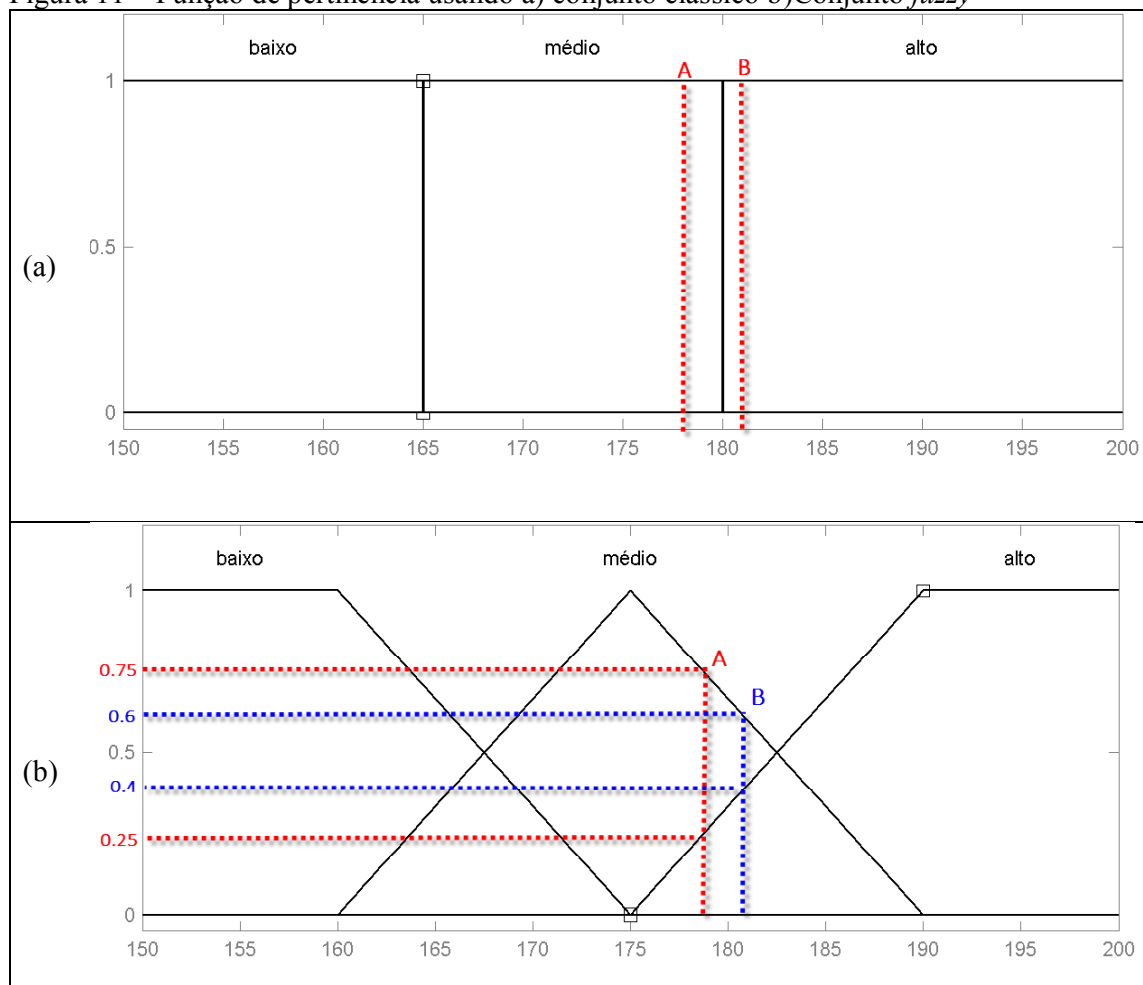
2.4.3 Funções pertinência

Conjuntos *fuzzy* são representados por funções pertinência, que servem para facilitar a notação e avaliação do elemento, uma vez que não possui somente dois valores, pertencendo ou não a um conjunto (por exemplo é alto ou não é alto) (CHEN, 2000).

Um conjunto *fuzzy* A em um universo de discurso X é definido por uma função pertinência $\mu_A(x)$, representado por $\mu_A: X \rightarrow \{0, 1\}$. Onde o valor de $\mu_A(x)$ para o conjunto *fuzzy* A é chamado de grau de pertinência de $x \in X$. A função pertinência representa o grau de participação de x em um conjunto *fuzzy* A (CHEN, 2000).

Por exemplo, tem-se um conjunto *fuzzy* altura dividido em baixo, médio e alto em um universo de discurso variando entre 1,50 m e 2,0 m. Considerando a altura de duas pessoas $A = 1.78\text{m}$ e $B = 1.81\text{m}$, o grau de ativação dos seguintes conjuntos clássico e *fuzzy* são apresentados nas Figura 10 (a) e (b) e na Tabela 1, respectivamente.

Figura 11 – Função de pertinência usando a) conjunto clássico b) Conjunto *fuzzy*



Fonte: Autoria Própria

Tabela 1 – Avaliação das funções pertinência

Pessoa	Altura	Pertinência Clássica			Pertinência <i>Fuzzy</i>		
		Baixo	Médio	Alto	Baixo	Médio	Alto
A	178 cm	0	1	0	0	0.75	0.25
B	181 cm	0	0	1	0	0.6	0.4

Seja por exemplo duas alturas A=178cm e B=181cm conforme apresentado na Tabela 1. Sendo a altura dos indivíduos A e B acima da média.

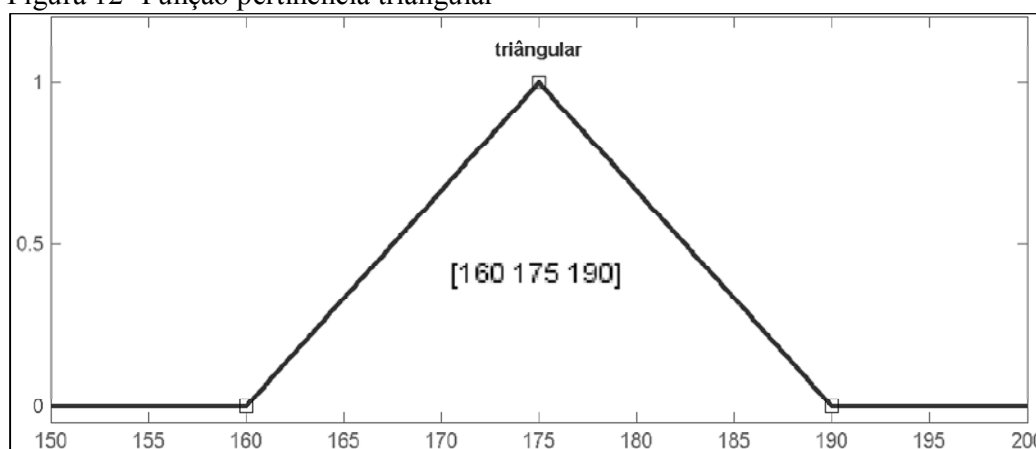
Na lógica clássica o indivíduo A é classificado como médio e o B como alto. Mesmo tendo pouca diferença de altura entre os dois (3 cm).

Na lógica *fuzzy* isso não ocorre, pois estas entradas são sempre representados por grau de pertinência, para A tem grau de pertinência 0,75 para médio e 0,25 para alto e B 0,6 para médio e 0,4 par alto. Representando melhor está diferença entre as alturas.

As funções pertinência podem ter forma triangular, trapezoidal, gaussiana, função S, função Z, sigmoidal e diversas outras. Nesse trabalho utilizou-se as funções de forma triangular e trapezoidal, que são definidas abaixo (CHEN, 2000).

A forma triangular é definida pelos parâmetros $\{a,b,c\}$, referentes a posição de seus vértices. A Figura 12 apresenta uma função pertinência triangular.

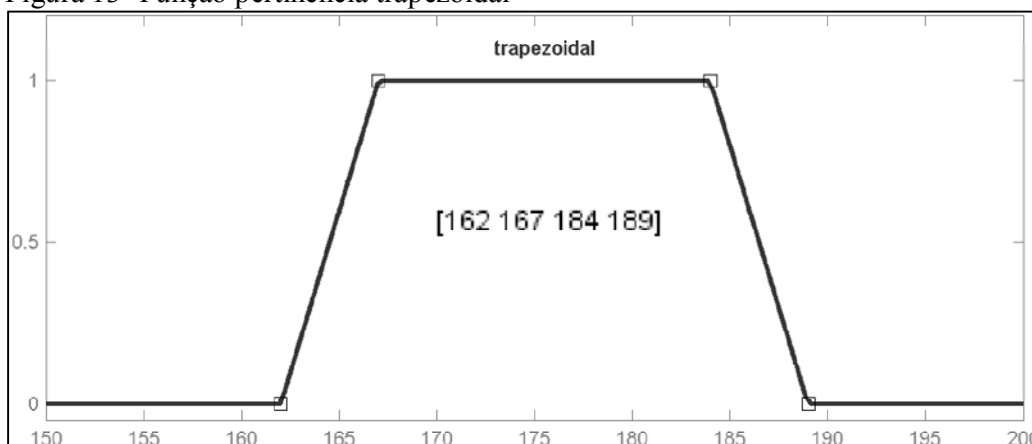
Figura 12- Função pertinência triangular



Fonte: Autoria Própria

A função pertinência trapezoidal é definida pelos parâmetros $\{a,b,c,d\}$, também referentes a posição dos seus vértices da esquerda para a direita, como ilustrado na Figura 13.

Figura 13- Função pertinência trapezoidal



Fonte: Autoria Própria

2.4.4 Operadores Fuzzy

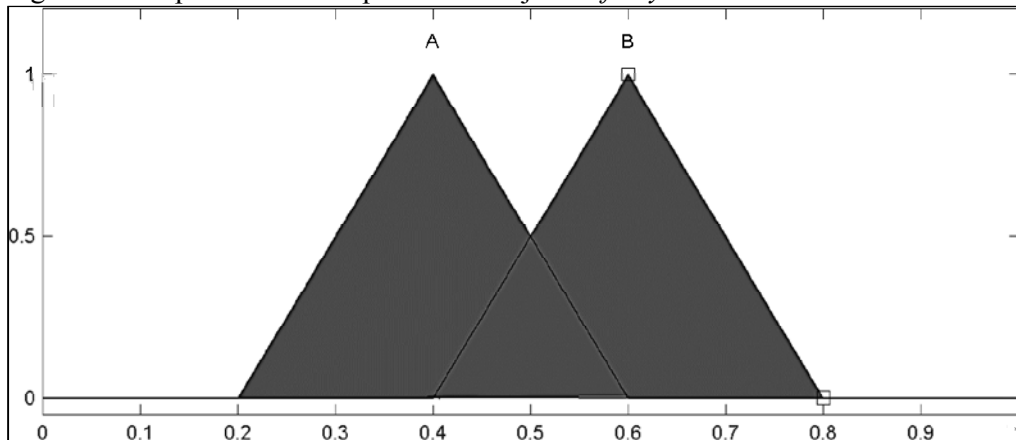
As operações união, intersecção e complemento de conjuntos *fuzzy* podem ser descritos como operações de suas funções pertinência definido a seguir baseando-se no trabalho Rizol (2009).

A união de dois conjuntos *fuzzy* A e B é um conjunto *fuzzy* definido pela função pertinência apresentada na equação (1).

$$\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x) = \begin{cases} \mu_A(x) & \mu_A(x) \geq \mu_B(x) \\ \mu_B(x) & \mu_A(x) < \mu_B(x) \end{cases} = \max\{\mu_A(x), \mu_B(x)\} \quad (1)$$

Podendo também ser expressa graficamente conforme a Figura 14.

Figura 14 - Operador união aplicado a conjuntos fuzzy



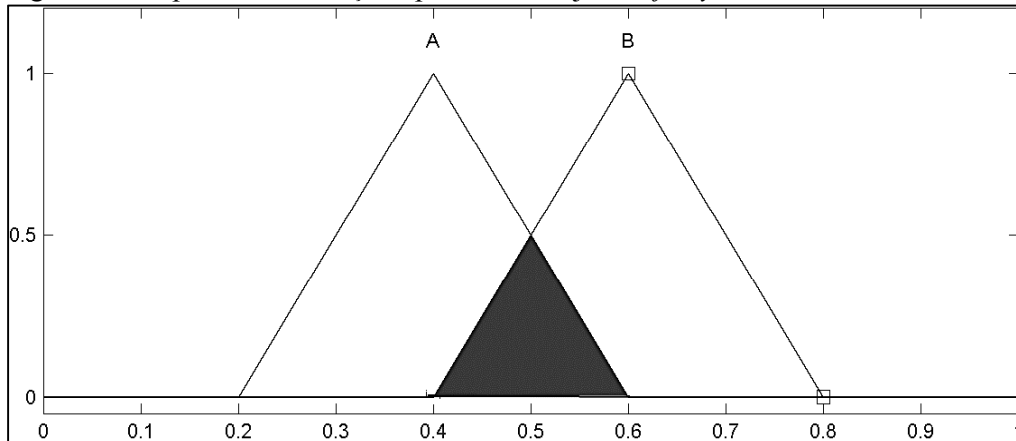
Fonte: Autoria Própria

A intersecção de um conjuntos *fuzzy* A e B é expressa matematicamente pela equação (2).

$$\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x) = \begin{cases} \mu_A(x) & \mu_A(x) \leq \mu_B(x) \\ \mu_B(x) & \mu_A(x) > \mu_B(x) \end{cases} = \min\{\mu_A(x), \mu_B(x)\} \quad (2)$$

Graficamente é definida conforme a Figura 15.

Figura 15 – Operador interseção aplicado a conjuntos *fuzzy*



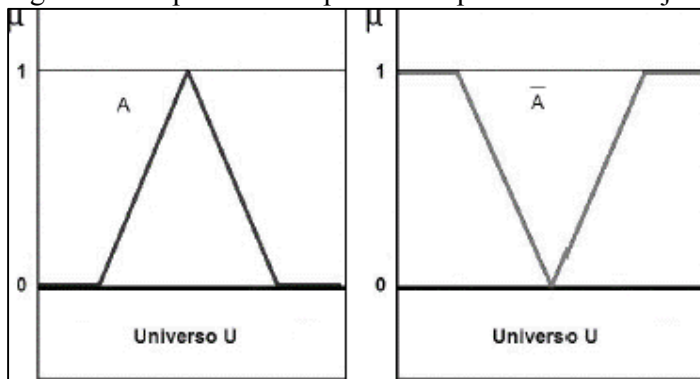
Fonte: Autoria Própria

A operação complemento de um conjunto *fuzzy* A é modelado matematicamente pela equação (3).

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (3)$$

Graficamente é demonstrado conforme a Figura 16.

Figura 16 – Operador complemento aplicado a um conjunto *fuzzy*



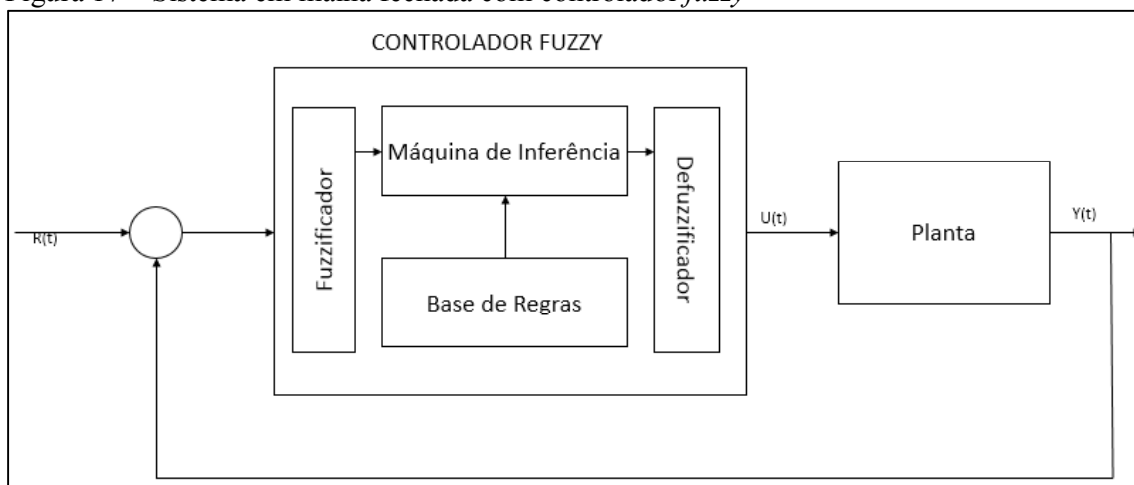
Fonte: (RIZOL, 2008)

Além das operações aqui demonstradas existem outras operações aplicáveis a conjuntos *fuzzy* apresentadas no livro de Klir(1995).

2.4.5 Controlador *Fuzzy*

Como mostrado na Figura 17 o controlador *fuzzy* é composto por 4 sub blocos: Fuzzificador, Máquina de Inferência, Base de Regra e Defuzzificador, que serão apresentados separadamente ao longo deste capítulo (MENDEL, 1995).

Figura 17 – Sistema em malha fechada com controlador *fuzzy*



Fonte: adaptado de (RODRIGUEZ, 2010)

2.4.6 Fuzzificador

A fuzzificação é o processo de transformar um sinal de entrada em um ou mais conjuntos *fuzzy*. Para isso, o fuzzificador identifica a entrada e seu intervalo e o nomeia, tornando-os compatíveis com os conjuntos *fuzzy* do controlador (SIMÕES, 2010).

2.4.7 Base de regras

A base de regras é formada por proposições do tipo se-então definidas pelo especialista a fim de se obter um bom controle, baseando-se no conhecimento do sistema a ser controlado.

Usa-se as variáveis linguísticas para a implementação das regras. Seja por exemplo, um controlador *fuzzy* PD com entradas erro e variação de erro e saída variação da rotação, pode-se definir uma regra com o seguinte formato: Se erro é muito positivo e variação do erro nulo então variação da rotação positiva. De modo que cada regra do controlador assume a forma exemplificada, generalizando pode-se definir a regra como se (condição), então (ação), sendo chamada de antecedente as condições e consequente a ação (ROSS, 2009).

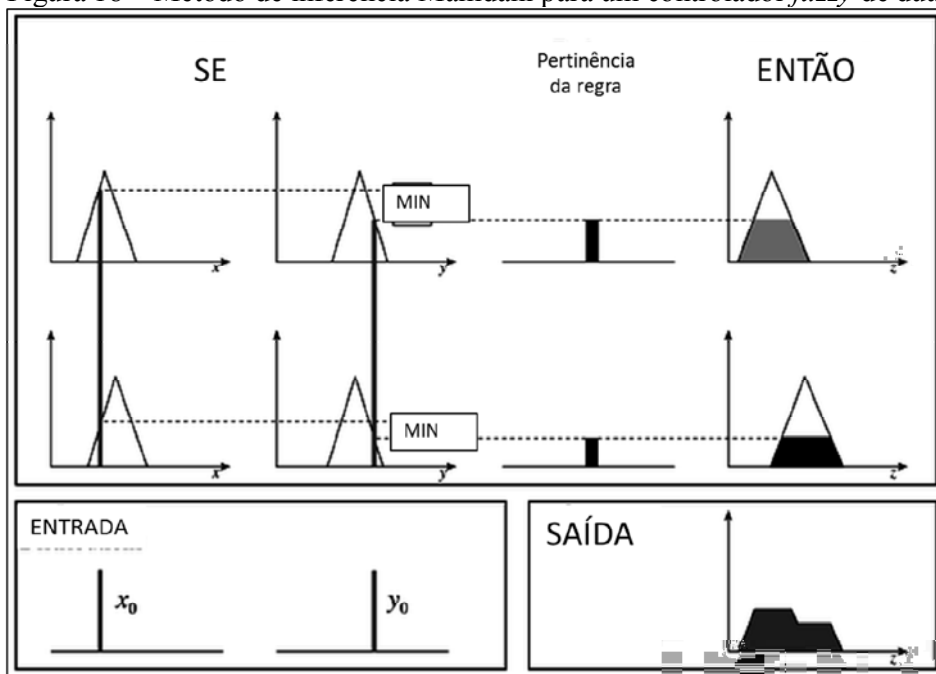
2.4.8 Inferência *fuzzy*

Neste subcapítulo será descrito o método de inferência Mamdani, que será posteriormente utilizado no simulador.

Um sistema de inferência *fuzzy* utiliza os conjuntos *fuzzy* de entrada para mapear a saída associada as entradas. O sistema de inferência Mamdani, também chamado de máximo e mínimo, utiliza os passos apresentados abaixo para calcular o valor de saída baseado nos valores de entrada, sendo os seguintes passos necessários (KNAPP, 2004).

- Determinar o conjunto de regras *fuzzy* ativados pela entrada;
- Combina-se o valor de entrada fuzzificado com as regras para estabelecer a pertinência de cada regra, a seguir, realiza-se o mínimo entre esses graus de pertinências;
- Encontra-se a consequência da regra combinando a pertinência anterior com a função pertinência de saída;
- Combina-se então as consequências realizando o máximo e determina-se a saída;
- Aplica-se o defuzzificador para converter o valor no universo *fuzzy* de volta para o formato necessário a ser aplicado no sistema.

Figura 18 – Método de inferência Mamdani para um controlador *fuzzy* de duas entradas



Fonte: Adaptado de (KNAPP, 2004)

2.4.9 Defuzzificador

Este bloco é responsável pela conexão entre a máquina de inferência e a planta a ser controlada. Este bloco tem como função transformar os valores *fuzzy* do controlador de volta

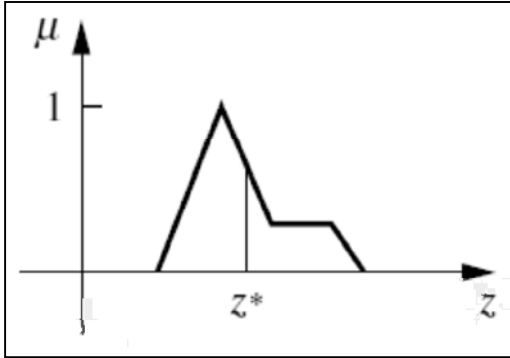
para valores *crisp* para a atuação na planta, podendo ser entendida como o inverso do fuzzificador (SIMOES, 2010).

O método mais conhecido para defuzzificação é o método do centro de área, onde é determinado o centro da área abaixo da função pertinência associada a sua saída, podendo ser descrito matematicamente pela equação (4) (RIZOL, 2008).

$$u^* = \frac{\int_u u \times \mu_U(u) du}{\int_u \mu_U(u) du} \quad (4)$$

Sendo μ_U a união da área das funções pertinência de saída e u é o valor de ativação da respectiva regra armazenada na base de regras, conforme ilustrado na Figura 19 (RIZOL, 2008).

Figura 19 – Método de defuzzificação centro de área



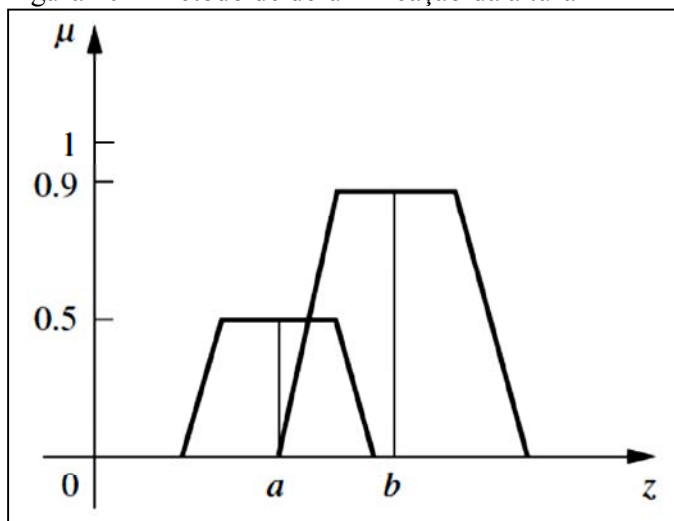
Fonte: (RODRIGUEZ, 2010)

Outro método bastante difundido é o método da Altura, também conhecido como centro dos máximos, este método se baseia na soma ponderada do valor máximo de todas as funções pertinência associadas com os termos das conclusões das regras, podendo então ser calculado pela equação (5) (RIZOL, 2008).

$$u^* = \frac{\sum_{k=1}^m c^{(k)} f_k}{\sum_{k=1}^m f_k} \quad (5)$$

Sendo $c^{(k)}$ a posição onde encontra-se a altura máxima da função pertinência associada a saída, f_k o valor máximo que a função pertinência atinge em um número de funções pertinência, e u^* a saída do controlador *fuzzy*, ilustrado pela Figura 20.

Figura 20 – Método de defuzzificação da altura



Fonte: (RODRIGUEZ, 2010)

Conclui-se então a introdução e inicia-se na próxima seção a modelagem matemática e implementação do simulador.

3 MODELO DO QUADCOPTER

A seguir, será apresentado a modelagem do *quadcopter*, esta etapa é essencial para a simulação e controle, pelo fato de que o conhecimento pleno da dinâmica auxiliará no desenvolvimento do modelo virtual assim como o controlador.

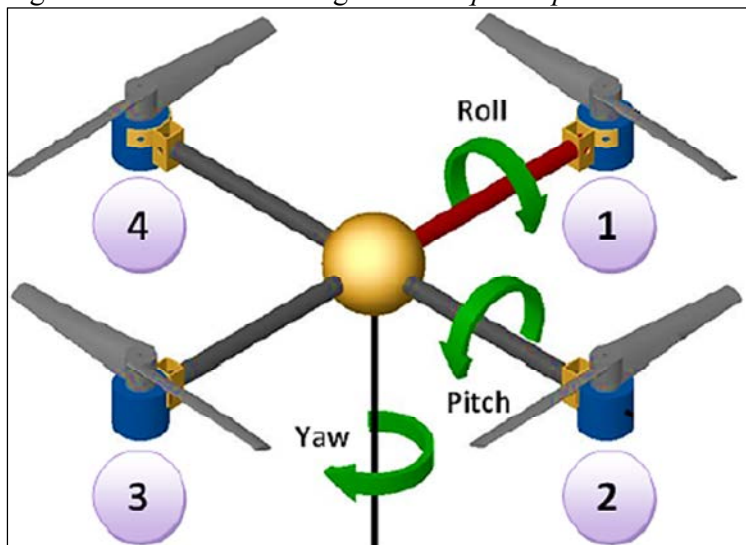
3.1 CONCEITOS DE FUNCIONAMENTO

Os motores do *quadcopter* são distribuídos nas extremidades de um frame em X, sendo que a rotação de cada motor é implementada de forma a eliminar as forças horizontais, facilitando assim o controle do ângulo *yaw*. O conceito de funcionamento consiste em dois pares de motores, cada par girando em sentido contrário ao outro, formado por motores do mesmo eixo, 1 e 3 e o outro 2 e 4 (BOUABDALLAH, 2007).

Desse modo existem duas forças horizontais que se anulam, quando a rotação tem modulo igual para todos motores, ao passo que se todos os motores girassem no mesmo sentido o *quadcopter* giraria ininterruptamente em torno de seu próprio eixo impossibilitando o controle do *yaw* (BOUABDALLAH, 2007).

Pode-se separar os movimentos do *quadcopter* em quatro movimentos fundamentais sendo um translacional correspondente a altura e três rotacionais (*roll*, *pitch* e *yaw*) representados na Figura 21, caracterizando assim a dinâmica do *quadcopter*.

Figura 21 - Movimentos angulares do *quadcopter*



Fonte: (DOMINGUES, 2009)

Para facilitar a nomenclatura ao longo do trabalho adota-se uma variável para cada movimento fundamental sendo as mesmas expressas na Tabela 2.

Tabela 2 – Associação dos movimentos fundamentais a variáveis de força e torque

Movimento fundamental	Força/Torque associado
	U1[N]
Altura	
	U2[N.m]
<i>Roll</i>	
	U3[N.m]
<i>Pitch</i>	
	U4[N.m]
<i>Yaw</i>	

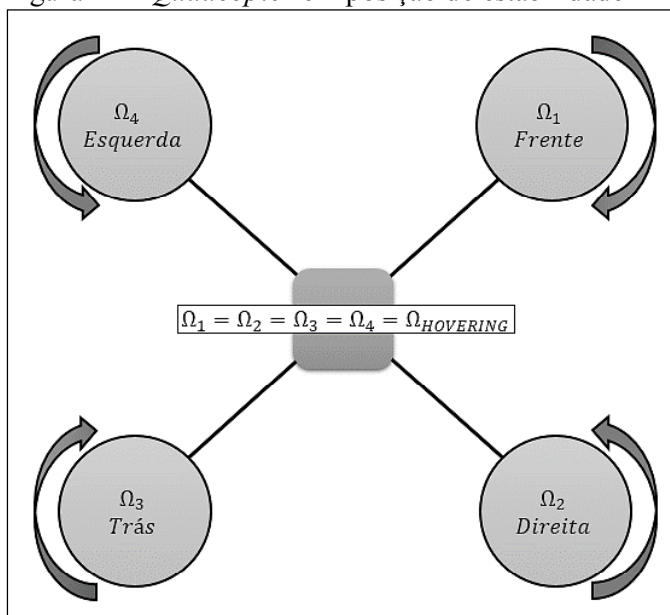
Ao longo das próximas seções os movimentos serão apresentados com maiores detalhes explicitando como devem ser as rotações usadas para obter o movimento desejado

3.1.1 Hovering

Hovering é a situação de repouso do *quadcopter* onde o mesmo se encontra estável no ar, sem efetuar nenhum movimento translacional ou rotacional, ou seja, todas as forças resultantes e torques resultantes são nulos. A velocidade aplicada aos motores fornecem uma força vertical que anula o peso do *quadcopter*. O controle a ser implementado tem como uma de suas funções ser capaz de manter o *quadcopter* em *Hovering* quando não houver entrada do usuário para efetuar algum movimento (NICE, 2004).

Na Figura 21 apresenta-se a estrutura do *quadcopter*, assim como, o sentido com que os motores funcionam, sendo o motor 1 e 3 girantes no sentido horário e 2 e 4 no sentido anti-horário. A cor cinza simboliza que a velocidade é a necessária para manter o *quadcopter* estável e a partir de variações positivas ou negativas em torno dessa velocidade é possível efetuar os movimentos translacionais e rotacionais (NICE, 2004).

Figura 22 – *Quadcopter* em posição de estabilidade



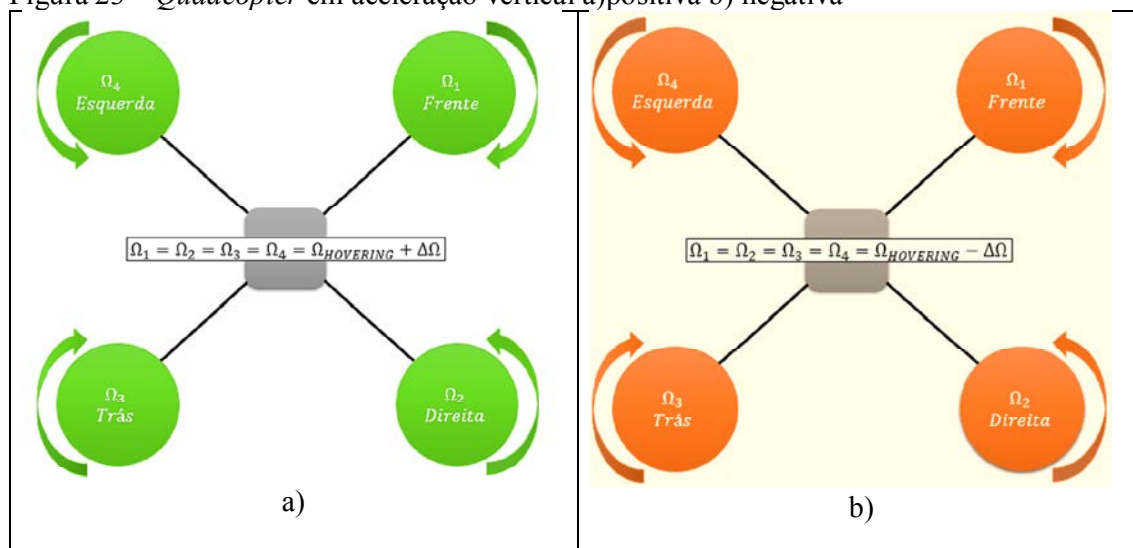
Fonte: Autoria Própria

3.1.2 Altura – $U1[N]$

A aceleração causada por $U1$ é o movimento no eixo Z , a altura é modificada atuando na velocidade de todos os motores igualmente, com isso a resultante será somente uma força vertical, podendo ser positiva ou negativa, dependendo do valor da variação da velocidade de rotação dos motores (SØRENSEN, 2010).

Na Figura 23, apresenta-se as duas situações de aceleração, a positiva representada pela Figura 23 a) e a negativa pela Figura 23b).

Figura 23 – *Quadcopter* em aceleração vertical a) positiva b) negativa

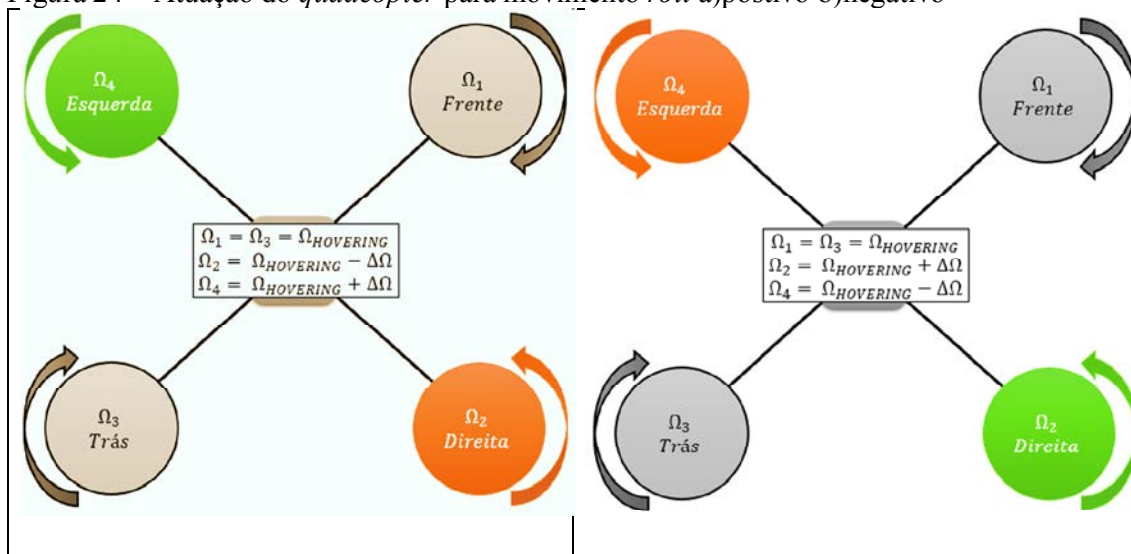


Fonte: Autoria Própria

3.1.3 *Roll* – U2[N.m]

Para atuar no ângulo de *roll*, mantém-se a velocidade angular dos motores um e três, e aplica-se uma variação de rotação definida pelo controlador nos motores quatro e dois, aumentando a velocidade de um dos motores e diminuindo a do outro, de forma que o *quadcopter* gire em torno do eixo formado pelos motores um e três, como apresentado na Figura 24. O movimento de *roll* considerado positivo corresponde ao aumento da velocidade do motor quatro e diminuição do motor dois, e o negativo a diminuição do motor quatro e aumento do motor dois, em ambos os casos mantém-se a velocidade dos motores um e três (SØRENSEN, 2010).

Figura 24 – Atuação do *quadcopter* para movimento *roll* a) positivo b) negativo

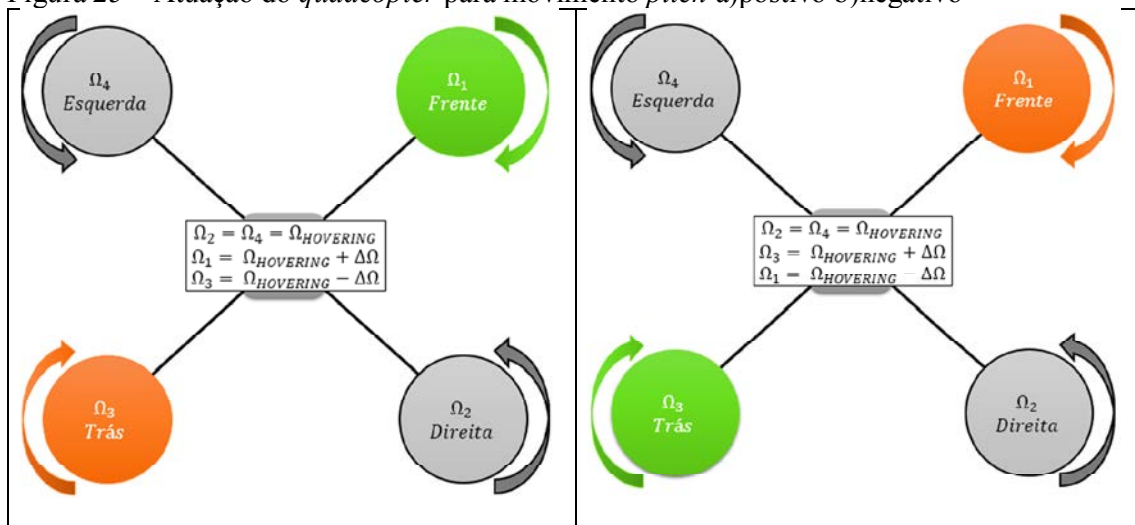


Fonte: Autoria Própria

3.1.4 Pitch – U3[N.m]

O movimento *pitch* é similar ao *roll*, porém mantém-se constante a velocidade dos motores quarto e dois e varia-se dos motores um e três, para obter o torque U3 positivo deve-se atuar de forma que a velocidade angular do motor um seja maior que a do motor três, e para obter-se torque U3 de valor negativo a velocidade do motor um tem que ser menor do que a do motor três, como demonstrada na Figura 25 (SØRENSEN, 2010).

Figura 25 – Atuação do *quadcopter* para movimento *pitch* a) positivo b) negativo

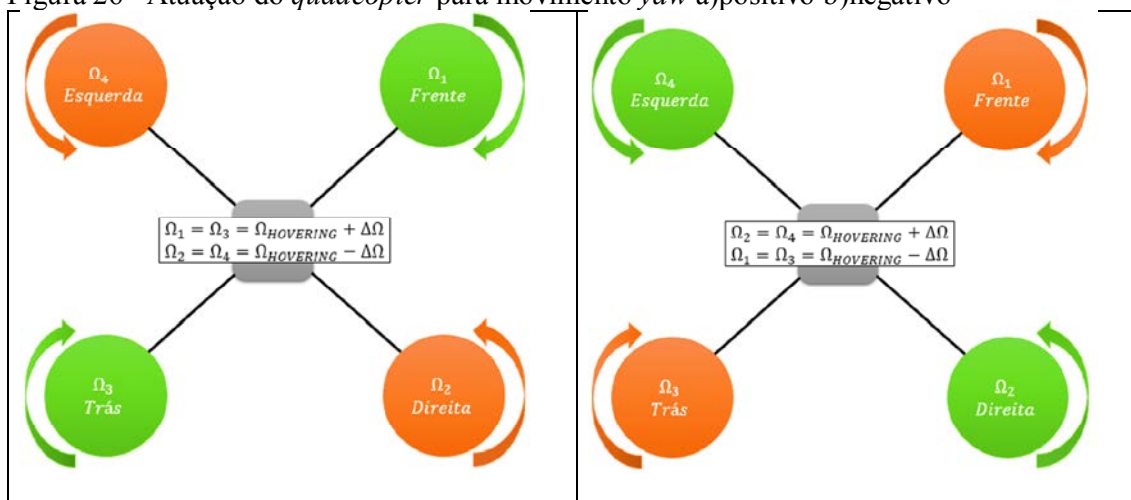


Fonte: Autoria Própria

3.1.5 *Yaw* – U4[N.m]

O torque U4 responsável por controlar o movimento de *Yaw*. Este torque é controlado fazendo com que as velocidades dos motores que giram no sentido horário seja diferente que dos motores de sentido anti-horário. Desta forma existe uma força horizontal resultante, e consequentemente um torque capaz de girar o *quadcopter* em torno do seu eixo em sentido horário quando a velocidade dos motores um e três são maiores do que a velocidade dos motores quatro e dois, e anti-horário quando a velocidade do par de motores quatro e dois é maior do que a do par um e três, como apresentado na Figura 26 (SØRENSEN, 2010).

Figura 26– Atuação do *quadcopter* para movimento yaw a) positivo b) negativo



Fonte: Autoria Própria

Resumidamente podemos inferir o movimento feito pelo *quadcopter* de acordo com a Tabela 3.

Tabela 3 – Visão geral da dinâmica do *quadcopter*

Motor 1	Motor 2	Motor 3	Motor 4	Movimento
$\Omega_{HOVERING}$	$\Omega_{HOVERING}$	$\Omega_{HOVERING}$	$\Omega_{HOVERING}$	<i>Hovering</i>
$\Omega_{HOVERING} + \Delta\Omega$	$\Omega_{HOVERING} + \Delta\Omega$	$\Omega_{HOVERING} + \Delta\Omega$	$\Omega_{HOVERING} + \Delta\Omega$	<i>+Altura</i>
$\Omega_{HOVERING} - \Delta\Omega$	$\Omega_{HOVERING} - \Delta\Omega$	$\Omega_{HOVERING} - \Delta\Omega$	$\Omega_{HOVERING} - \Delta\Omega$	<i>-Altura</i>
$\Omega_{HOVERING}$	$\Omega_{HOVERING} - \Delta\Omega$	$\Omega_{HOVERING}$	$\Omega_{HOVERING} + \Delta\Omega$	<i>+Roll</i>
$\Omega_{HOVERING}$	$\Omega_{HOVERING} + \Delta\Omega$	$\Omega_{HOVERING}$	$\Omega_{HOVERING} - \Delta\Omega$	<i>-Roll</i>
$\Omega_{HOVERING} - \Delta\Omega$	$\Omega_{HOVERING}$	$\Omega_{HOVERING} + \Delta\Omega$	$\Omega_{HOVERING}$	<i>+Pitch</i>
$\Omega_{HOVERING} + \Delta\Omega$	$\Omega_{HOVERING}$	$\Omega_{HOVERING} - \Delta\Omega$	$\Omega_{HOVERING}$	<i>-Pitch</i>
$\Omega_{HOVERING} - \Delta\Omega$	$\Omega_{HOVERING} + \Delta\Omega$	$\Omega_{HOVERING} - \Delta\Omega$	$\Omega_{HOVERING} + \Delta\Omega$	<i>+Yaw</i>
$\Omega_{HOVERING} + \Delta\Omega$	$\Omega_{HOVERING} - \Delta\Omega$	$\Omega_{HOVERING} + \Delta\Omega$	$\Omega_{HOVERING} - \Delta\Omega$	<i>-Yaw</i>

3.2 MODELO DE NEWTON EULER

A modelagem inicia-se com objetivo de identificar as equações básicas que descrevem a cinemática e dinâmica de um corpo rígido com 6 graus de liberdade utilizando o formalismo Newton-Euler, esta seção foi baseada nos trabalhos de Bresciani (2008), de Rodrigues(2008), de Sousa (2011) e no livro do Ardema(2005).

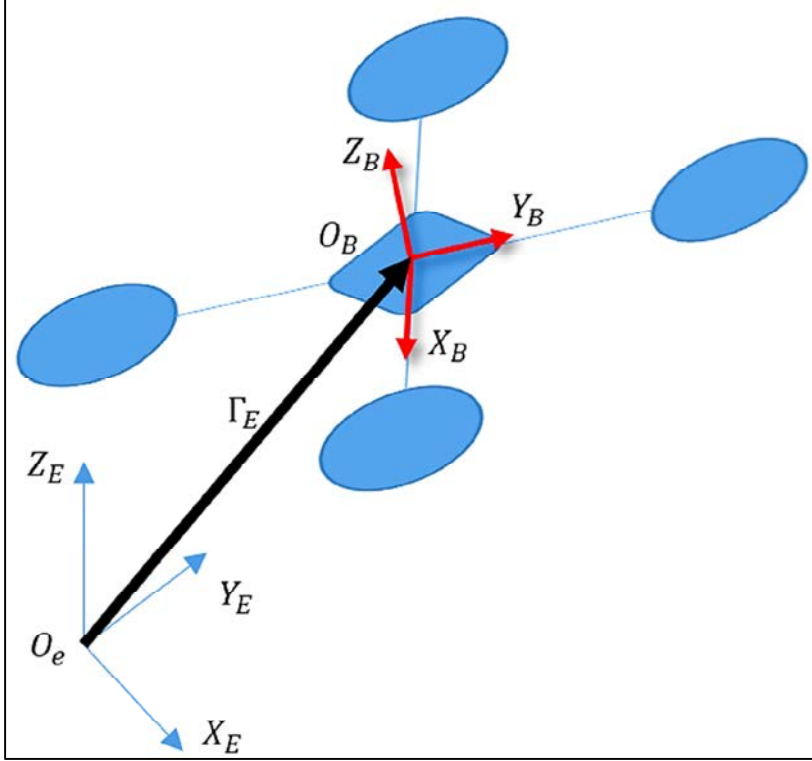
A abordagem inicial se dá pelo estudo da cinemática, o qual descreve o movimento de um corpo sem considerar as forças e torques que atuam sobre o corpo, para isso define-se dois sistemas de coordenadas, sendo um deles fixado a terra. Sendo denominado sistema de coordenadas E ($o_E x_E y_E z_E$), servindo de referência inercial, onde x_E aponta para o norte, y_E para o oeste, z_E para cima, ou seja saindo da terra, o_E é a origem desse sistema de coordenadas. O sistema de coordenadas E fixado a terra será utilizado para determinar as posições instantâneas do *quadcopter* definidas por $\Gamma^E[m]$ que representa a posição translacional e $\Theta^E[rad]$ a posição rotacional.

O segundo sistema de coordenadas tem como origem o_B que se localiza no centro de massa do *quadcopter*, x_B é direcionado para a frente do *quadcopter*, y_B aponta para a esquerda, tomando como base a frente que foi anteriormente definida, z_B aponta para cima. Este sistema de coordenadas será nomeado como B($o_B x_B y_B z_B$) e será posteriormente utilizado para determinar a velocidade translacional $V^B[m.s^{-1}]$, a velocidade rotacional $\omega^B[rad.s^{-1}]$, as forças relacionadas ao corpo de quadcopter $F^B[N]$ e também os torques $\tau^B[N.m]$.

A posição linear em relação a um ponto dado da terra é determinado pelo vetor com origem no centro do sistema de coordenada E até o centro do sistema B, sendo definida pela equação (6) e ilustrada pela Figura 27.

$$\Gamma^E = [X \ Y \ Z]^T \quad (6)$$

Figura 27 – Representação da posição do *quadcopter* em relação a terra



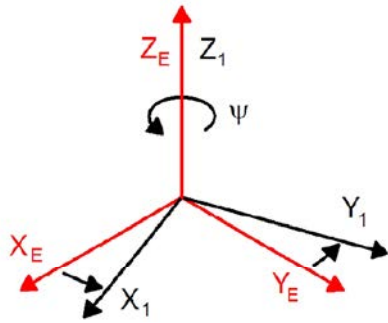
Fonte: Autoria Própria

A posição angular Θ^E do *quadcopter* é calculada como a orientação do sistema de coordenadas B em relação ao sistema E, denominando os ângulos como os movimentos angulares fundamentais *roll*, *pitch* e *yaw* como ϕ , θ e ψ respectivamente, como mostrado na equação (7).

$$\Theta^E = [\phi \ \theta \ \psi]^T \quad (7)$$

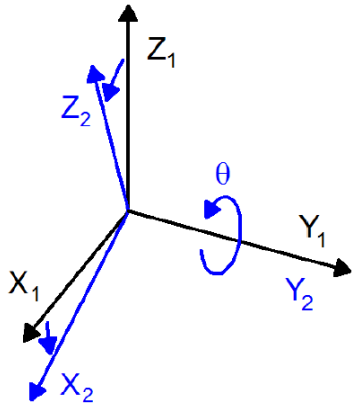
A matriz de rotação R_Θ é obtida pela multiplicação das três matrizes de rotação básicas, considerando a notação $\cos(x) = c_x$ e $\sin(x) = s_x$, tem-se então:

- Rotação em torno do eixo z_E dada pelo ângulo ψ , definida pela equação (8).



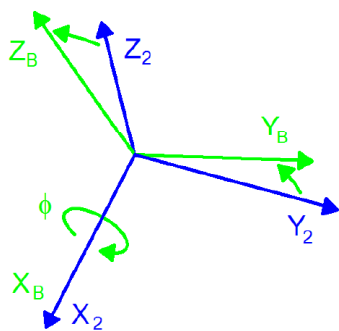
$$R(\psi, z) = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

• Rotação em torno do eixo y_1 dada pelo ângulo θ , definida pela equação (9).



$$R(\theta, y) = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \quad (9)$$

• Rotação em torno do eixo x_2 dada pelo ângulo ϕ , definida pela equação (10).



$$R(\phi, x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} \quad (10)$$

E então finalmente tem-se a matriz de rotação R_Θ dada pela equação (11).

$$R_\Theta = R(\psi, z) R(\theta, y) R(\phi, x) = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\theta + c_\psi s_\theta s_\phi & s_\psi s_\theta + c_\psi s_\theta c_\phi \\ s_\psi c_\theta & c_\psi c_\theta + s_\psi s_\theta s_\phi & -c_\psi s_\theta + s_\psi s_\theta c_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \quad (11)$$

As equações (12) e (13) apresentam a velocidade linear e angular respectivamente, definindo as suas componentes tridimensionais.

$$V^B = [u \ v \ w]^T \quad (12)$$

$$\omega^B = [p \ q \ r]^T \quad (13)$$

Agrega-se as componentes lineares e angulares, com objetivo de representar de forma completa a posição e velocidade do *quadcopter* no espaço, para isso, define-se ξ como vetor posição em relação a terra e ν o vetor velocidade em relação ao centro de massa do *quadcopter*, os vetores são apresentados pelas equações (14) e (15) respectivamente.

$$\xi = [\Gamma^E \ \Theta^E]^T = [X \ Y \ Z \ \phi \ \theta \ \psi]^T \quad (14)$$

$$\nu = [V^B \ \omega^B]^T = [u \ v \ w \ p \ q \ r]^T \quad (15)$$

Utilizando a matriz rotacional R_Θ pode-se definir a velocidade linear em relação a terra multiplicando-a pela velocidade linear definida para o sistema de coordenadas B como na equação (16), pois a matriz rotacional implementa a mudança do sistema de coordenadas.

$$V^E = \dot{\Gamma}^E = R_\Theta V^B \quad (16)$$

Conforme apresentado a relação entre as velocidades dos dois sistemas, o mesmo ocorre para as velocidades angulares, utilizando uma matriz transferência T_Θ para efetuar a mudança do sistema de coordenadas, como nas equações (17) e (18).

$$\omega^B = T_\Theta^{-1} \dot{\Theta}^E \quad (17)$$

$$\dot{\Theta}^E = T_\Theta \omega^B \quad (18)$$

A matriz T_Θ é determinada por meio da equação (19), referente a velocidade angular em relação ao centro de massa do *quadcopter*, obtendo então T_Θ^{-1} e T_θ nas equações (20) e (21) respectivamente.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R(\phi, x)^{-1} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R(\phi, x)^{-1} R(\theta, y)^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} = T_\theta^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (19)$$

$$T_{\Theta}^{-1} = \begin{bmatrix} 1 & 0 & -s_{\theta} \\ 0 & c_{\phi} & c_{\theta}s_{\phi} \\ 0 & -s_{\phi} & c_{\theta}c_{\phi} \end{bmatrix} \quad (20)$$

$$T_{\theta} = \begin{bmatrix} 1 & s_{\phi}t_{\theta} & c_{\phi}t_{\theta} \\ 0 & c_{\phi} & -s_{\phi} \\ 0 & s_{\phi}/c_{\theta} & c_{\phi}/c_{\theta} \end{bmatrix} \quad (21)$$

Generalizando, define-se a derivada da posição em relação a terra pela equação (22), onde J_{Θ} é uma matriz transferência que contem R_{Θ} e T_{Θ} na forma da equação (23), assim transformando v para o sistema de coordenadas em relação a terra.

$$\dot{\xi} = J_{\Theta} v \quad (22)$$

$$J_{\Theta} = \begin{bmatrix} R_{\Theta} & 0_{3 \times 3} \\ 0_{3 \times 3} & T_{\Theta} \end{bmatrix} \quad (23)$$

Termina-se assim então a modelagem dos parâmetros cinemáticos, passando então para uma abordagem em relação a dinâmica do corpo, considerando as de forças e torques.

Usa-se o sistema de coordenadas B devido a possibilidade de simplificação no equacionamento por conta de ser um corpo simétrico, as medidas e o controle serão também feitos em relação ao corpo e não a terra.

Presume-se para o seguinte equacionamento que a origem do sistema de coordenadas B coincide com o centro de massa do corpo e os eixos do sistema de coordenadas coincidem com os braços do *quadcopter*, tornando a matriz do momento de inercia diagonal.

A segunda lei de Newton apresentada na equação (24) pode ser trabalhada de modo a expressar a força no corpo em função da massa $m[\text{kg}]$ e velocidades linear e angular, como demonstrado nas equações (25), (26) e (27), utilizando operadores derivativos para expandir a aceleração linear inicialmente equacionada.

$$m\ddot{\Gamma}^E = F^E \quad (24)$$

$$m(R_{\Theta}\dot{V}^B + \dot{R}_{\Theta}V^B) = R_{\Theta}F^B \quad (25)$$

$$m(\dot{V}^B + \omega^B \times V^B) = R_{\Theta}F^B \quad (26)$$

$$m(\dot{V}^B + \omega^B \times V^B) = F^B \quad (27)$$

Correspondentemente, aplica-se o mesmo princípio para as componente angulares, que por sua vez geram torques iniciando o equacionamento com a equação (28), obtendo como resultado a equação (29) onde há relação entre o torque no corpo com a sua matriz inercial e velocidade e posição angular em relação ao seu centro de massa.

$$I\ddot{\Theta}^E = \tau^E \quad (28)$$

$$I\dot{\omega}^B + \dot{\omega}^B \times (I\omega^B) = T_{\Theta}\tau^B \quad (29)$$

Finalizando o modelo de um corpo com 6 graus de liberdade colocando as equações (27) e (28) em forma matricial na equação (30), onde $I_{3 \times 3}$ representa uma matriz identidade de ordem 3, e I é a matriz inercial.

A equação (30) é a forma genérica da modelagem newton-euler para um corpo rígido de seis graus de liberdade.

$$\begin{bmatrix} mI_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{bmatrix} \begin{bmatrix} \dot{V}^B \\ \dot{\omega}^B \end{bmatrix} + \begin{bmatrix} \omega^B \times (mV^B) \\ \omega^B \times (I\omega^B) \end{bmatrix} = \begin{bmatrix} F^B \\ \tau^B \end{bmatrix} \quad (30)$$

Define-se um vetor Λ pela equação (31), o qual contém as forças e torques presentes no corpo do *quadcopter*.

$$\Lambda = [F^B \tau^B]^T = [F_x \ F_y \ F_z \ \tau_x \ \tau_y \ \tau_z]^T \quad (31)$$

Pode-se, então reescrever a equação (30) na forma da equação (32), onde $\dot{v}$ é a aceleração translacional no sistemas de coordenadas B, onde o centro coincide com o centro de massa do *quadcopter*.

$$M_B \dot{v} + C_B(v)v = \Lambda \quad (32)$$

A matriz que representa o sistema inercial do corpo é definido por M_B na equação (33), as suposições iniciais simplifica o sistema de tal forma que obtém-se M_B que é uma matriz com valores constantes não nulos na sua diagonal referente a sua massa e constantes de inercia e todos os outros elementos nulos.

$$M_B = \begin{bmatrix} mI_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{XX} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{YY} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{ZZ} \end{bmatrix} \quad (33)$$

A matriz C_B apresentada na equação (34) é referente as forças e torques centrípetos.

$$C_B(v) = \begin{bmatrix} 0_{3 \times 3} & -m S(V^B) \\ 0_{3 \times 3} & -S(I \omega^B) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & m\omega & -mv \\ 0 & 0 & 0 & -m\omega & 0 & mu \\ 0 & 0 & 0 & -mv & -mu & 0 \\ 0 & 0 & 0 & 0 & I_{ZZ}r & -I_{YY}q \\ 0 & 0 & 0 & -I_{ZZ}r & 0 & I_{XX}p \\ 0 & 0 & 0 & I_{YY}q & -I_{XX}p & 0 \end{bmatrix} \quad (34)$$

Define-se $S(k)$ na equação (35) como uma matriz antissimétrica de operador k definido pela equação (36).

$$S(k) = -S^T(k) = \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix} \quad (35)$$

$$k = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} \quad (36)$$

Inicia-se então a abordagem para o caso específico do *quadcopter*, divide-se então as forças e torques presentes no corpo em três partes: a contribuição gravitacional, onde será levado em conta a aceleração devido à gravidade, os efeitos giroscópicos produzidos pelas hélices e por fim as forças e torques diretamente aplicados ao *quadcopter* por meio dos atuadores, ou seja, motores.

O vetor G_B é definido pela contribuição da aceleração da gravidade $g[ms^{-2}]$, pelo fato de ser uma força e não um torque só afetará as componentes translacionais, a equação (37) mostra o desenvolvimento do vetor G_B .

$$G_B(\xi) = \begin{bmatrix} F_G^B \\ 0_{3 \times 1} \end{bmatrix} = \begin{bmatrix} R_\Theta^{-1} F_G^E \\ 0_{3 \times 1} \end{bmatrix} = \begin{bmatrix} R_\Theta^T \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \\ 0_{3 \times 1} \end{bmatrix} = \begin{bmatrix} mgs_\theta \\ -mgc_\theta s_\phi \\ -mgc_\theta s_\phi \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (37)$$

A segunda componente a ser tratada é a causada pelo efeito giroscópico das hélices, pelo fato de um par de hélice girar no sentido horário e o outro par no sentido anti-horário haverá

um desbalanceamento quando as velocidades dos motores não forem iguais, causando então o efeito giroscópico representado na equação (38), onde $J_{TP}[Nms^2]$ é o momento de inercia rotacional total em torno do eixo da hélice, sabe-se então que esse efeito só gera torques atuando somente nas componentes angulares.

$$\begin{aligned}
 O_B(v)\Omega &= \left[-\sum_{k=1}^4 J_{TP} \left(\omega^B \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) (-1)^k \Omega_k \right] = \begin{bmatrix} 0_{3 \times 1} \\ J_{TP} \begin{bmatrix} -q \\ p \\ 0 \end{bmatrix} \Omega \end{bmatrix} \\
 &= J_{TP} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ q & -q & q & -q \\ -p & p & -p & p \\ 0 & 0 & 0 & 0 \end{bmatrix} \Omega
 \end{aligned} \tag{38}$$

A velocidade resultante, ou seja, a soma das rotações dos quatro motores é dado por $\Omega[\text{rads}^{-1}]$, apresentado na equação (39), e Ω é um vetor constituído pelas velocidades de cada motor, como na equação (40).

$$\Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \tag{39}$$

$$\Omega = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} \tag{40}$$

A terceira e última componente a ser avaliada é a contribuição das forças e torques diretamente aplicadas ao *quadcopter* para efetuar uma ação desejada, é considerado que as forças e torques são proporcionais a soma dos quadrados das velocidades angulares dos motores envolvidos no movimento dado em radianos por segundo.

As forças na direção X e Y do *quadcopter* são nulas devido a implementação dos motores de forma que um par gira no sentido horário e o outro no sentido anti-horário eliminando assim as componentes X e Y.

Conforme apresentado anteriormente, $U1[N]$ representa a força no eixo Z do sistema de coordenadas B, dado pela soma dos módulos das velocidades multiplicado pela constante de empuxo $b[Ns^2]$, sendo essa uma característica aerodinâmica da hélice.

Os torques $U2[Nm]$, $U3[Nm]$ e $U4[Nm]$ são relacionados aos movimentos de *roll*, *pitch* e *yaw* respectivamente.

A constante l [m] é a distância do centro do *quadcopter* até o centro da hélice e $d[Nms^2]$ é o coeficiente de arrasto da hélice, a equação (41) apresenta de forma matricial as relações expostas.

$$U_B(\Omega) = E_B \Omega^2 = \begin{bmatrix} 0 \\ 0 \\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ bl(\Omega_4^2 - \Omega_2^2) \\ bl(\Omega_3^2 - \Omega_1^2) \\ d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{bmatrix} \quad (41)$$

A matriz E_B , apresentada na equação (42), constitui-se de constantes aerodinâmicas que multiplicam as velocidades angulares dos motores elevados ao quadrado.

$$E_B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ b & b & b & b \\ 0 & -bl & 0 & bl \\ -bl & 0 & -d & 0 \\ -d & d & -d & d \end{bmatrix} \quad (42)$$

Com isso, pode-se igualar a equação (32) com a somas das três componentes discutidas, obtendo assim a equação (43).

$$M_B \dot{v} + C_B(v)v = G_B(\xi) + O_B(v)\Omega + E_B \Omega^2 \quad (43)$$

Ao isolar-se a derivada da velocidade no sistema coordenadas B, denominado como $\dot{v}$, conforme a equação (44).

$$\dot{v} = M_B^{-1}(-C_B(v)v + G_B(\xi) + O_B(v)\Omega + E_B \Omega^2) \quad (44)$$

Abrindo-se então a forma matricial em um sistema de equações mostrado na equação (45) e (46).

$$\begin{cases} \dot{u} = (vr - wq) + gs_\theta \\ \dot{v} = (wp - ur) - gc_\theta s_\phi \\ \dot{w} = (uq - vp) - gc_\theta s_\phi + \frac{U_1}{m} \\ \dot{p} = \frac{I_{YY} - I_{ZZ}}{I_{XX}} qr - \frac{J_{TP}}{I_{XX}} q\Omega + \frac{U_2}{I_{XX}} \\ \dot{q} = \frac{I_{ZZ} - I_{XX}}{I_{YY}} pr + \frac{J_{TP}}{I_{YY}} p\Omega + \frac{U_3}{I_{YY}} \\ \dot{r} = \frac{I_{XX} - I_{YY}}{I_{ZZ}} pq + \frac{U_4}{I_{ZZ}} \end{cases} \quad (45)$$

$$\begin{cases} U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = lb(-\Omega_2^2 + \Omega_4^2) \\ U_3 = lb(-\Omega_1^2 + \Omega_3^2) \\ U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ \Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \end{cases} \quad (46)$$

A dinâmica do *quadcopter* é descrita pela equação (45), porém é muito útil para a implementação do controle da dinâmica se a dinâmica for expressa de forma que as componentes lineares sejam relativas ao sistema de coordenadas E as rotacionais em B, como apresentado na equação (47).

$$\zeta = [\dot{r}^E \quad \omega^B]^T = [\dot{X} \quad \dot{Y} \quad \dot{Z} \quad p \quad q \quad r]^T \quad (47)$$

Chamando então esse novo sistema de coordenadas de H, tem-se que a equação (43) pode ser escrita na forma da equação (48).

$$M_H \dot{\zeta} + C_H(\zeta)\zeta = G_H + O_H(\zeta)\Omega + E_H(\xi)\Omega^2 \quad (48)$$

Onde $\dot{\zeta}$ é o vetor de aceleração no sistema de coordenadas H, define-se que M_H é igual a M_B de acordo com a equação (49) e C_H definido pela equação (50).

$$M_H = M_B = \begin{bmatrix} mI_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{XX} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{YY} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{ZZ} \end{bmatrix} \quad (49)$$

$$C_H(\zeta) = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & -S(I \omega^B) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{ZZ}r & -I_{YY}q \\ 0 & 0 & 0 & -I_{ZZ}r & 0 & I_{XX}p \\ 0 & 0 & 0 & I_{YY}q & -I_{XX}p & 0 \end{bmatrix} \quad (50)$$

O vetor gravitacional G_H agora afeta somente o componente do eixo Z, diferentemente do mesmo vetor no sistema de coordenadas B, define-se então a equação (51).

$$G_H = \begin{bmatrix} F_G^E \\ 0_{3 \times 1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (51)$$

A componente gerada pelo efeito giroscópico se mantém a mesma do sistema de coordenadas B, pois afeta somente as componentes angulares que se mantem no mesmo sistema de coordenadas, conforme a equação (52).

$$O_H(\zeta)\Omega = O_B(v)\Omega = \begin{bmatrix} 0_{3 \times 1} \\ J_{TP} \begin{bmatrix} -q \\ p \\ 0 \end{bmatrix} \end{bmatrix} \Omega = J_{TP} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ q & -q & q & -q \\ -p & p & -p & p \\ 0 & 0 & 0 & 0 \end{bmatrix} \Omega \quad (52)$$

A matriz de movimento denominada E_H difere de E_B pois agora a força U_1 afeta as três componentes translacionais resultando na equação (53).

$$E_H(\xi)\Omega^2 = \begin{bmatrix} R_\Theta & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} E_B\Omega^2 = \begin{bmatrix} (s_\psi c_\phi + c_\psi s_\theta c_\phi)U_1 \\ (-c_\psi s_\phi + s_\psi s_\theta c_\phi)U_2 \\ (c_\theta c_\phi)U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \quad (53)$$

Organizando-se as equações acima pode-se descrever a dinâmica do *quadcopter* no sistema de coordenadas H na forma da equação (54).

$$\dot{\zeta} = M_H^{-1}(-C_H(\zeta)\zeta + G_H + O_H(\zeta)\Omega + E_H(\xi)\Omega^2) \quad (54)$$

Finalmente, obtém-se a equação (55) que representa o sistema de equações obtidos da equação matricial (53).

$$\left\{ \begin{array}{l} \ddot{X} = (s_\psi s_\phi + c_\psi s_\theta c_\phi) \frac{U_1}{m} \\ \ddot{Y} = (-c_\psi s_\phi + s_\psi s_\theta c_\phi) \frac{U_1}{m} \\ \ddot{Z} = -g + (c_\theta c_\phi) \frac{U_1}{m} \\ \dot{p} = \frac{I_{YY} - I_{ZZ}}{I_{XX}} qr - \frac{J_{TP}}{I_{XX}} q\Omega + \frac{U_2}{I_{XX}} \\ \dot{q} = \frac{I_{ZZ} - I_{XX}}{I_{YY}} pr + \frac{J_{TP}}{I_{YY}} p\Omega + \frac{U_3}{I_{YY}} \\ \dot{r} = \frac{I_{XX} - I_{YY}}{I_{ZZ}} pq + \frac{U_4}{I_{ZZ}} \end{array} \right. \quad (55)$$

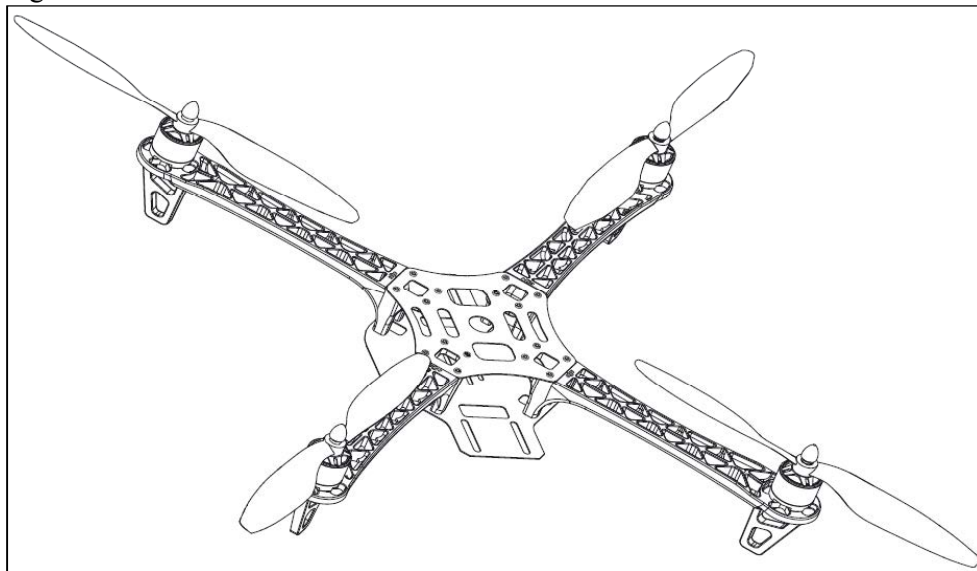
3.3 ARQUITETURA DO PROTÓTIPO

Um *quadcopter* necessita de uma série de componentes para um protótipo ser implementado, tendo funções como comunicação, alimentação, controle, sensoramento e atuação. Nas seções seguintes serão apresentadas os componentes fundamentais para a construção de um protótipo de um *quadcopter* (CANETTA, 2007).

3.3.1 Frame

Estrutura que dá rigidez ao corpo do protótipo, possui quatro braços e duas peças metálicas centrais, onde junta-se os quatro braços, como na Figura 28. No frame são efetuadas as ligações elétricas entre motores e controladores. Este deve ser de material que absorva eficientemente vibrações e também seja relativamente resistente, levando em conta também o seu peso e tamanho, sendo esses fatores os que definem a sua aplicação (DJI, 2013).

Figura 28 – Desenho técnico do frame DJI450

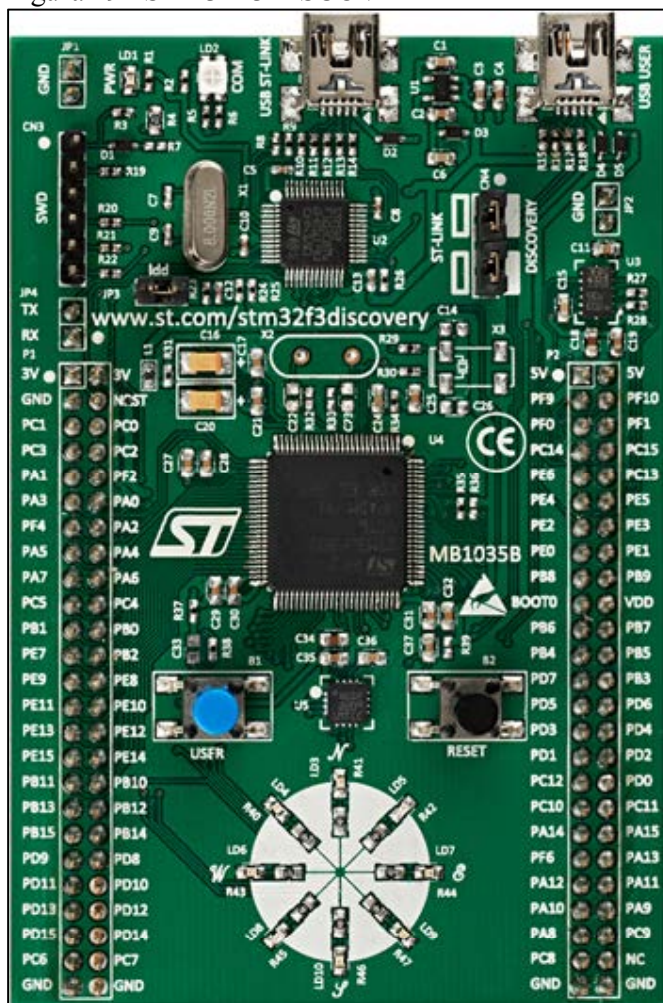


Fonte: (DJI, 2013)

3.3.2 Microcontrolador

É responsável pela interface entre os sensores e o transmissor receptor que envia e recebe dados de uma estação remota que efetua o cálculo do controle e envia para o microcontrolador o valor PWM a ser aplicado em cada controlador de motor. Algumas placas de desenvolvimento já possuem os sensores necessários para a estimação da posição. Este é o caso do STM32F3DISCOVERY apresentado na Figura 29, esta placa de desenvolvimento possui um giroscópio L3GD20 e um acelerômetro encapsulado com um magnetômetro LSM303DLHC (STMicroelectronics, 2013).

Figura 29 –STM32F3DISCOVERY

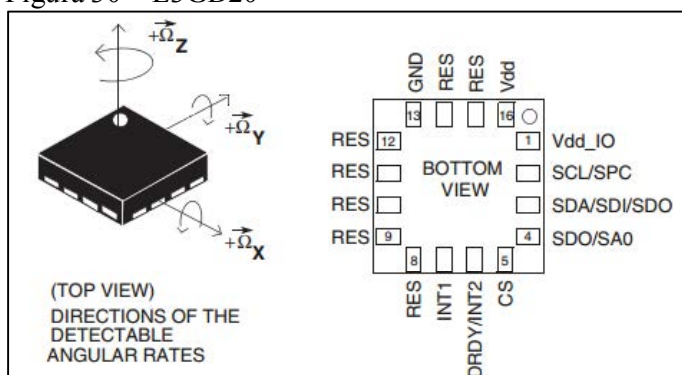


Fonte: (STMicroeletronics, 2013)

3.3.3 Giroscópio

O giroscópio é um sensor de velocidade angular, geralmente com saída digital e comunicação com o microcontrolador por protocolo de comunicação pré-definido, na Figura 30 é apresentado um exemplo de giroscópio, o L3GD20, presente na placa de desenvolvimento STM32F3DISCOVERY. O L3GD20 é um giroscópio de três eixos com saída digital por meio de protocolo de comunicação I2C ou SPI com as seguintes características : três escalas de operação 250/500/2000 graus por segundo, dado de saída com 16 bits e filtros integrados para atenuar ruídos (STMicroeletronics, 2013).

Figura 30 – L3GD20

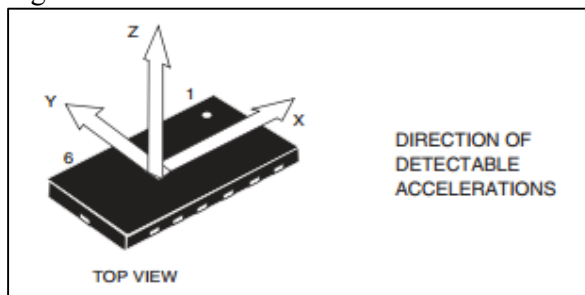


Fonte: (STMicroelectronics, 2013)

3.3.4 Acelerômetro

Sensor de aceleração linear, geralmente com saída digital e comunicação com o microcontrolador por protocolo de comunicação pré-definido. Na Figura 31 é ilustrado o acelerômetro LSM303DLHC, também incluído na placa de desenvolvimento STM32F3DISCOVERY, as características técnicas dele são as seguintes: quatro escalas selecionáveis $\pm 2g/\pm 4g/\pm 8g/\pm 16g$, dado de saída de 16 bit, comunicação por I2C (STMicroelectronics, 2013).

Figura 31 - Acelerômetro LSM303DLHC

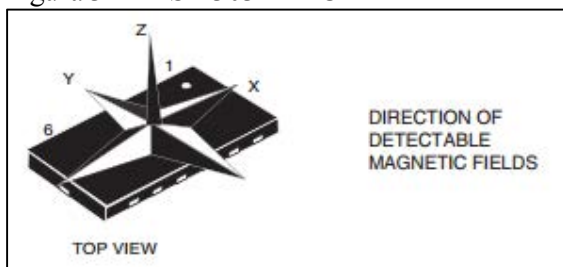


Fonte: (STMicroelectronics, 2013)

3.3.5 Magnetômetro - LSM303DLHC

Sensor de campo magnético servindo como bússola para o controle. Na Figura 32 é ilustrado o LSM303DLHC, o magnetômetro presente no microcontrolador STM32F3DISCOVERY, com as seguintes características : 3 canais, opera dentro da faixa de ± 1.3 até ± 8.1 gauss e comunicação via I2C (STMicroelectronics, 2013).

Figura 32 – LSM303DLHC

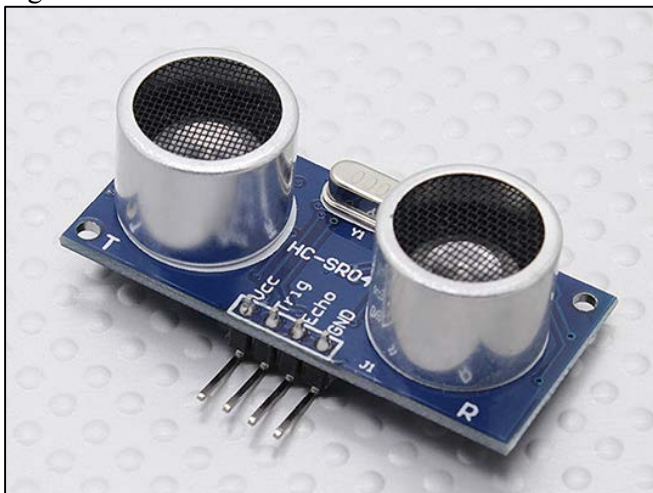


Fonte: (STMicroelectronics, 2013)

3.3.6 Sonar

Sensor utilizado para definir a distância de um objeto, emitindo um som de alta frequência e analisando o seu eco, normalmente aplicados para evitar colisões. O sonar da Figura 33, o HC-SR04, possui alcance de 2 cm até 500 cm, com resolução de 0,3 cm (HobbyKing, 2013).

Figura 33 - HC-SR04



Fonte: (HobbyKing, 2013)

3.3.7 Motor

É um motor sem escovas alimentado indiretamente pela bateria, onde tem como intermediário um controlador de velocidade, o qual varia linearmente com a tensão aplicada. Na Figura 34 é apresentado um modelo de motor com constante de conversão 900rpm/V, corrente máxima de 18 A e então propõe-se o uso de um controlador de velocidade (ESC) de capacidade entre 25 a 30 A (HobbyKing, 2013).

Figura 34 – Modelo NTM Prop Drive Series 28-30S 900kv



Fonte: (HobbyKing, 2013)

3.3.8 Hélice

As hélices serão conectadas aos motores nas quatro extremidades da estrutura em formato de X montado pelo frame, é responsável pelo empuxo que a rotação do motor gera, tendo assim as forças e torques que utilizamos para controlar o *quadcopter*. Na Figura 35 é apresentada um modelo bem comum para aplicação em *quadcopters* (HobbyKing, 2013).

Figura 35 – Hélices 10x4.5



Fonte: (HobbyKing, 2013)

3.3.9 ESC

É um controlador eletrônico de velocidade que tem como entrada a bateria e um sinal de controle provindo do *quadcopter* no formato de PWM, atuando de forma que a tensão de saída

seja correspondente ao valores do ciclo de trabalho do PWM aplicado, controlando assim a velocidade do motor. A escolha do ESC é diretamente ligado a bateria e motor que será utilizado, supondo o uso do motor proposto na seção 3.3.7, deve-se usar um ESC com capacidade entre 25 a 30 A, capacidade suprida pelo ESC presente na Figura 36 (HobbyKing, 2013).

Figura 36 - Hobby King 30A ESC 3A UBEC

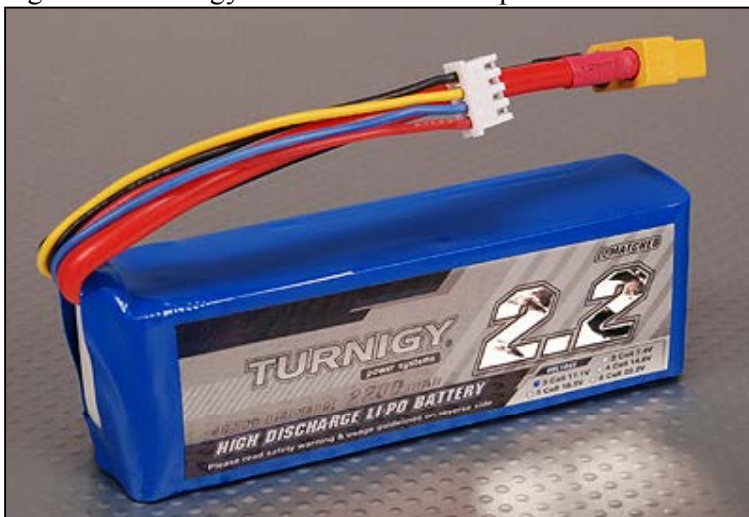


Fonte: (HobbyKing, 2013)

3.3.10 Bateria

A bateria utilizada para esta aplicação normalmente é a de polímero de lítio (LiPO), pela sua alta densidade de armazenamento e capacidade de alimentar com altas correntes e rápida descarga. A bateria escolhida para este possível protótipo foi a da Figura 37, que possui três células, tensão de saída 11.1V, 200g de peso e capacidade mínima de 2200mAh (HobbyKing, 2013).

Figura 37 - Turnigy 2200mAh 3S 40C Lipo Pack

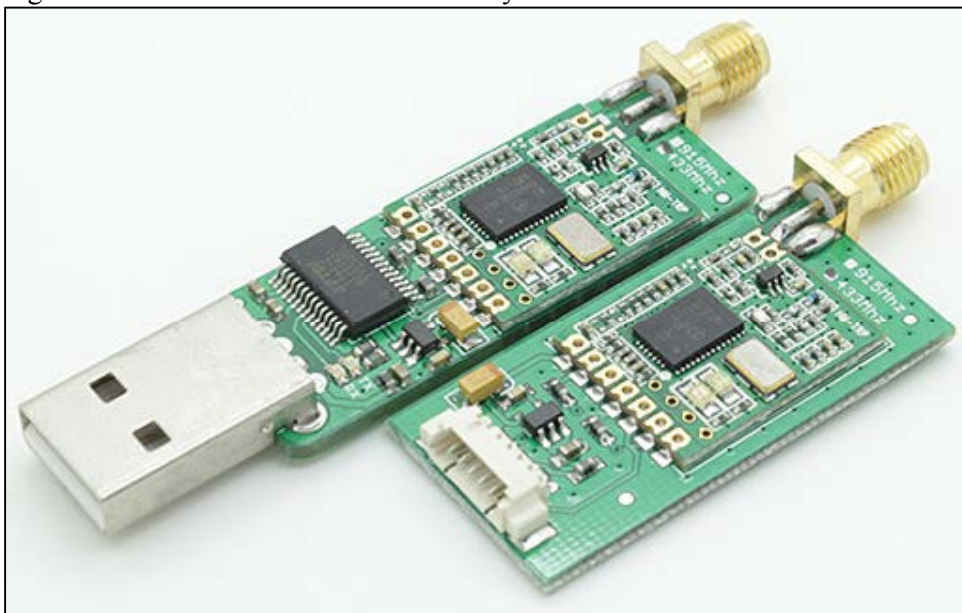


Fonte: (HobbyKing, 2013)

3.3.11 Sistema de Telemetria

Responsável pela interface entre o *quadcopter* e a estação base, tem como função enviar os dados dos sensores para estação base e receber os valores de PWM a ser aplicado em cada motor. O sistema de telemetria apresentado na Figura 38, o mesmo opera a 433Mhz com potência de saída ajustável entre 1 até 20dBm. O sistema de telemetria apresentado é baseado em código livre e possui software para configurá-lo, assim como definir os parâmetros do protocolo e comunicação (HobbyKing, 2013).

Figura 38 - FPV 433Mhz Radio Telemetry Kit 100mW V1.1

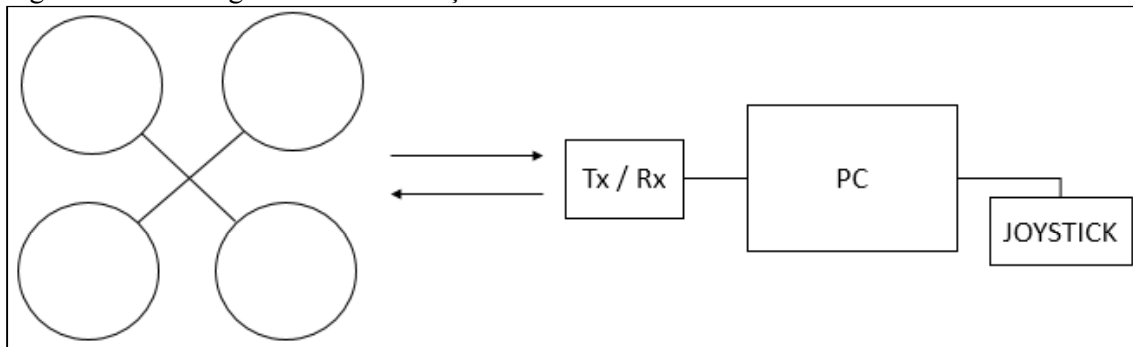


Fonte: (HobbyKing, 2013)

3.3.12 Estação Base

Responsável pela lógica de controle, utilizando as leituras dos sensores e a entrada por um *joystick* efetuada pelo usuário, calcula-se então os valores PWM a ser aplicado em cada motor e transmite-se essa informação para o *quadcopter* por meio do Tx-Rx, como ilustrado na Figura 39.

Figura 39 – Visão geral da comunicação dos sistemas



Fonte: Autoria Própria

Com isso, define-se os componentes necessários para a construção de um *quadcopter*, sendo então proposto como trabalho futuro a implementação do mesmo. Na próxima seção será apresentado o desenvolvimento do modelo virtual utilizando o *software simulink*, baseando-se nos componentes propostos para o protótipo.

3.4 MODELO VIRTUAL NO SIMULINK

O objetivo do desenvolvimento do modelo é implementar as equações obtidas na seção 3.2, e atuar sobre o modelo de forma a mantê-lo estável utilizando um controlador *fuzzy* para controlar a rotação dos motores para eliminar o erro à entradas para altura, *roll pitch* e *yaw*.

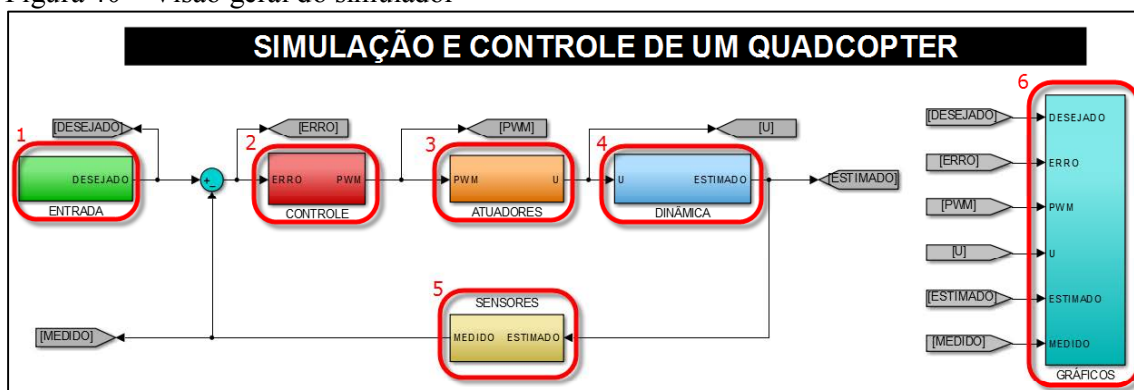
A Tabela 4 apresenta os valores das constantes que foram adotadas para o simulador, de acordo com os cálculos e testes práticos demonstrados em (DE SOUSA,2011).

Tabela 4 – Constantes adotadas

Constante		Valor
Aceleração Gravitacional	g	9.81 m/s^2
Massa total	m	1 kg
Momento de Inércia do eixo X	I_{xx}	0.04 Nm*s^2
Momento de Inércia do eixo Y	I_{yy}	0.04 Nm*s^2
Momento de Inércia do eixo Z	I_{zz}	0.08 Nm*s^2
Coeficiente de impulso	b	$56*10^{-6} \text{ Ns}^2/\text{rad}^2$
Coeficiente de arrasto	d	$1.1*10^{-6} \text{ Nms}^2/\text{rad}^2$
Distancia entre CM e centro do motor	L	0.225 m
Momento de inercia de rotação total	J_{tp}	$7.33*10^{-5} \text{ Nm*s}^2$

Na Figura 40 é apresentado a visão geral do sistema simulador, o qual é constituído de 6 subsistemas, destacados por números de um a seis, sendo eles entrada, controle, atuadores, dinâmica, sensores e gráficos, cada um dos subsistemas será descrito ao longo deste capítulo indicando sua função e o modo que foi implementado.

Figura 40 – Visão geral do simulador

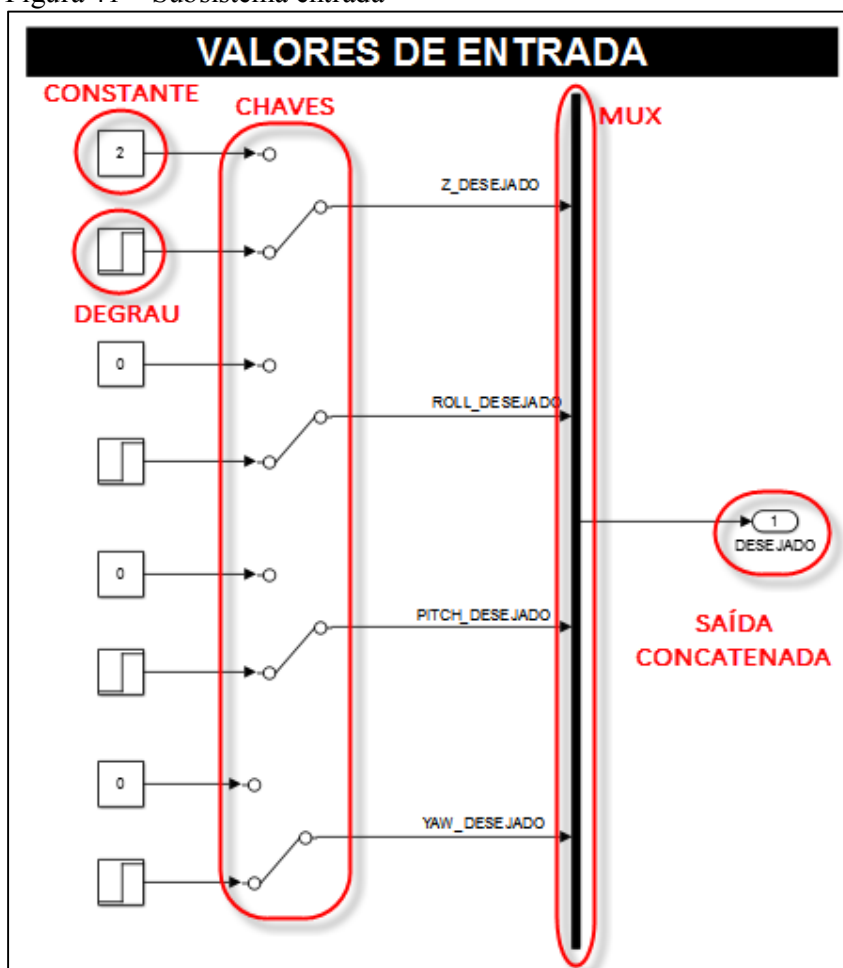


Fonte: Autoria Própria

3.4.1 Entrada

O subsistema ENTRADA, apresentado na Figura 41, é onde o usuário determina os valores que as variáveis controláveis devem atingir durante a simulação, sendo estas altura, *roll*, *pitch* e *yaw*. Foram utilizadas chaves para variar a entrada dos parâmetros entre um valor constante e uma função degrau, de forma mais eficiente, ficando claro a opção definida. Após os valores de entrada serem definidos, utiliza-se um mux 4:1 concatenando as variáveis para o próximo subsistema.

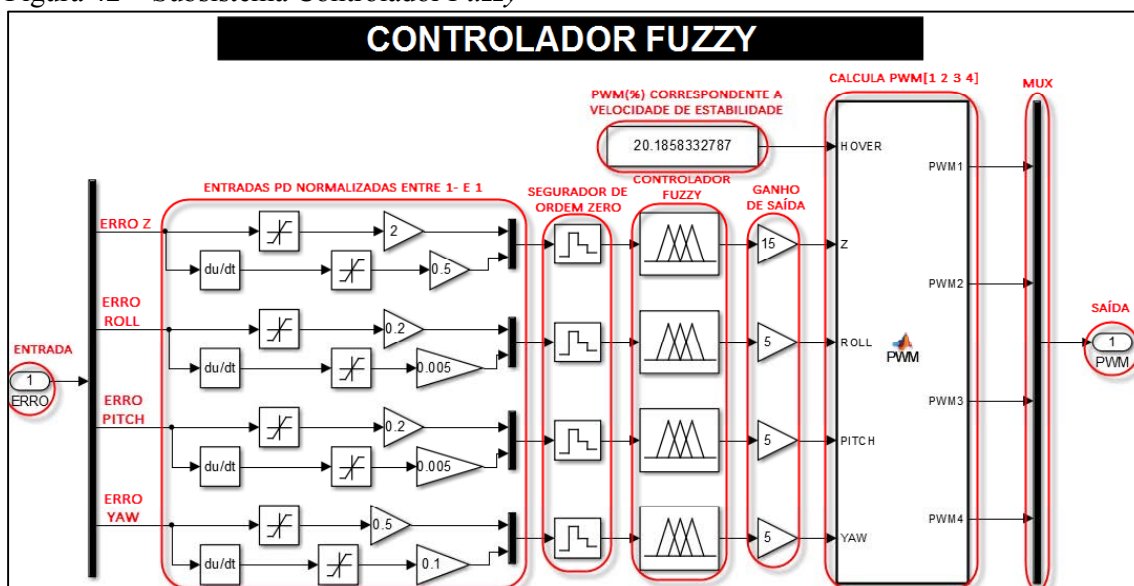
Figura 41 – Subsistema entrada



Fonte: Autoria Própria

3.4.2 Controlador *Fuzzy*

O subsistema do Controlador *Fuzzy* é apresentado na Figura 42 inicia a sua lógica por um *demux* 1:4, obtendo assim os quatro sinais de erro, os quais foram previamente calculados pela subtração entre o valor desejado e o medido.

Figura 42 – Subsistema Controlador *Fuzzy*

Fonte: Autoria Própria

O tipo de controlador utilizado foi um PD *fuzzy*, aplicado a cada variável a ser controlada, totalizando assim 4 controladores, que são implementados normalizando as entradas proporcional e derivativa de tal forma que o controlador receba valores entre -1 e 1, para garantir esta condição, utilizou-se blocos de saturação e ganhos, de tal forma que os limites da saturação multiplicado pelo não ultrapassa-se um em módulo. A Tabela 5 apresenta os parâmetros definidos no simulador para os blocos de saturação e ganho aplicados de forma a condicionar a entrada do controlador *fuzzy*.

Tabela 5 – Ganho e saturação dos controladores

Sinal	Proporcional		Derivativo	
	Saturação	Ganho	Saturação	Ganho
Erro Z	$\pm 0.5m$	2	$\pm 2m/s$	0.5
Erro Roll	$\pm 5^\circ$	0.2	$\pm 200^\circ/s$	0.005
Erro Pitch	$\pm 5^\circ$	0.2	$\pm 200^\circ/s$	0.005
Erro Yaw	$\pm 2^\circ$	0.5	$\pm 10^\circ/s$	0.1

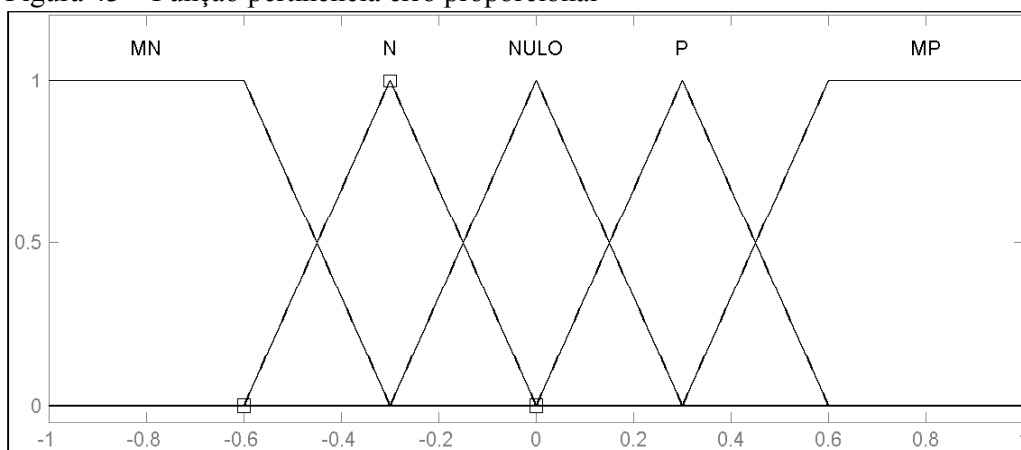
Após o condicionamento do sinal é utilizado um segurador de ordem zero com frequência de operação de 1kHz.

Com isso, os sinais de erro e derivada do erro, já normalizados, alcançam o controlador *fuzzy*. O controlador *fuzzy* utilizado nos quatro controladores utiliza o mesmo modelo *fuzzy* de

controlador, que opera com duas entradas: erro e derivada do erro, ambas normalizadas e com valores entre -1 e 1, a saída do controlador é um nível de atuação com valores entre -1 e 1. O controlador foi desenvolvido utilizando o *Toolbox fuzzy* disponível no software *matlab 2012b*.

As três funções pertinência, erro, derivada do erro e nível de atuação são iguais e possuem a representação gráfica da Figura 43, onde MN representa um valor muito negativo, N um valor negativo, Nulo um valor próximo de zero, P um valor positivo e MP um valor muito positivo. A representação textual da função pertinência é apresentada na Tabela 6 (SIVANANDAM, 2007).

Figura 43 – Função pertinência erro proporcional



Fonte: Autoria Própria

Tabela 6 – Parâmetros da função pertinência erro proporcional

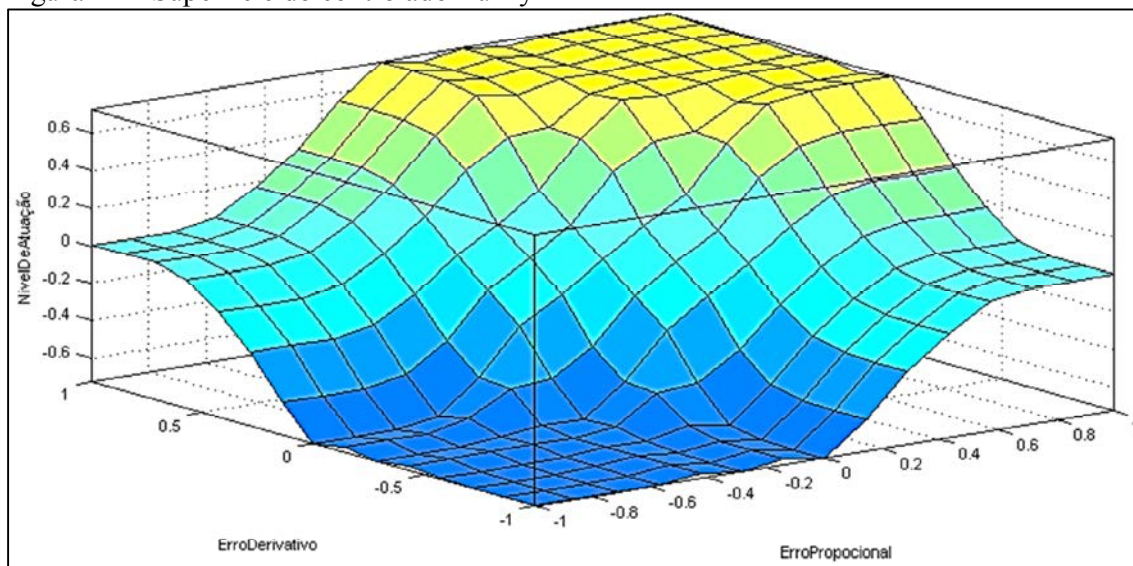
Conjunto <i>Fuzzy</i>	Definição
MN	Trapezoidal [-2.8 -1.2 -0.6 -0.3]
N	Triangular [-0.6 -0.3 0]
NULO	Triangular [-0.3 0 0.3]
P	Triangular [0 0.3 0.6]
MP	Trapezoidal [0.3 0.6 1.2 2.8]

A base de regras criada para o controlador é apresentada na Tabela 7 e a superfície do controlador *fuzzy* é ilustrada na

Figura 44. No Apêndice A é apresentado a estrutura *fis* completa do controlador.

Tabela 7 – Base de regras do controlador *fuzzy*

Derro	Erro	MN	N	NULO	P	MP
MN		MN	MN	MN	N	NULO
N		MN	MN	N	NULO	P
NULO		MN	N	NULO	P	MP
P		N	NULO	P	MP	MP
MP		NULO	P	MP	MP	MP

Figura 44 – Superfície do controlador *fuzzy*

Fonte: Autoria Própria

Após a determinação do nível de atuação pelo controlador *fuzzy* é aplicado um ganho a cada controlador, sendo de valor 15 para a altura e 5 para os ângulos *roll*, *pitch* e *yaw*. Valores tais, que foram determinados por meio de testes e avaliação da resposta do sistema ao controlador, assim como os ganhos e saturações previamente discutidos.

Após a aplicação dos ganhos nos controladores é implementado um bloco que faz os cálculos presentes nas equações (56), (57), (58) e (59). Cálculo que descreve a relação dos motores com os movimentos fundamentais apresentados na Tabela 3.

$$PWM_1 = HOVER + Z - PITCH - YAW \quad (56)$$

$$PWM_2 = HOVER + Z - ROLL + YAW \quad (57)$$

$$PWM_3 = HOVER + Z + PITCH - YAW \quad (58)$$

$$PWM_4 = HOVER + Z + ROLL + YAW \quad (59)$$

O valor da constante de *hovering* é determinada pela força U_1 , de tal forma que seja igual a força do peso, ou seja, sem aceleração e estável. Nas equações (60) e (61) determina-se a velocidade dos motores para que o *quadcopter* mantenha-se estável no espaço, considerando m igual a 1 kg, g igual a $9,81 \text{ m/s}^2$ e b , a constante de empuxo, determinada, pela hélice de $56 * 10^{-6}$, como previamente adotado na seção 3.4.

$$U_1 = m * g = 4 * b * \Omega^2 \quad (60)$$

$$\Omega_{HOVERING} = \sqrt{\frac{m * g}{4 * b}} = \sqrt{\frac{1 * 9.81}{4 * 56 * 10^{-6}}} = 209.271696 \text{ rad/s} \quad (61)$$

Deve-se então calcular o valor em PWM(%) que ao ser aplicado faça com que o motor atinja a velocidade de *hovering*, assim considera-se a voltagem da bateria como 11 Volts e o ganho do motor como 900 rpm/V, sabe-se que ambos os parâmetros são diretamente proporcionais a velocidade angular, podemos então determinar o PWM(%) pela equação (62).

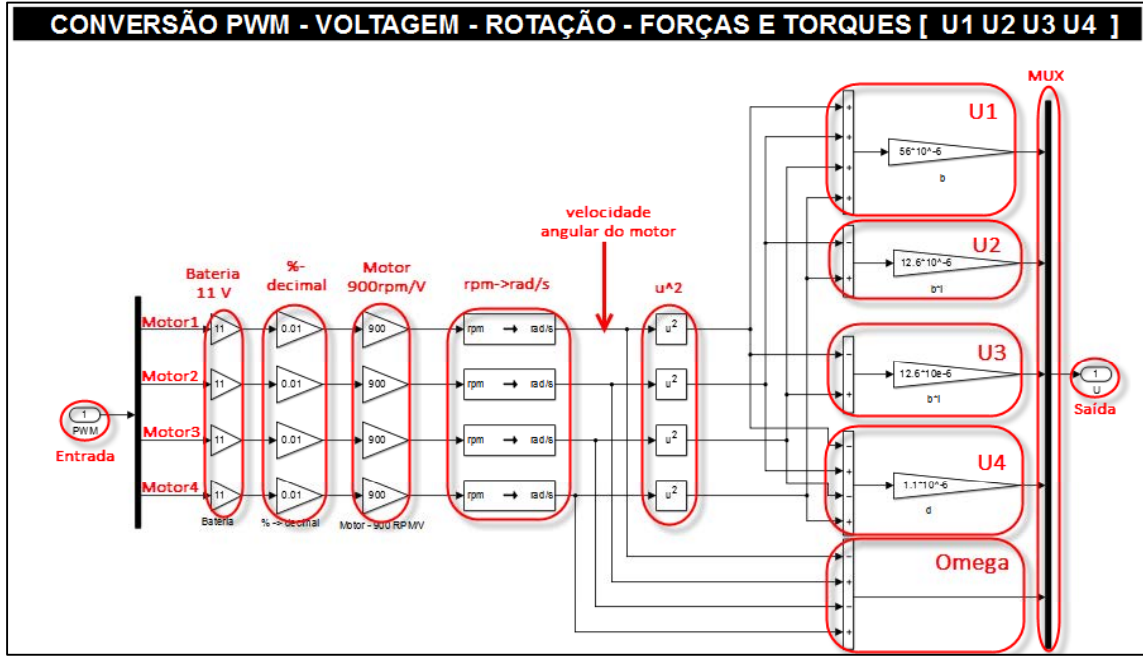
$$\begin{aligned} PWM_{HOVERING}(\%) &= \frac{\Omega_{HOVERING}}{V_{Bateria} * K_{motor} * K_{rpm \rightarrow \frac{rad}{s}}} * 100\% \\ &= \frac{209.271696}{11 * 900 * \frac{2 * \pi}{60}} * 100\% = 20.185833\% \end{aligned} \quad (62)$$

Tem-se assim como saída do subsistema controlador, o valor de porcentagem do ciclo ativo de um sinal PWM, para os quatro motores separadamente.

3.4.3 Atuadores

No subsistema atuadores, o valor PWM a ser aplicado em cada motor será convertido em voltagem, rotação e com as mesmas será determinado os valores de forças e torques para os movimentos fundamentais conforme a visão geral apresentada na Figura 45.

Figura 45 – Subsistema atuadores



Fonte: Autoria Própria

A conversão do valor em PWM para a rotação do motor dado em rad/s é feita através da multiplicação do sinal PWM pela voltagem da bateria, em seguida uma conversão para retirar o efeito da porcentagem, com isso tem-se o valor da tensão aplicada ao motor, considerando o motor como um ganho de 900rpm/V, e convertendo a velocidade angular de rpm para rad/s, tem-se então a velocidade angular dos quatro motores em rad/s.

Para o cálculo da força e torques implementou-se em diagrama de blocos as equações do grupo (63).

$$\begin{cases} U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = lb(-\Omega_2^2 + \Omega_4^2) \\ U_3 = lb(-\Omega_1^2 + \Omega_3^2) \\ U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ \Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \end{cases} \quad (63)$$

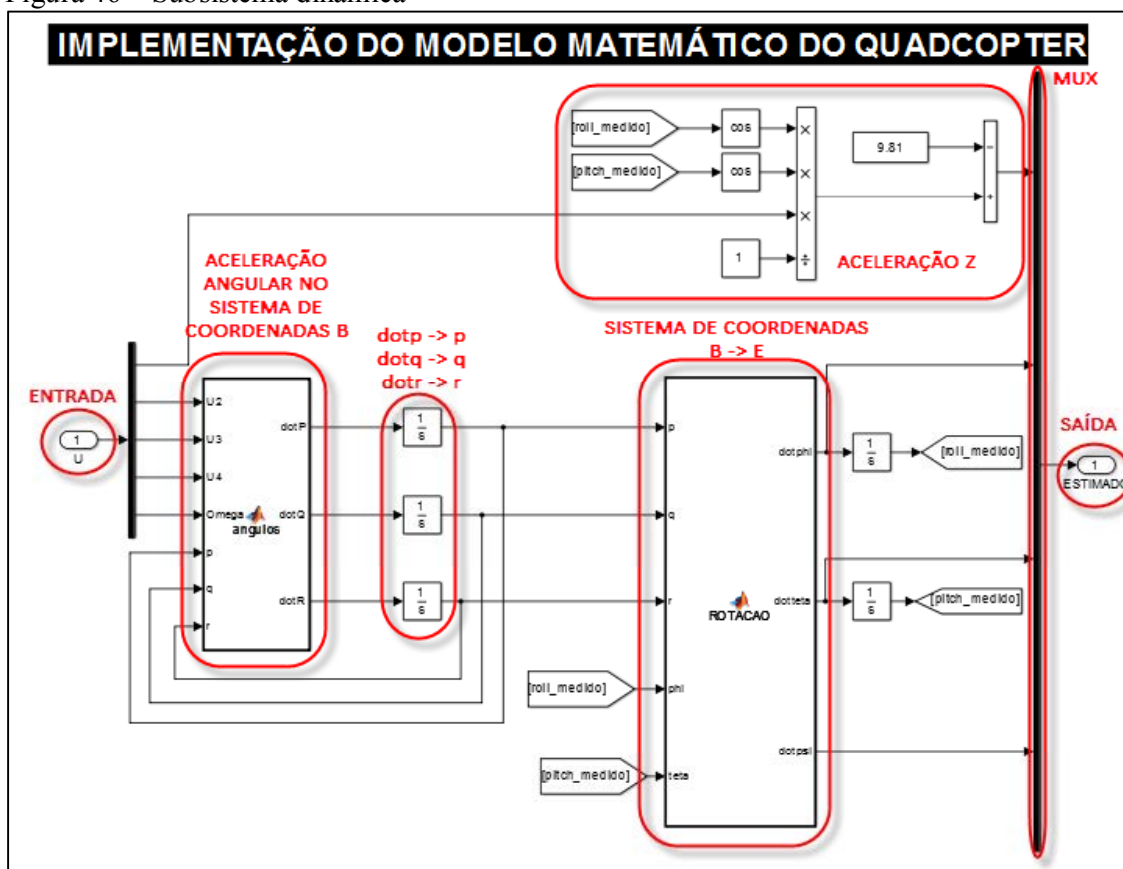
3.4.4 Dinâmica do quadcopter

Uma vez definida a força e os torques para os movimentos fundamentais, pode-se avaliar a resposta do modelo das mesmas. Para isso, implementa-se as equações que descrevem a dinâmica do *quadcopter* no *simulink* por meio de diagrama de blocos e também programação. As equações do grupo (64) são implementadas nesse subsistema, que possui como saída a aceleração translacional $\ddot{Z}$ e as velocidades angulares $\dot{\phi}$, $\dot{\theta}$ e $\dot{\psi}$.

$$\left\{ \begin{array}{l} T_{\theta} = \begin{bmatrix} 1 & s_{\phi}t_{\theta} & c_{\phi}t_{\theta} \\ 0 & c_{\phi} & -s_{\phi} \\ 0 & s_{\phi}/c_{\theta} & c_{\phi}/c_{\theta} \end{bmatrix} \\ \dot{\Theta}^E = T_{\Theta}\omega^B \\ \ddot{Z} = -g + (c_{\theta}c_{\phi})\frac{U_1}{m} \\ \dot{p} = \frac{I_{YY} - I_{ZZ}}{I_{XX}}qr - \frac{J_{TP}}{I_{XX}}q\Omega + \frac{U_2}{I_{XX}} \\ \dot{q} = \frac{I_{ZZ} - I_{XX}}{I_{YY}}pr + \frac{J_{TP}}{I_{YY}}p\Omega + \frac{U_3}{I_{YY}} \\ \dot{r} = \frac{I_{XX} - I_{YY}}{I_{ZZ}}pq + \frac{U_4}{I_{ZZ}} \end{array} \right. \quad (64)$$

O cálculo implementado no diagrama de bloco calcula inicialmente as acelerações angulares em relação ao sistema de coordenadas B. Após um integrador utiliza-se a matriz T_{θ} para mudar o sistema de coordenadas, obtendo assim as velocidades angulares no sistema de coordenadas . Utiliza-se então os valores dos ângulos no sistema de coordenadas E para determinar a aceleração translacional do eixo Z, conforme a Figura 46.

Figura 46 – Subsistema dinâmica

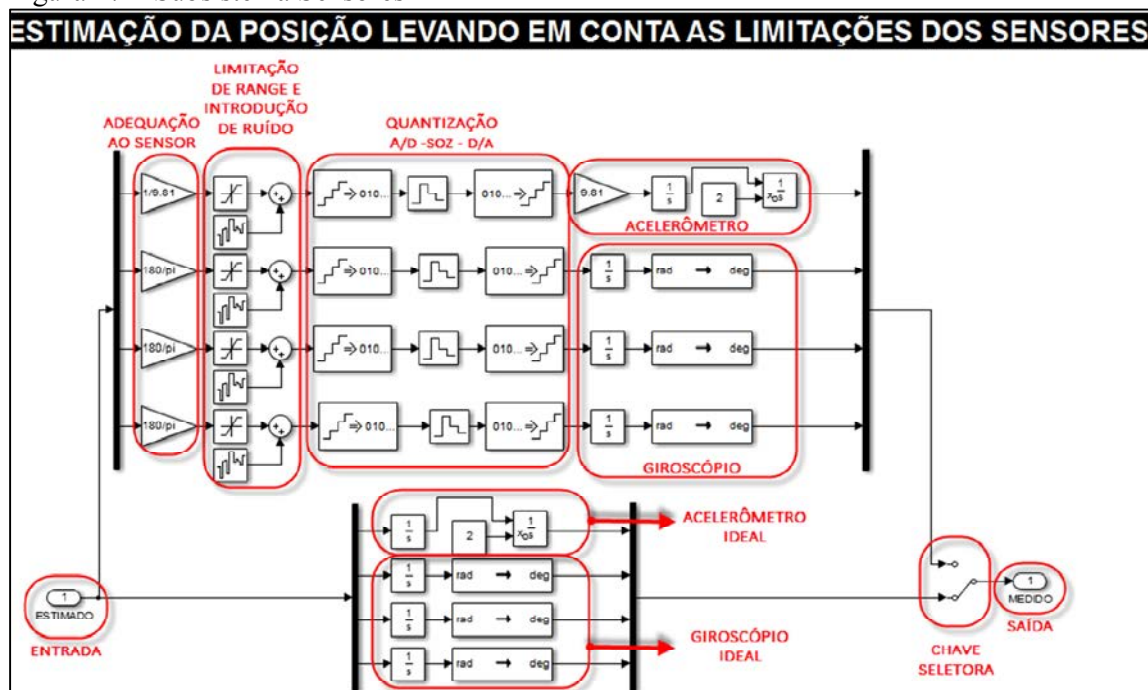


Fonte: Autoria Própria

3.4.5 Sensores

A função do subsistema Sensores é retornar valores referentes a posição translacional Z e rotacional *roll*, *pitch* e *yaw*, a partir dos valores de aceleração do eixo Z e das velocidades rotacionais provenientes do subsistema dinâmica. Foi implementado duas variações de sensores para comparar os resultados: sensores giroscópio e acelerômetro com limitações de intervalo e frequência de operação, assim como introdução de ruídos e erros gerados por quantização, a outra abordagem foi implementada utilizando sensores ideais, sem erros ou limitações definidos. A fim de facilitar a troca de sistema de sensores foi implementada uma chave seletora que define qual sistema será utilizado. O subsistema é apresentado na Figura 47.

Figura 47 – Subsistema Sensores

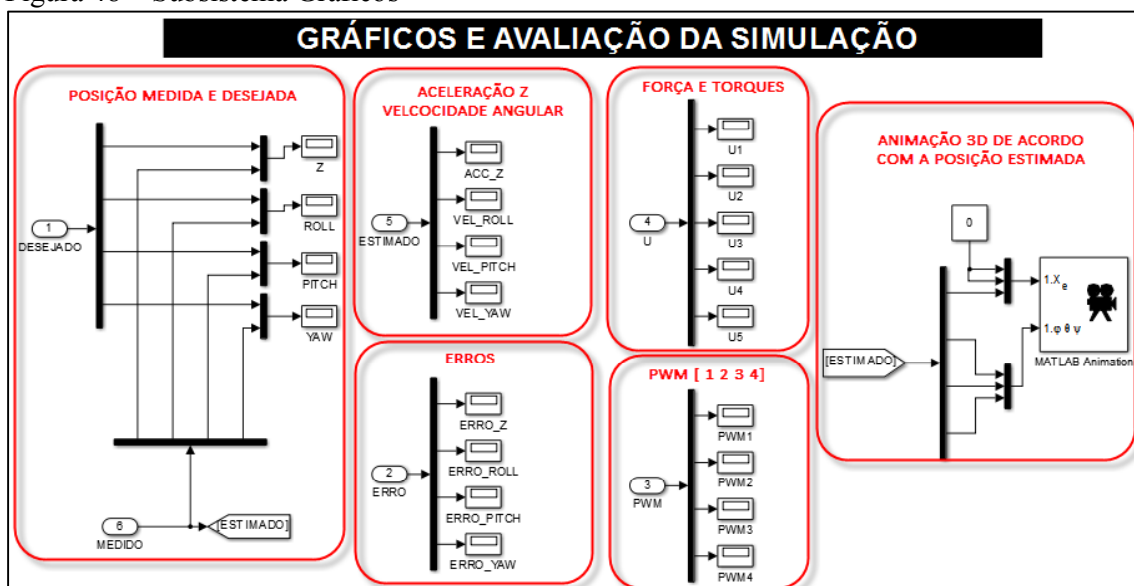


Fonte: Autoria Própria

3.4.6 Gráficos

Um bloco extra que não influencia no sistema é o subsistema Gráfico onde as variáveis de interesse foram organizadas de forma a facilitar a sua avaliação por meio de gráficos e também uma animação 3d. O subsistema Gráficos é apresentado na Figura 48.

Figura 48 – Subsistema Gráficos



Fonte: Autoria Própria

Após elaboração do modelo virtual do *quadcopter*, foi feita a análise do resultado do mesmo. Esta análise será apresentada no próximo capítulo.

4 ANÁLISE DE RESULTADOS

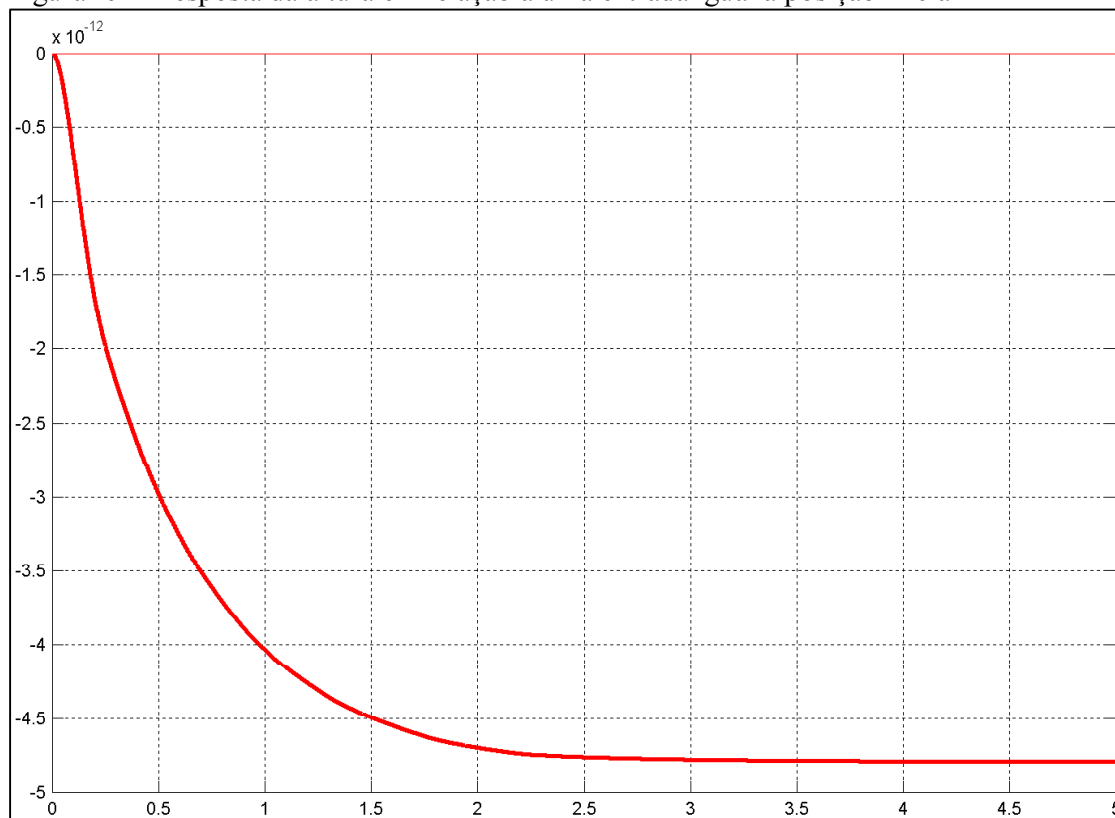
Ao longo deste capítulo serão apresentados os gráficos de saída mostrando a resposta do sistema a uma entrada degrau. Primeiramente será apresentado a resposta do sistema utilizando os sensores ideais na seção 4.1 e, posteriormente, na seção 4.2 será apresentado a resposta do sistema com sensores não ideais.

4.1 SIMULAÇÃO USANDO SENSORES IDEAIS

4.1.1 Hovering

Para a análise do *hovering* as entradas são todas constantes e iguais as condições iniciais do sistemas. Obtém-se então como saída a altura constante com erro desprezível, apresentado na Figura 49, a altura medida pelo subsistema sensores.

Figura 49 – Resposta da altura em relação a uma entrada igual a posição inicial



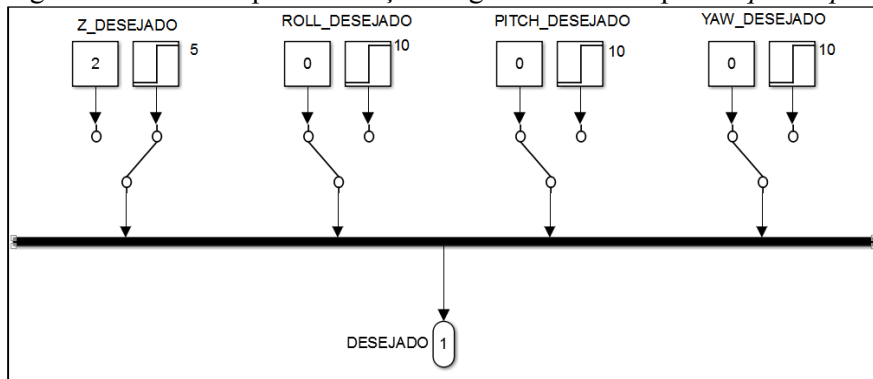
Fonte: Autoria Própria

4.1.2 Altura

Para se analisar a situação de ganho de altura, aplica-se uma entrada degrau com valor inicial de 2 metros, que também é o valor inicial da altura do *quadcopter*. Após um segundo, o valor do `Z_DESEJADO` sobe para 5 metros, devendo então o controlador atuar para

acompanhar este sinal. O subsistema ENTRADA para essa avaliação é apresentado na Figura 50.

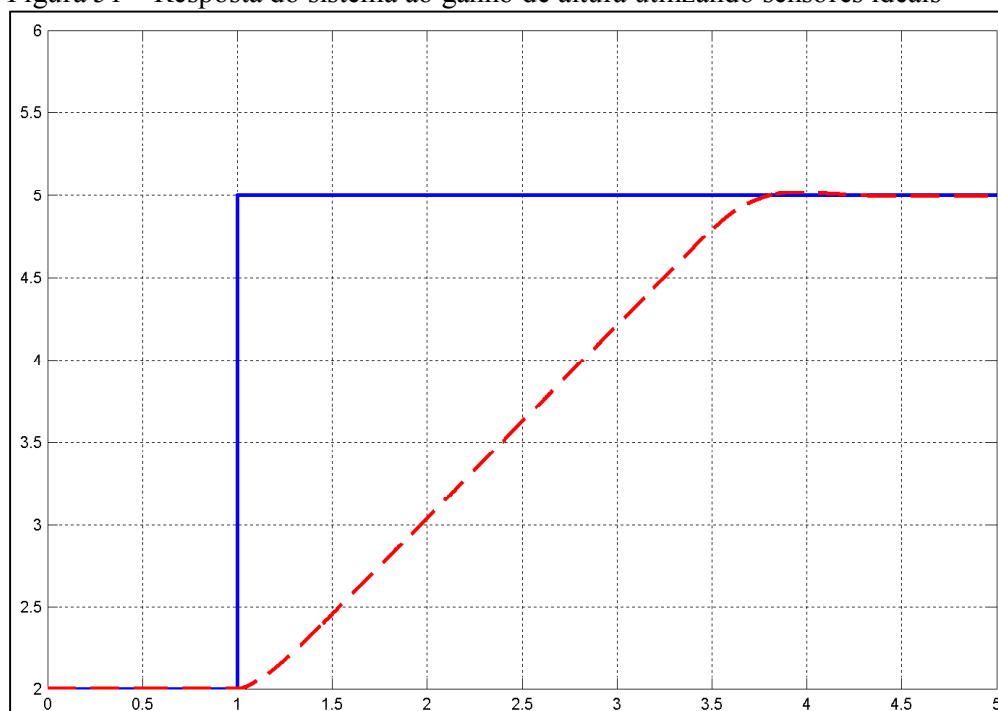
Figura 50 – Entrada para avaliação de ganho de altura para o *quadcopter*



Fonte: Autoria Própria

A resposta do sistema é apresentada na Figura 51, onde a curva azul representa a altura desejada e a curva em vermelho tracejado representa a altura medida. Verifica-se que o *quadcopter* atinge o valor desejado de 5 metros depois de aproximadamente 2,750 segundos, com *overshoot* menor que 10 cm.

Figura 51 – Resposta do sistema ao ganho de altura utilizando sensores ideais

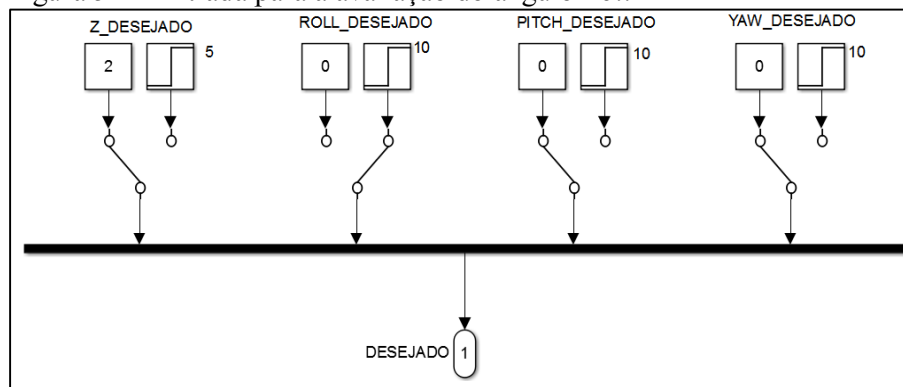


Fonte: Autoria Própria

4.1.3 Roll

Para avaliar a resposta do *Roll*, aplica-se como entrada um degrau com valor inicial zero e valor final dez após um segundo, sendo implementado de acordo com a Figura 52.

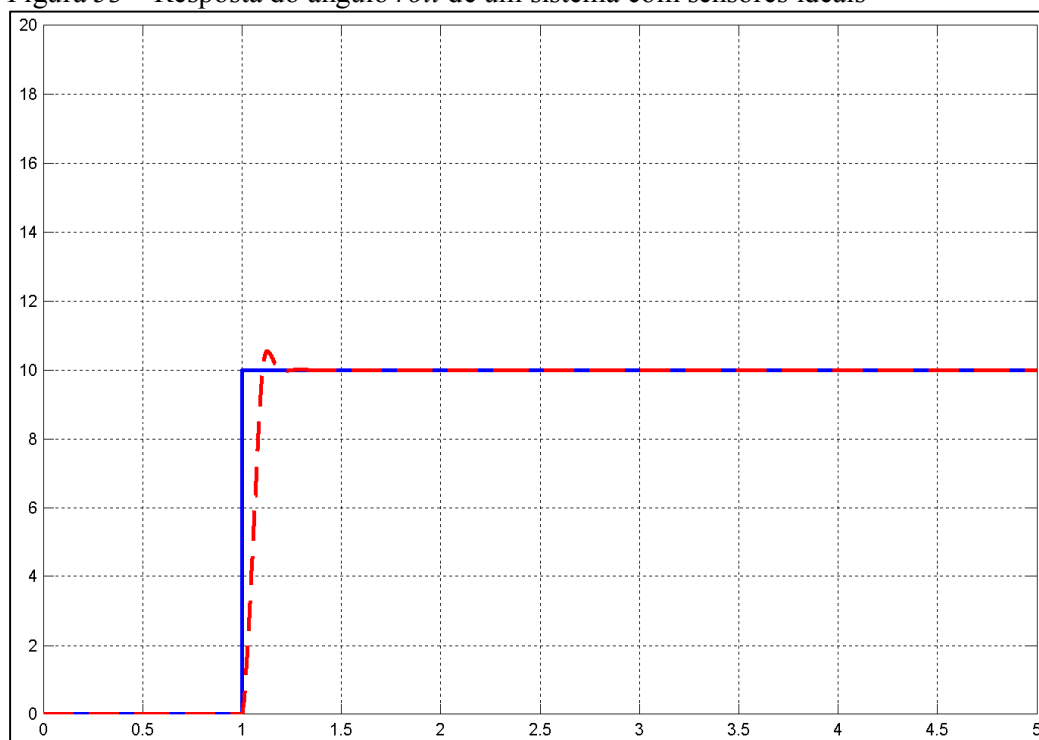
Figura 52 – Entrada para a avaliação do ângulo *Roll*



Fonte: Autoria Própria

Na Figura 53 são apresentadas as curvas da rotação *roll*, sendo a de cor azul a rotação desejada e a curva vermelha a medida. Verifica-se que o modelo virtual atinge o valor desejado de 10° por volta de 0,25 s após a aplicação da mudança, com *overshoot* por volta de 1°.

Figura 53 – Resposta do ângulo *roll* de um sistema com sensores ideais

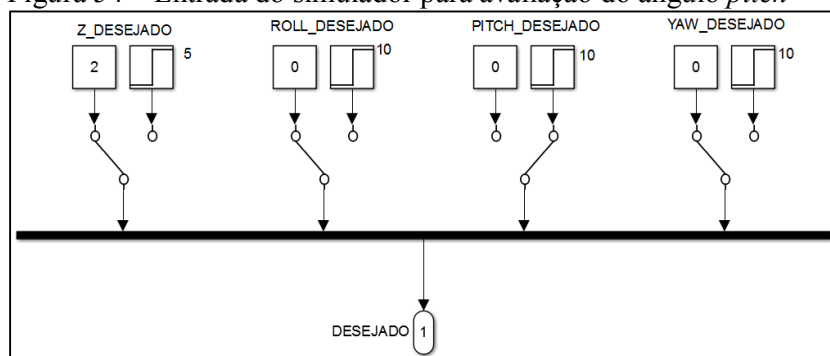


Fonte: Autoria Própria

4.1.4 Pitch

Para avaliar o *Pitch*, aplica-se como entrada um degrau valor inicial zero e valor final dez após um segundo, sendo implementado de acordo com a Figura 54.

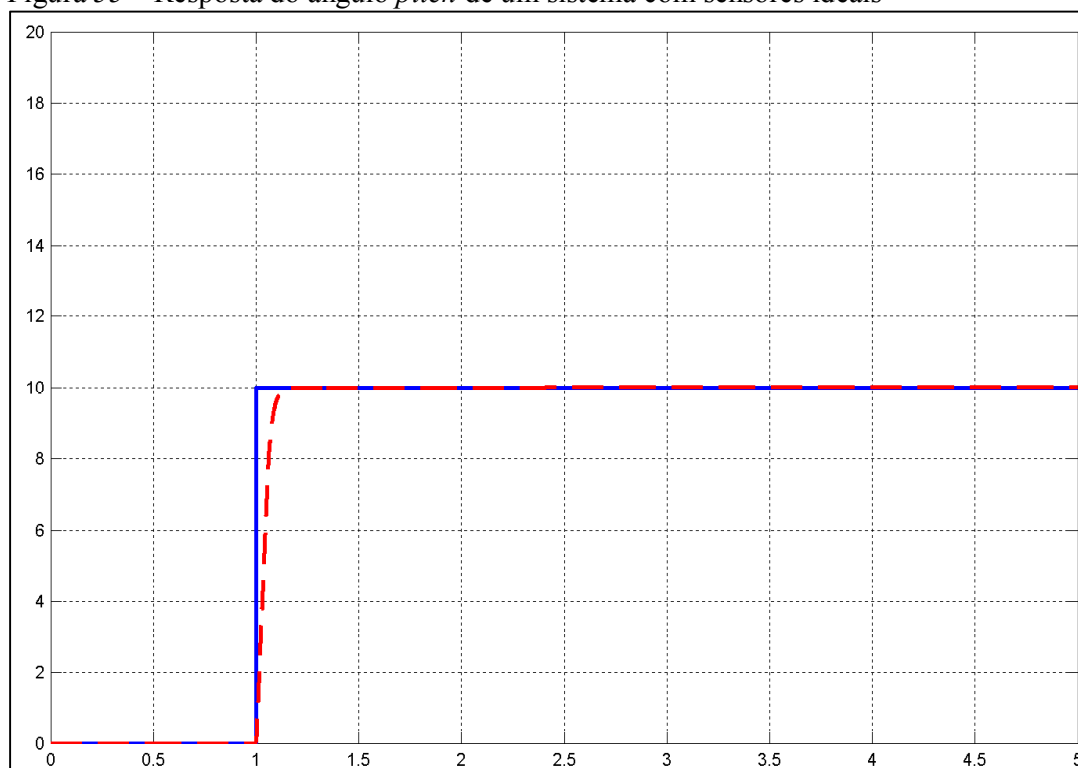
Figura 54 – Entrada do simulador para avaliação do ângulo *pitch*



Fonte: Autoria Própria

Na Figura 55 são apresentadas as curvas da rotação *pitch*, sendo a de cor azul a rotação desejada e a curva vermelha a medida. Verifica-se que o modelo virtual atinge o valor desejado de 10° por volta de 0,25 s após a aplicação da mudança, com *overshoot* insignificante.

Figura 55 – Resposta do ângulo *pitch* de um sistema com sensores ideais

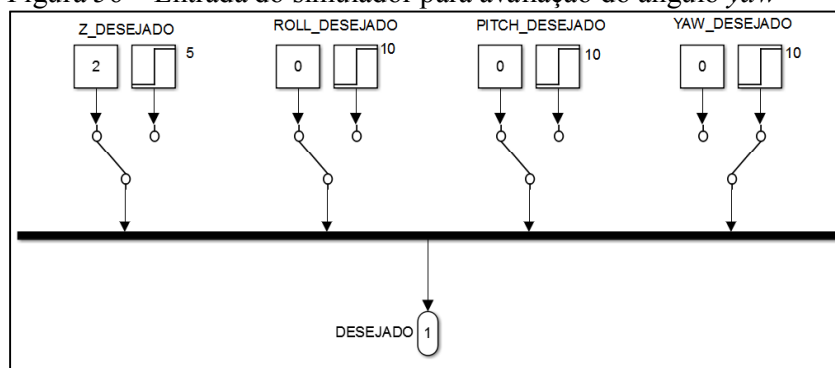


Fonte: Autoria Própria

4.1.5 Yaw

Para avaliar o *Yaw*, aplica-se como entrada um degrau valor inicial zero e valor final dez após um segundo, sendo implementado de acordo com a Figura 56.

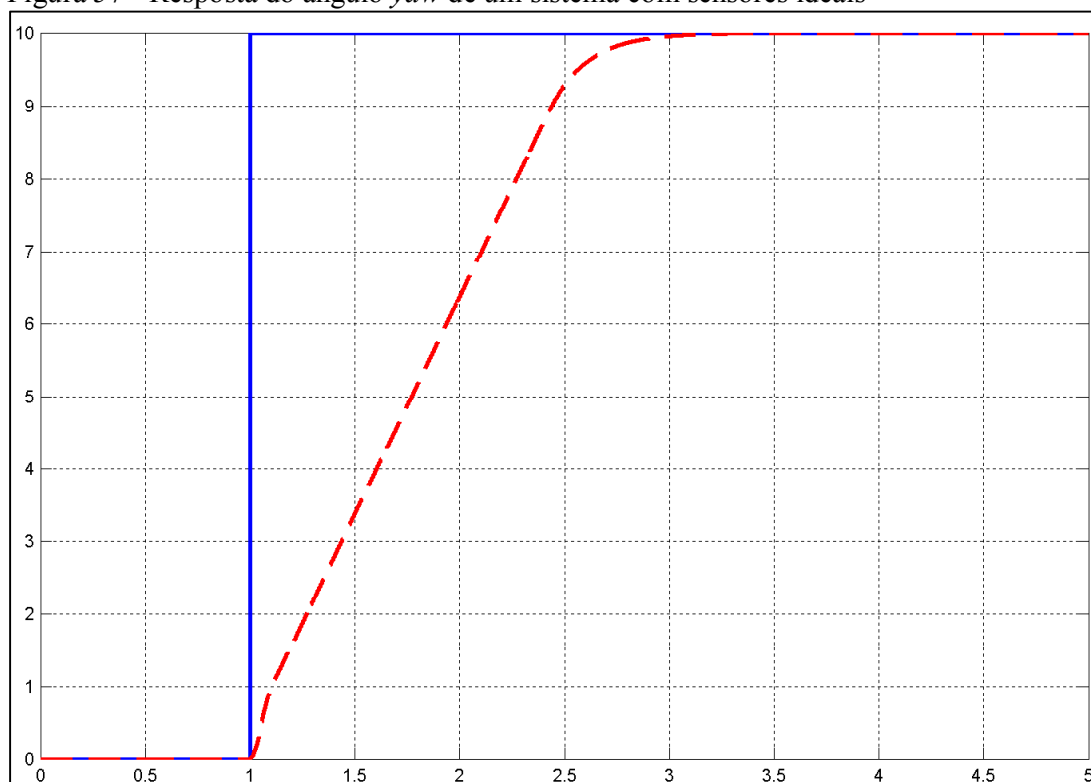
Figura 56 – Entrada do simulador para avaliação do ângulo *yaw*



Fonte: Autoria Própria

Na Figura 57 são apresentadas as curvas da rotação *yaw*, sendo a de cor azul a rotação desejada e a curva vermelha a medida. Verifica-se que o modelo virtual atinge o valor desejado de 10° por volta de 2 s após a aplicação da mudança, com *overshoot* insignificante.

Figura 57 - Resposta do ângulo *yaw* de um sistema com sensores ideais



Fonte: Autoria Própria

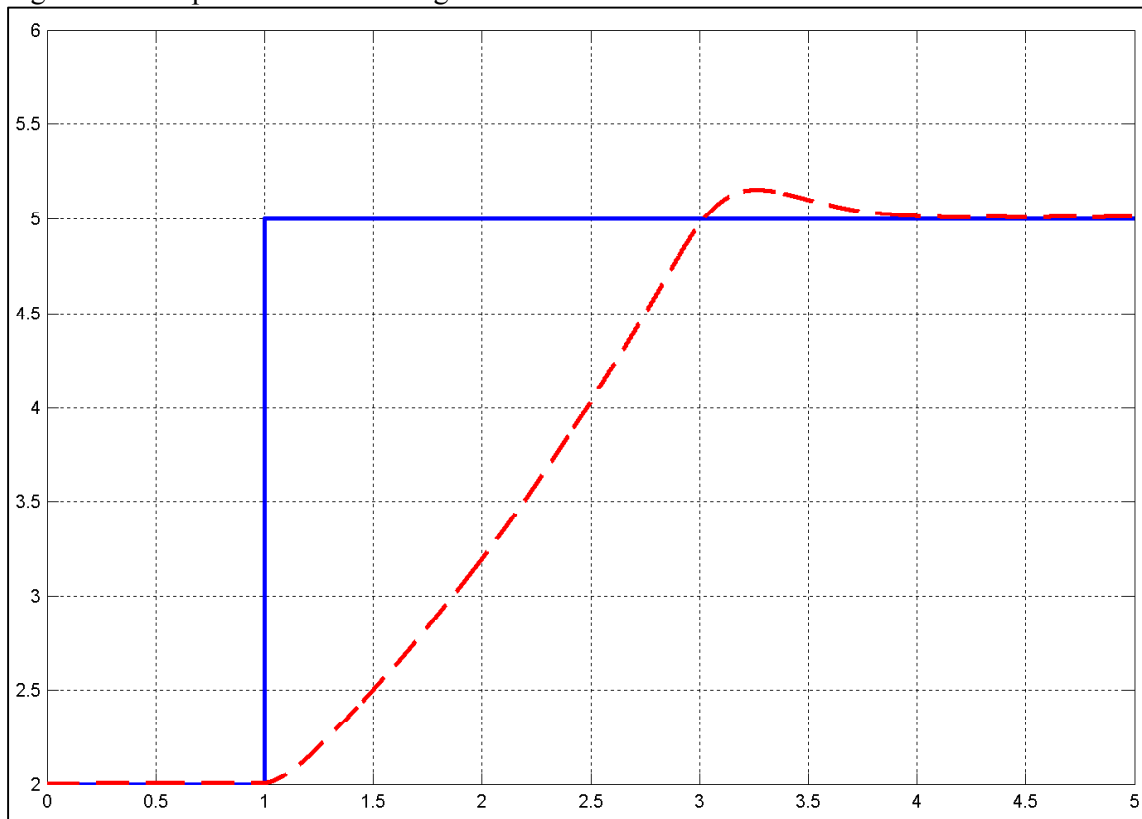
4.2 LIMITAÇÃO DOS SENSORES

Nessa seção serão apresentados os resultados referentes ao sistema utilizando os sensores modelados com as limitações e intervalo de operação, frequência de atualização de dados, erros de quantização e ruídos.

4.2.1 Altura

Para uma entrada implementada da forma apresentada na Figura 50, tem-se como resposta a curva vermelha apresentada na Figura 58, onde a curva azul representa o valor da altura desejado. Verifica-se que o sistema alcança o valor desejado 3 segundos, com *overshoot* por volta de 20 cm.

Figura 58 - Resposta do sistema ao ganho de altura utilizando sensores não ideais

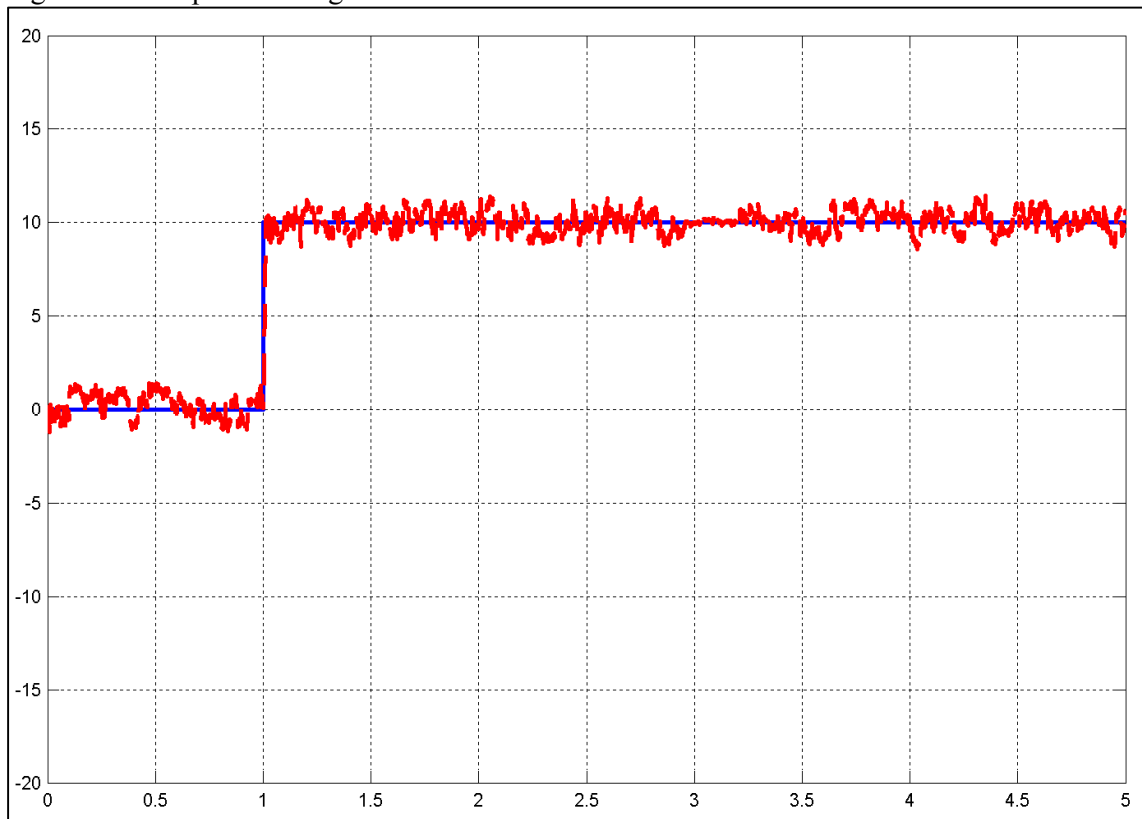


Fonte: Autoria Própria

4.2.2 Roll

Para uma entrada implementada da forma da Figura 52, tem-se como resposta a curva vermelha apresentada na Figura 59, a curva azul representa o valor de *roll* desejado. Verifica-se que o sistema alcança rapidamente o valor desejado, porém existe um ruído de magnitude de aproximadamente um grau.

Figura 59 - Resposta do ângulo *roll* de um sistema com sensores não ideais

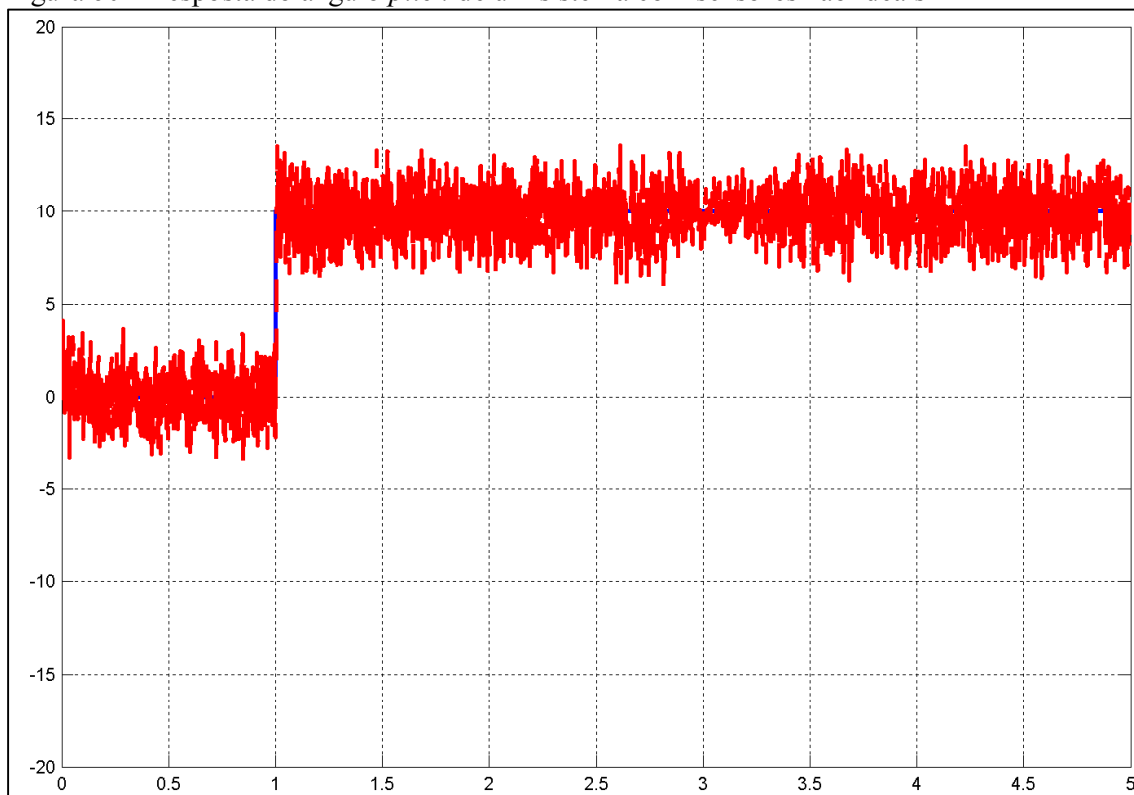


Fonte: Autoria Própria

4.2.3 *Pitch*

Para uma entrada implementada da forma da Figura 54, tem-se como resposta a curva vermelha apresentada na Figura 60, onde a curva azul representa o valor de *pitch* desejado. Verifica-se que o sistema alcança rapidamente o valor desejado, porém existe um ruído de magnitude de aproximadamente três graus.

Figura 60 - Resposta do ângulo *pitch* de um sistema com sensores não ideais

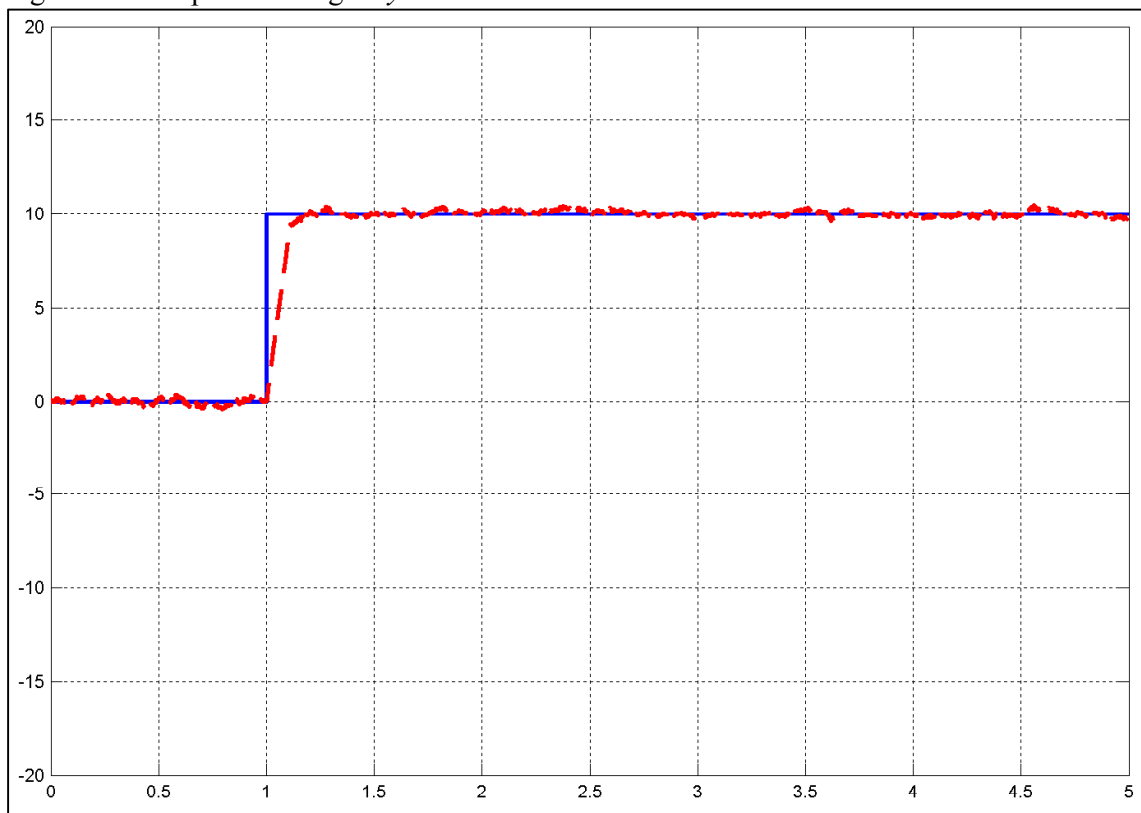


Fonte: Autoria Própria

4.2.4 Yaw

Para uma entrada implementada da forma da Figura 56, tem-se como resposta a curva vermelha apresentada na Figura 60, onde a curva azul representa o valor de *yaw* desejado. Verifica-se que o sistema alcança o valor desejado rapidamente, com baixa magnitude de *overshoot* e ruído menor que um grau.

Figura 61 - Resposta do ângulo *yaw* de um sistema com sensores não ideais



Fonte: Autoria Própria

5 CONCLUSÃO

Analisando-se os resultados obtidos no Capítulo 4 fica claro que a aplicação do método de controle PD *fuzzy* obteve sucesso e conseguiu estabilizar o modelo virtual implementado, controlando a altura e os ângulos ϕ , θ e ψ . Percebeu-se que os ângulos de *roll* e *pitch* respondem mais rapidamente ao controle, devido as constantes de inércia serem de menor magnitude, sendo a rotação do eixo X nomeada de *roll* e a do eixo y nomeada como *pitch*. O ângulo *yaw* representa a rotação no eixo z que possui momento de inércia maior que dos outros eixos, ficando claro então o porquê de sua resposta ser mais lenta.

Comparando os resultados do modelo com sensores ideais com o modelo com sensores não ideais, tem-se que as limitações dos sensores geram ruídos na medição dificultando o controle, causando então erros na resposta ao sinal de referência. Os ruídos tem maior magnitude para os ângulos de *roll* e *pitch*, devido a sua maior sensibilidade, por possuir menor inércia.

Uma solução seria a aplicação de filtros para os parâmetros medidos pelos sensores, diminuindo-se o ruído gerado pelo os mesmos, outro fator determinante na melhora da estabilidade do sistema como um todo é o aumento da taxa de atualização do controlador, ou seja aumentar a frequência dos seguradores de ordem zero, do controlador e dos sensores.

O controle da altura também foi alcançado nos dois modelos, sendo que modelo com sensores não ideais obteve um *overshoot* considerável, devido a presença de ruídos na medição das posições.

6 SUGESTÕES E TRABALHOS FUTUROS

Na seção 3.3 deste trabalho foram apresentados os componentes necessários para a construção um protótipo de *quadcopter*, baseado nesses componentes e no simulador desenvolvido ao longo deste trabalho, propõe-se a implementação do veículo e teste práticos inicialmente sendo controlado remotamente por telemetria, e posteriormente integralização do controle *onboard* tornando-o autônomo. Outro possível caminho é a utilização de *gps* e, ao se utilizar os *softwares* necessários fazer com que o *quadcopter* percorra um caminho pré-definido sem a necessidade de intervenção humana, conseguindo detectar objetos e evitar possíveis colisões tendo total conhecimento da sua posição no espaço e como deve efetuar a ação dada por um usuário.

REFERÊNCIAS

ALTUG, Erdinc; OSTROWSKI, James P.; MAHONY, Robert. Control of a quadrotor helicopter using visual feedback. In: **Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on**. IEEE, 2002. p. 72-77.

ARDEMA, Mark D. **Newton Euler dynamics**. springer, 2005.

BOUABDALLAH, Samir. **Design and control of quadrotors with application to autonomous flying**. Ecole Polytechnique Federale de Lausanne, 2007.

BRESCIANI, T. **Modelling, identification and control of a quadcopter helicopter**. 2008. Tese de Mestrado (Departamento de controle automático) – Universidade de Lund. 2008

CANETTA, Carlo et al. **Quad-rotor unmanned aerial vehicle**. Columbia University, Final Report, Engineering Design, MECE, v. 3410, 2007.

CASTILLO, Oscar; MELIN, Patricia. **Type-2 fuzzy logic**. Springer Berlin Heidelberg, 2008.

CHEN, Guanrong; PHAM, Trung Tat. **Introduction to fuzzy sets, fuzzy logic, and fuzzy control systems**. CRC press, 2000.

DE RODRIGUES QUEMEL, Pedro Henrique; SANTANA, Assis. **Concepção de um Veículo Aéreo não-tripulado do tipo quadrrrotor**. Universidade de Brasília. 2008

DE SOUSA, José Duarte Alves. **Development of unmanned aerial four-rotor vehicle**, Faculdade de Engenharia da Universidade do Porto 2011.

DJI. Disponível em: <<http://www.dji.com/>> Acesso em: 14 de out de 2013.

DOMINGUES, J.M.B. **Quadcopter prototype**. 2009. Tese de mestrado (Engenharia Mecânica) – Universidade Técnica de Lisboa. 2009

ENGEL, Jakob; STURM, Jurgen; CREMERS, Daniel. Camera-based navigation of a low-cost quadrocopter. In: **Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on**. IEEE, 2012. p. 2815-2821.

ENOMOTO, João Luiz Ferreira. **Controle de inclinação utilizando lógica fuzzy**. Universidade Federal do Paraná. 2010

GELLERSEN, Hans. The magazine archive includes every article published in Communications of the ACM for over the past 50 years. **Communications of the ACM**, v. 48, n. 3, p. 66. 2009.

Hobbyking. Disponível em:
<<http://www.hobbyking.com/hobbyking/store/index.asp>>. Acesso em: 14 de out. de 2013.

JADHAV, Hemant L. **Application of mechatronics in design and control of a quadcopter flying robot for aerial surveillance**. 2013.

KLIR, George J.; YUAN, Bo. **Fuzzy sets and fuzzy logic**. New Jersey: Prentice Hall, 1995.

KNAPP, B.R. **Fuzzy Sters and Pattern Recognition**. 2004. Universidade de Princeton Disponível em
<<http://www.cs.princeton.edu/courses/archive/fall07/cos436/HIDDEN/Knapp/fuzzy.htm>.> Acessado em 15 de out de 2013.

KRUSE, Rudolf; GEBHARDT, Joan E.; KLOWON, F. **Foundations of fuzzy systems**. John Wiley & Sons, Inc., 1994.

LAFLEUR, Karl et al. Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain-computer interface. **Journal of neural engineering**, v. 10, n. 4, p. 046003, 2013.

MENDEL, Jerry M. **Fuzzy logic systems for engineering: a tutorial**. **Proceedings of the IEEE**, v. 83, n. 3, p. 345-377, 1995.

NICE, Eryk Brian. **Design of a four rotor hovering vehicle**. 2004. Tese de Doutorado. Cornell University. 2004.

OURSLAND, Jacob. The design and implementation of a quadrotor flight controller using the quest algorithm. In: **Midwest Instruction and Computing Symposium**. 2010. p. 11.

PITMAN, David. **Collaborative micro aerial vehicle exploration of outdoor environments**. MASSACHUSETTS INST OF TECH CAMBRIDGE DEPT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE, 2010.

RIZOL, P.M.S.R **Introdução a lógica Difusa**. Universidade Estadual Paulista – Campus Guaratinguetá. Departamento de Engenharia Elétrica. 2008.

RODRIGUEZ, C.F.O. **Estudo de aplicação utilizando controladores fuzzy**. 2010. Trabalho de graduação (Graduação em Engenharia Elétrica) – Faculdade de Engenharia de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2010.

ROSS, Timothy J. **Fuzzy logic with engineering applications**. John Wiley & Sons, 2009.

SIMOES, Marcelo Godoy. **Introduction to fuzzy control**. Colorado School of Mines, Engineering Division, Golden, Colorado.2010.

SIVANANDAM, S. N.; SUMATHI, Sai; DEEPA, S. N. **Introduction to fuzzy logic using MATLAB**. Springer, 2007.

SKIBA, Thiago Lechinowski. **Desenvolvimento de um protótipo quadrotor utilizando lógica reconfigurável como plataforma para estudo de técnicas de controle**. 2013.

SØRENSEN, A. F. **Autonomous control of a miniature quadrotor following fast trajectories**. 2010. Tese de Mestrado (Engenharia de Controle) – Universidade de Aalborg, Aalborg, 2010.

STMicroelectronics. Disponível em: <<http://www.st.com/web/en/home.html>> Acesso em: 14 de out de 2013.

VIEIRA, J.C.S. **Plataforma móvel aérea quadrotor**. 2011. Tese de Mestrado (Engenharia Eletrônica Industrial e Computadores) – Universidade do Minho, 2011

APÊNDICE A – ESTRUTURA FIS DO CONTROLADOR

[System]

Name='ControladorFuzzy'

Type='mamdani'

Version=2.0

NumInputs=2

NumOutputs=1

NumRules=25

AndMethod='min'

OrMethod='max'

ImpMethod='min'

AggMethod='max'

DefuzzMethod='centroid'

[Input1]

Name='ErroPropocional'

Range=[-1 1]

NumMFs=5

MF1='N': 'trimf', [-0.6 -0.3 0]

MF2='NULO': 'trimf', [-0.3 0 0.3]

MF3='P': 'trimf', [0 0.3 0.6]

MF4='MN': 'trapmf', [-2.8 -1.2 -0.6 -0.3]

MF5='MP': 'trapmf', [0.3 0.6 1.2 2.8]

[Input2]

Name='ErroDerivativo'

Range=[-1 1]

NumMFs=5

MF1='N': 'trimf', [-0.6 -0.3 0]

MF2='NULO': 'trimf', [-0.3 0 0.3]

MF3='P': 'trimf', [0 0.3 0.6]

MF4='MN': 'trapmf', [-2.8 -1.2 -0.6 -0.3]

MF5='MP': 'trapmf', [0.3 0.6 1.2 2.8]

[Output1]

Name='NivelDeAtuação'

Range=[-1 1]

NumMFs=5

MF1='N': 'trimf', [-0.6 -0.3 0]

MF2='NULO': 'trimf', [-0.3 0 0.3]

MF3='P': 'trimf', [0 0.3 0.6]

MF4='MN': 'trapmf', [-2.64 -1.36 -0.6 -0.3]

MF5='MP': 'trapmf', [0.3 0.6 1.2 2.8]

[Rules]

4 4, 4 (1) : 1

4 1, 4 (1) : 1

4 2, 4 (1) : 1

4 3, 1 (1) : 1

4 5, 2 (1) : 1

1 4, 4 (1) : 1

1 1, 4 (1) : 1

1 2, 1 (1) : 1

1 3, 2 (1) : 1

1 5, 3 (1) : 1

2 4, 4 (1) : 1

2 1, 1 (1) : 1

2 2, 2 (1) : 1

2 3, 3 (1) : 1

2 5, 5 (1) : 1

3 4, 1 (1) : 1

3 1, 2 (1) : 1

3 2, 3 (1) : 1

3 3, 5 (1) : 1

3 5, 5 (1) : 1

5 4, 2 (1) : 1

5 1, 3 (1) : 1

5 2, 5 (1) : 1

5 3, 5 (1) : 1

5 5, 5 (1) : 1