

Controlo de Quadcopter

PROJETO DE MESTRADO

Rui Miguel Freitas Martins

MESTRADO EM ENGENHARIA ELETROTÉCNICA-TELECOMUNICAÇÕES



UNIVERSIDADE da MADEIRA

A Nossa Universidade

www.uma.pt

fevereiro | 2019

Controlo de Quadcopter

PROJETO DE MESTRADO

Rui Miguel Freitas Martins

MESTRADO EM ENGENHARIA ELETROTÉCNICA-TELECOMUNICAÇÕES

ORIENTAÇÃO

João Dionísio Simões Barros

Resumo

O controlo de voo dos *quadcopters* tem grande interesse devido ao aumento de número de aplicações que recorrem a este tipo de aeronaves. As vantagens que o recurso a 4 conjuntos de motor e hélice como meio de propulsão apresenta em relação a outro tipo de *drones*, são a capacidade de se manter imóvel no ar e a capacidade de descolar e aterrar na vertical. Ao mesmo tempo que um *quadcopter* tem estas vantagens, o sistema não linear tem 6 graus de liberdade, com apenas 4 atuadores o que coloca assim desafios no controlo de orientação e posição.

O objetivo principal deste projeto consistiu em estudar a dinâmica do *quadcopter*, e aplicar diversas técnicas de controlo que permitiram controlar a orientação e altura do *drone*, tanto em simulação como a nível experimental. As técnicas de controlo foram a técnica PID, a técnica Fuzzy-PID e a técnica de *backstepping*. Das três técnicas utilizadas a que teve o desempenho mais robusto foi a técnica de *backstepping* e a que exigiu menos processamento computacional foi a técnica PID.

A nível experimental efetuou-se a montagem do *quadcopter* e implementou-se um controlador de voo que recorreu a controladores PID para controlar a orientação, com resultados que se comportaram como na simulação.

Palavras chave: *Quadcopter*, *drone*, controlo, PID, Fuzzy-PID, *Backstepping*.

Abstract

The flight control of quadcopters is of great interest due to the increase in the number of applications that use this type of aircraft. The advantages of the 4 motor and propeller sets as a means of propulsion, relative to other type of drones, are the ability to hover and the ability to take off and land vertically. While it has these advantages, the non-linear system also has 6 degrees of freedom with only 4 actuators, which thus poses challenges in the control of orientation and position.

This project studied the dynamics of the quadcopter and applied several control techniques that allow controlling the orientation and height of the drone, both in simulation and in practice. The control techniques discussed in the simulation were the PID technique, the Fuzzy-PID technique and the backstepping technique. Of the three techniques used, the one with the most robust performance was the backstepping technique, and the one that required less digital processing effort was the PID technique.

In practice, the quadcopter was assembled, and a flight controller was implemented that used PID controllers to control the orientation. We were able to implement orientation control, with results similar to the simulations.

Keywords: Quadcopter, Drone, control, PID, Fuzzy-PID, Backstepping.

Agradecimentos

Um agradecimento especial ao meu orientador, Prof. Dionísio Barros, por todo o apoio, disponibilidade, motivação e conhecimentos transmitidos.

Quero agradecer ao meu colega Lúri Viveiros por todo o tempo que passou neste projeto e pelas sugestões que me transmitiu.

À minha família por todo apoio dado e por me terem proporcionado os meios para que esta etapa fosse alcançável.

Aos meus colegas, Dino Vasconcelos, Pedro Nunes, Sérgio Vieira, Rafael Velosa, Duarte Alves, Rodrigo Teixeira e Jonathan Cavaleiro pelos bons momentos passados e pelo apoio e dicas que deram.

Aos engenheiros Filipe Santos e Jorge Lopes pelos conhecimentos passados e a assistência prestada nos laboratórios.

Lista de abreviaturas

AFHDS – *Automatic Frequency Hoping Digital System*

ANAC – *Autoridade Nacional de Aviação Civil*

BLDC – *Brushless Direct Current*

DSP – *Digital Signal Processor*

ESC – *Electronic Speed Controller*

FAA – *Federal Aviation Administration*

GPS – *Global Positioning System*

I²C – *Inter-Integrated Circuit*

IMU – *Inertial Measurement Unit*

LIDAR – *Light Detection And Ranging*

LiPO – *Lithium Polimer*

MEMS – *Microelectromechanical Systems*

MOSFET – *Metal Oxide Semiconductor Field Effect Transistor*

PID – *Proporcional Integral e Derivativo*

PWM – *Pulse Width Modulation*

rpm – *rotações por minuto*

UAV – *Unmanned Aerial Vehicle*

Wi-Fi – *Wireless Fidelity*

Lista de símbolos

$A_{x,y,z}$ – Aceleração do acelerómetro

a_x – Aceleração no eixo do x

a_y – Aceleração no eixo do y

a_z – Aceleração no eixo do z

C_q – Coeficiente aerodinâmico do momento das hélices

C_t – Coeficiente aerodinâmico da impulsão das hélices

F_i – Força de impulsão produzida pelo motor i

I_{xx} – Momento de inércia sobre o eixo do x

I_{yy} – Momento de inércia sobre o eixo do y

I_{zz} – Momento de inércia sobre o eixo do z

J_r – Inércia dos motores

K_d – Ganho diferencial

K_f – Constante aerodinâmica da força de impulsão

K_i – Ganho integral

K_M – Constante aerodinâmica do binário

K_p – Ganho proporcional

M_G – Momentos giroscópicos dos motores

M_i – Momento produzido pelo motor i

T_{ref} – Sinal de comando do acelerador

x_{ref} – Valor de referência da posição x

y_{ref} – Valor de referência da posição y

z_{ref} – Altura de referência

θ_A – Ângulo *pitch* obtido através da aceleração do acelerómetro

θ_G – Ângulo *pitch* medido pelo giroscópio

θ_K – Ângulo *pitch* obtido pelo filtro complementar

θ_{ref} – Ângulo *pitch* de referência

ϕ_A – Ângulo *roll* obtido através da aceleração do acelerómetro

ϕ_G – Ângulo *roll* medido pelo giroscópio
 ϕ_K – Ângulo *roll* obtido pelo filtro complementar
 ϕ_{ref} – Ângulo *roll* de referência
 ψ_G – Ângulo *yaw* medido pelo giroscópio
 ψ_K – Ângulo *yaw* obtido pelo filtro complementar
 ψ_{ref} – Ângulo *yaw* de referência
 Ω_i – Velocidade de rotação do motor *i*
 Ω_r – Velocidade relativa dos motores
 $[F_B]$ – Vetor das forças não gravitacionais que atuam sobre o *quadcopter*
 $[M_{Bx}]$ – Total dos momentos a atuarem sobre o *quadcopter* com estrutura em **x**
 $[M_B]$ – Total dos momentos que atuam sobre o *quadcopter* no referencial do corpo
 $[R_r]$ – Matriz de rotação inversa
 $[R_x]$ – Matriz de rotação elementar em *x*
 $[R_y]$ – Matriz de rotação elementar em *y*
 $[R_z]$ – Matriz de rotação elementar em *z*
 $[U_x]$ – Vetor dos sinais de controlo para o *quadcopter* com estrutura em **x**
 $[J]$ – Matriz de inércia do *quadcopter*
 $[R]$ – Matriz de rotação
 $[r]$ – Vetor da posição do *quadcopter*
 $[U]$ – Vetor dos sinais de controlo
 $[X]$ – Vetor das variáveis de estado
 $[\eta]$ – Vetor dos ângulos de Euler
 $[\omega]$ – Vetor das velocidades angulares do corpo
A – Área das hélices
e – Erro entre o valor de referência e o valor atual
g – Constante da aceleração gravítica, $g = 9,81 \text{ m/s}^2$
l – Distância do motor ao centro de massa do *quadcopter*
m – Massa do *quadcopter*

r – Raio das hélices

θ – Ângulo *pitch*

ρ – Densidade do ar

ϕ – Ângulo *roll*

ψ – Ângulo *yaw*

Índice

Resumo	I
Abstract	II
Agradecimentos	III
Lista de abreviaturas	IV
Lista de símbolos	V
Índice	VIII
1 Introdução	1
1.1 Motivação	1
1.2 Objetivos	1
1.3 Organização e conteúdo	2
2 Revisão da bibliografia	3
2.1 Descrição do quadcopter	3
2.2 Princípio de funcionamento	3
2.2.1 Ângulos de Euler.....	6
2.2.2 Trabalhos académicos	8
2.2.3 Drones comerciais	10
2.3 Arquitetura típica de um quadcopter	12
2.4 Sensores	13
2.4.1 Acelerómetro de três eixos	13
2.4.2 Giroscópio de três eixos	15
2.4.3 Fusão de sensores	16
2.5 Motores	17
2.6 Telecomando	22
2.7 Controlo	23
2.8 Legislação	24
3 Modelos e controlo do drone	27
3.1 Cinemática do quadcopter	27
3.1.1 Equações do movimento rotacional.....	28
3.1.2 Equações do movimento translacional	31
3.1.3 Efeitos aerodinâmicos	32
3.2 Modelo em variáveis de estado	33
3.2.1 Vetor das entradas de controlo.....	33
3.2.2 Equação do movimento rotacional	34
3.2.3 Equação do movimento translacional	35
3.2.4 Representação em variáveis de estado	36

3.3	Técnicas de controlo.....	37
3.3.1	Controlador PID	39
3.3.2	Controlador Fuzzy-PID	41
3.3.3	Controlador <i>Backstepping</i>	46
3.3.3.1	Controlo de <i>roll</i>	47
3.3.3.2	Controlo de <i>pitch</i>	48
3.3.3.3	Controlo de <i>yaw</i>	49
3.3.3.4	Controlo de altitude	50
4	<i>Simulação do controlo do drone</i>	51
4.1	Implementação das equações do <i>drone</i>.....	51
4.1.1	Modelo em linguagem MATLAB	51
4.1.2	Modelo em diagrama de blocos	53
4.2	Implementação dos controladores	55
4.2.1	Controlador do modelo em linguagem MATLAB.....	55
4.2.2	Controlador do modelo em diagrama de blocos.....	57
4.3	Simulação do controlo de altura e orientação.....	58
4.3.1	Controlo com a técnica PID	59
4.3.2	Controlo com a técnica Fuzzy-PID	63
4.3.3	Controlo com a técnica de <i>Backstepping</i>	65
4.4	Comparação de resultados entre as técnicas	68
5	<i>Verificação experimental do quadcopter</i>.....	71
5.1	Processador.....	71
5.2	Telecomando.....	71
5.3	Sensores.....	74
5.4	Reguladores de velocidade.....	75
5.5	Estrutura do <i>quadcopter</i>.....	77
5.6	Verificação do controlo de voo	78
5.6.1	Controlador de voo.....	79
5.6.2	Programa completo	82
6	<i>Conclusões</i>	85
6.1	Conclusões gerais	85
6.2	Trabalhos futuros	87
7	<i>Referências</i>	89
<i>Anexo A – Momento de inércia do quadcopter</i>.....		93
A.1	Motores	94
A.2	ESC	95
A.3	Núcleo central	96
A.4	Braços	97
A.5	Matriz de inércia.....	98

Anexo B – Detecção da inclinação através do acelerómetro de três eixos	101
Anexo C – Modelos de simulação do quadcopter.....	105
C.1 Modelo em linguagem MATLAB	105
C.1.1 Código da dinâmica do <i>quadcopter</i>	105
C.2 Modelo em diagrama de blocos	108
C.3 Controladores	108
C.3.1 Controlador PID	108
C.3.2 Controlador Fuzzy-PID	109
C.3.3 Controlador <i>Backstepping</i>	110

1 Introdução

Os *drones* são aeronaves não tripuladas cujo o voo é controlado autonomamente através de *software* ou remotamente por uma pessoa, com auxílio a sensores a bordo para a estimação da orientação e posição da aeronave. Os *quadcopters* pertencem à classe dos *drones* e são caracterizados por utilizarem quatro conjuntos de motores e hélices para se deslocarem.

Neste capítulo apresenta-se a motivação para o desenvolvimento deste projeto, os objetivos que se pretendeu atingir e a forma como está estruturado o trabalho.

1.1 Motivação

Com o desenvolvimento tecnológico a nível de sensores, processadores, motores e baterias, cada vez mais as aeronaves do tipo *quadcopter* são utilizadas para vários fins. Apesar da popularidade deste tipo de *drones*, a área de controlo ainda apresenta desafios porque os *quadcopters* são sistemas não lineares com seis graus de liberdade e apenas quatro atuadores. Como existem mais graus de liberdade do que atuadores, o *quadcopter* é um sistema sub-atuado e as dificuldades do controlo advêm do acoplamento das não linearidades com os graus de liberdade e os atuadores. Como o número e a complexidade das aplicações dos *quadcopters* têm vindo a crescer, as técnicas de controlo também têm que melhorar de forma a tornar o desempenho melhor e mais versátil. As técnicas de controlo lineares permitem implementar com maior facilidade um controlo de voo em torno do ponto em que o *drone* está a pairar no ar. No entanto, com melhores técnicas de modelagem e com recurso a maior capacidade computacional, métodos de controlos não lineares permitem obter melhores desempenhos e sistemas mais robustos.

A orientação do *quadcopter* é fulcral no controlo de voo porque sem uma orientação bem controlada é impossível deslocar a aeronave no espaço com segurança. O controlo de posição do *drone* está dependente do controlo de orientação.

Um aspeto importante no controlo de *quadcopters* são os sensores utilizados para a estimação da orientação. Esses sensores pertencem à IMU (*Inertial Measurement Unit*) do *quadcopter*. A fusão da informação de sensores, como o giroscópio e o acelerómetro, permite eliminar algumas das suas limitações e determinar a orientação da aeronave.

1.2 Objetivos

Este projeto teve como objetivos principais estudar a dinâmica do *quadcopter* e implementar, em ambiente de simulação e na prática, o controlo de orientação. No contexto deste projeto, os objetivos a alcançar são os seguintes:

- Estudar os ângulos que descrevem a orientação de aeronaves num espaço tridimensional;
- Estudar os elementos que constituem um *quadcopter*: sensores para a estimação da orientação, reguladores de velocidade e sistema de controlo remoto;
- Demonstrar as equações que descrevem a dinâmica do *quadcopter*;
- Apresentar as técnicas de controlo PID, Fuzzy-PID e *backstepping* para o controlo de altura e orientação;
- Implementar em ambiente Simulink do MATLAB o modelo matemático do *drone* e avaliar a resposta do sistema para cada técnica de controlo;
- A nível experimental, proceder à montagem do *quadcopter* e ao desenvolvimento do respetivo controlador de voo.

1.3 Organização e conteúdo

Este relatório está subdividido em 6 capítulos. Após este capítulo inicial, no capítulo 2 referente à revisão da bibliografia apresenta-se uma descrição do *quadcopter*, o seu princípio de funcionamento, quais os sensores utilizados para auxiliar o voo, que tipo de motores são utilizados e como é que são controlados, os sistemas de controlo remoto, os objetivos do controlo e por último, os traços gerais da legislação portuguesa em relação ao voo deste tipo de aeronaves.

No capítulo 3 apresenta-se o modelo matemático que descreve a dinâmica do *quadcopter*, o modelo em variáveis de estado, e as técnicas de controlo de altura e de orientação.

No capítulo 4 descreve-se a implementação do modelo matemático do *quadcopter* no Simulink e apresentam-se os resultados da simulação para as diferentes técnicas de controlo. No final deste capítulo, comparam-se os resultados da simulação e avalia-se o desempenho de cada técnica de controlo.

No capítulo 5 apresenta-se todo o procedimento realizado para a conceção do *quadcopter* e explica-se o algoritmo do controlo de voo implementado.

Por último, no capítulo 6, apresentam-se as conclusões deste projeto e as sugestões para trabalhos futuros.

2 Revisão da bibliografia

2.1 Descrição do quadcopter

Os *quadcopters* são aeronaves não tripuladas (UAV – *Unmanned Aerial Vehicle*) baseadas em sistemas multirotores, que são muito utilizados em áreas como a da fotografia, militar, vigilância, busca e salvamento e transportes. As principais características dos *quadcopters*, que tornam atrativa a sua utilização nas diversas áreas, são a capacidade de descolagem e aterragem vertical e a capacidade de pairar no ar. Com os recentes avanços tecnológicos na eletrônica, nas baterias e nos motores, os *quadcopters* estão a tornar-se mais ágeis, conseguindo transportar cargas mais pesadas e com tempos de voo maiores [1].

Num *quadcopter* tem-se quatro conjuntos motor e hélice, todos posicionados à mesma distância, como apresentado na Figura 2.1. Desta forma tem-se um sistema de seis graus de liberdade com quatro variáveis controláveis (velocidade de cada um dos motores) [2].

Um *quadcopter* é um sistema com seis graus de liberdade porque as coordenadas de XYZ descrevem a sua posição e os ângulos de Euler θ , ϕ e ψ descrevem a sua orientação.

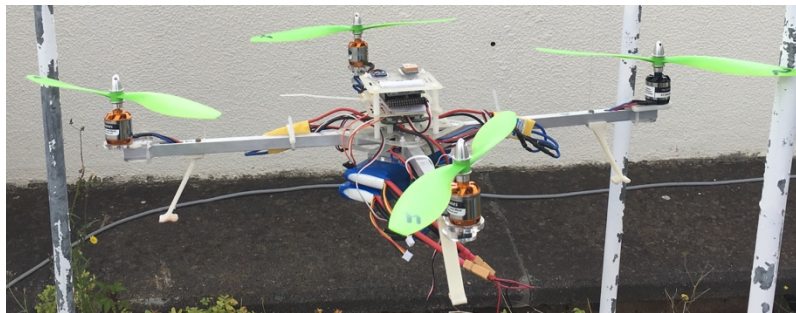


Figura 2.1 - Exemplo de um *quadcopter*, *Arducopter*.

Sem controlo, os *quadcopters* são sistemas inerentemente instáveis, tornando-se necessário utilizar diferentes sensores, como giroscópios, acelerómetros e barómetros para estimar a posição e a orientação do *quadcopter*. O controlo de orientação e posição é efetuado através do aumento e diminuição da velocidade de cada um dos motores [2].

2.2 Princípio de funcionamento

A velocidade de rotação de cada motor do *quadcopter* produz uma força de elevação e um binário. Dois motores rodam no sentido dos ponteiros do relógio e os outros dois rodam no sentido oposto, equilibrando assim o binário total do sistema, Figura 2.2 [2].

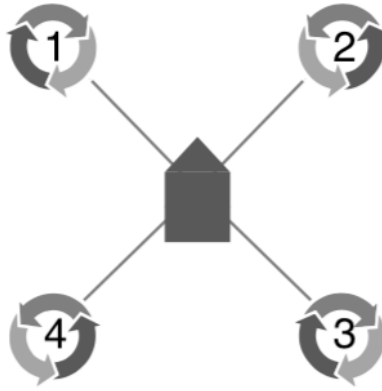


Figura 2.2 - Exemplo de disposição do sentido de rotação da velocidade dos motores [2].

Para que o *quadcopter* consiga descolar, o somatório das forças geradas pela rotação das hélices tem de ser superior ao peso do *quadcopter* e para que consiga pairar no ar, essas forças têm de igualar o peso do *quadcopter*.

A estrutura de um *quadcopter* pode ter diversas configurações, sendo as mais comuns a configuração em **x** e a configuração em **+**. A Figura 2.3 ilustra essas duas configurações [3].

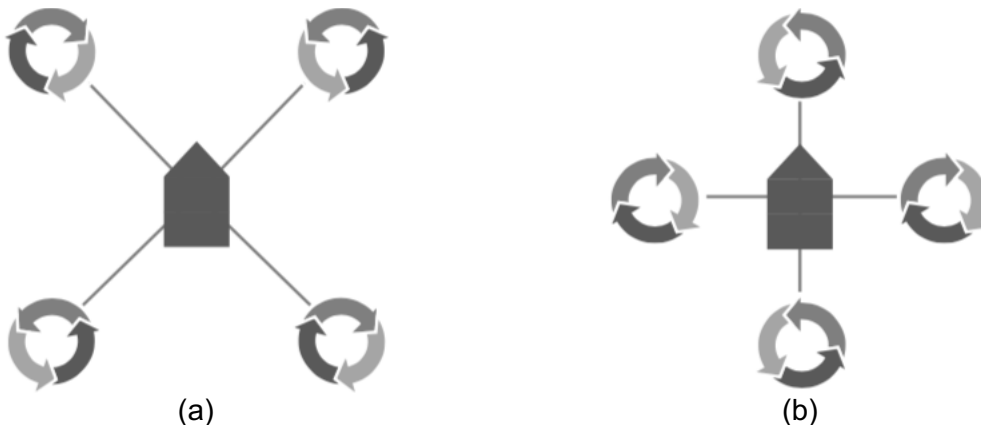


Figura 2.3 - Configurações em **x** (a) e em **+** (b). Nesta figura as setas representam a direção de rotação das hélices e as linhas e o centro representam a estrutura do *quadcopter* [3].

Ambas as configurações apresentam vantagens e desvantagens. A configuração em **x** é simétrica, de fácil construção e ideal para a utilização de câmaras, pois nesta configuração as hélices do *quadcopter* não são captadas pelo plano da câmara. A desvantagem desta configuração é que torna mais complexo o modelo matemático e o controlo do *quadcopter*. Na configuração em **+** o modelo matemático e o controlo são mais simples, mas não é ideal para o uso de câmaras [3].

O movimento no espaço por um corpo rígido como o de um *quadcopter* pode ser decomposto em duas componentes: movimentação do centro de massa e movimentação em torno do centro de massa. A deslocação no espaço é dada por seis graus de liberdade, sendo três translações (movimentação do centro de massa) e três rotações (movimentação em torno do centro de massa) ao longo de três eixos. No caso de um *quadcopter*, os seis graus de liberdade são controlados ajustando a

velocidade de rotação de cada um dos quatro motores. Estes seis graus de liberdade correspondem aos movimentos na vertical, na lateral, para a frente e para trás, e às rotações em *roll*, *pitch* e *yaw*. O movimento *yaw* dá-se quando o *quadcopter* gira em torno de si próprio, este movimento é causado pelo binário gerado por cada um dos motores. Quando todos os motores têm a mesma velocidade angular, o binário total fica equilibrado, logo não existe rotação *yaw*. Quando os motores não têm a mesma velocidade, o binário total fica desequilibrado, fazendo com que o *quadcopter* rode em torno de si próprio. Para que o *quadcopter* se mova na vertical, todos os motores têm de ter a mesma velocidade e variando o valor desta o *quadcopter* irá subir, descer ou pairar no ar.

Um *quadcopter* é considerado um sistema de quatro entradas e seis saídas, e de forma a conseguir controlá-lo é necessário considerar o *quadcopter* um corpo rígido com uma estrutura simétrica. Dependendo dos valores da velocidade de rotação de cada motor, é possível identificar quatro movimentos básicos do *quadcopter*, que estão representados da Figura 2.4 até à Figura 2.7 (*quadcopter* na configuração em +).

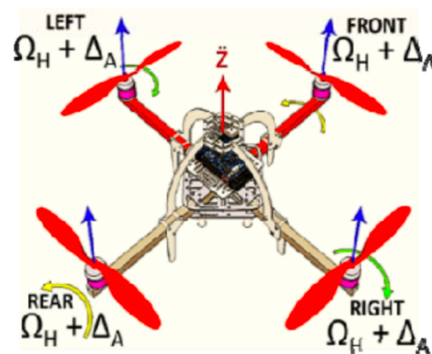


Figura 2.4 - Movimento de elevação [1].

No movimento de elevação, Figura 2.4, todos os motores aumentam a velocidade ($\Omega_H + \Delta_A$) com a mesma aceleração para que a força produzida por estes seja superior ao peso do *quadcopter*, fazendo com que este se eleve.

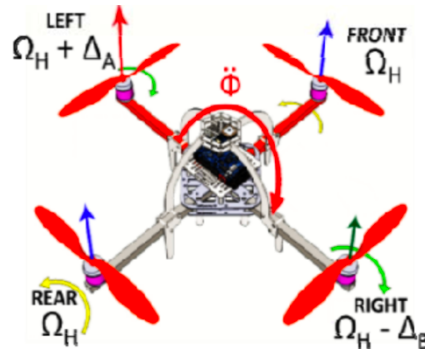


Figura 2.5 - Movimento *roll* [1].

No movimento de *roll*, Figura 2.5, os motores “*Front*” e “*Rear*” giram à mesma velocidade (Ω_H), enquanto que os motores “*Left*” e “*Right*” rodam a velocidades

diferentes ($\Omega_H + \Delta_A$ e $\Omega_H - \Delta_B$ respectivamente). A diferença de valor entre Δ_A e Δ_B controlam o sentido do movimento *roll*, fazendo desta forma com que o *quadcopter* se movimente no espaço para a esquerda ou para a direita.

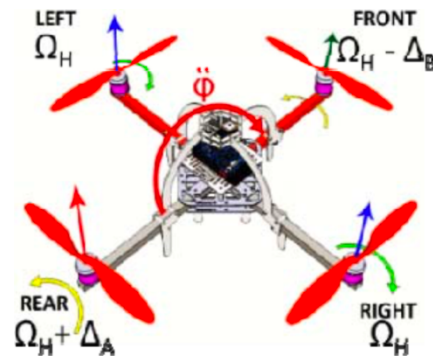


Figura 2.6 - Movimento *pitch* [1].

O movimento *pitch*, Figura 2.6, é semelhante ao movimento *roll*, mas neste caso os motores “*Left*” e “*Right*” rodam à mesma velocidade enquanto que os motores “*Front*” e “*Rear*” têm velocidades diferentes. A diferença de velocidade entre estes dois motores irá controlar o movimento *pitch*, fazendo assim com que o *quadcopter* se mova para a frente ou para trás no espaço.

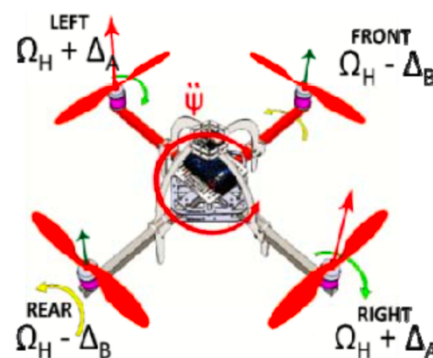


Figura 2.7 - Movimento *yaw* [1].

No movimento *yaw*, Figura 2.7, os pares de motores “*Front*”/“*Rear*” e “*Left*”/“*Right*” rodam a velocidades diferentes. Essa diferença de velocidades entre pares (Δ_A e Δ_B) controla o sentido do movimento *yaw*. Neste movimento o *quadcopter* gira em torno de si próprio.

2.2.1 Ângulos de Euler

A orientação de um corpo rígido num espaço tridimensional pode ser descrita pelos ângulos de Euler. Estes ângulos permitem descrever a orientação de um referencial em relação a outro referencial, como também transformar as coordenadas de um ponto num referencial para as coordenadas do mesmo ponto noutra referencial [4].

Na Figura 2.8 estão ilustrados os ângulos de Euler, e também o referencial fixo da terra e o referencial do *quadcopter*.

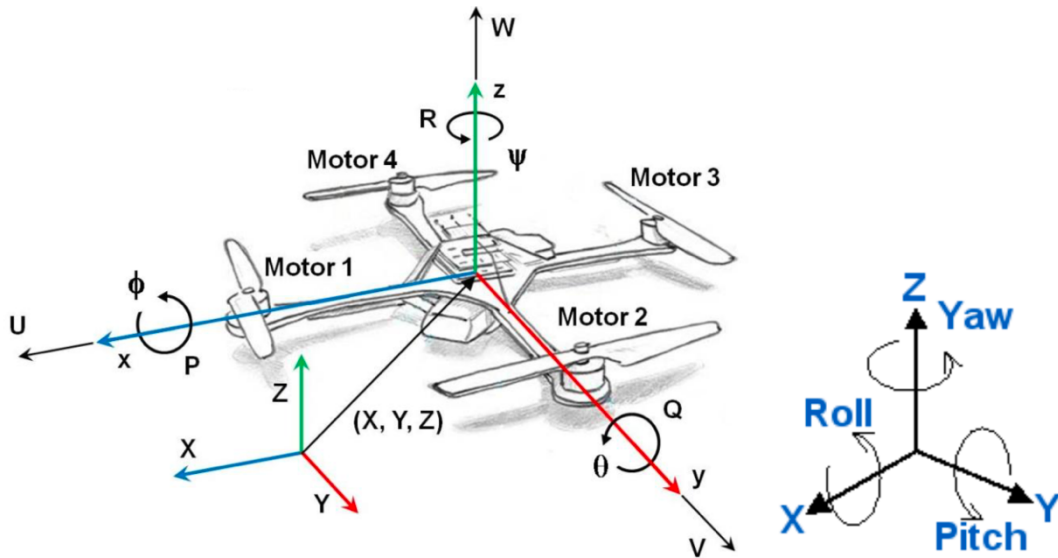


Figura 2.8 - Ângulos de Euler e os referenciais da terra e do *quadcopter* [5].

Uma rotação em torno do eixo do x é denominada de *roll* (ϕ), em torno de y é *pitch* (θ) e em torno de z é *yaw* (ψ). Para cada uma destas rotações, o *quadcopter* tem um momento de inércia associado a cada eixo. O anexo A descreve como é que é obtido o momento de inércia para cada um dos eixos do *quadcopter*.

A orientação de um corpo rígido num espaço tridimensional pode ser dada através de três rotações elementares (R_x , R_y e R_z). Estas rotações começam a partir de uma orientação conhecida e são descritas através das seguintes matrizes de rotação [4]:

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi) & c(\phi) \end{bmatrix}; \quad (2.1)$$

$$R_y(\theta) = \begin{bmatrix} c(\theta) & 0 & s(\theta) \\ 0 & 1 & 0 \\ -s(\theta) & 0 & c(\theta) \end{bmatrix}; \quad (2.2)$$

$$R_z(\psi) = \begin{bmatrix} c(\psi) & -s(\psi) & 0 \\ s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.3)$$

em que $c(\phi) = \cos(\phi)$, $s(\phi) = \sin(\phi)$, $c(\theta) = \cos(\theta)$, $s(\theta) = \sin(\theta)$, $c(\psi) = \cos(\psi)$, $s(\psi) = \sin(\psi)$. Com isto, a relação entre as coordenadas do referencial fixo e as coordenadas no referencial do corpo rígido é dada pela matriz de rotação $R_{zyx}(\phi, \theta, \psi)$:

$$R_{zyx}(\phi, \theta, \psi) = R_z(\psi) \cdot R_y(\theta) \cdot R_x(\phi)$$

$$= \begin{bmatrix} c(\theta)c(\psi) & s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi) \\ c(\theta)s(\psi) & s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) \\ -s(\theta) & s(\phi)c(\theta) & c(\phi)c(\theta) \end{bmatrix} \quad (2.4)$$

Na dinâmica do *quadcopter* existem grandezas que são representadas em diferentes referências devido ao facto de estarem dependentes ou não da orientação deste. A direção da força produzida pelos motores está associada à orientação do *drone*, logo esta é representada no referencial do *quadcopter*. A força gravítica exercida sobre a aeronave é indiferente à orientação desta, logo esta grandeza é representada no referencial fixo da terra. Com isto, a matriz de rotação R_{zyx} de (2.4) permite passar grandezas do referencial do *quadcopter* para o referencial fixo.

2.2.2 Trabalhos académicos

Ao longo das últimas duas décadas vários trabalhos de investigação interessaram-se no estudo dos *quadcopters*. Os trabalhos centraram-se principalmente na dinâmica dos *quadcopters*, no controlo (linear e não linear) da orientação, no controlo da trajetória e no controlo com perturbações.

Em 2001, Mistler *et al.* [6] propuseram um modelo dinâmico não linear para um helicóptero *quadrotor*. Também propuseram através de uma transformação de coordenadas, linearizar o modelo de forma a facilitar o seu controlo. A estabilidade e a robustez do controlador proposto nesse trabalho foram analisadas em simulação com a inclusão de perturbações como vento, turbulência e parâmetros incertos. Com esse trabalho chegaram à conclusão que a linearização do modelo tornou o controlo mais exigente a nível de processamento, sugerindo a utilização de técnicas de controlo com base nas funções de *Lyapunov* para poder tirar partido das não linearidades do sistema.

Madani e Benallegue em 2006 [7] propuseram um controlador baseado nas técnicas de *backstepping* para controlar a trajetória e o movimento *yaw* de um *quadcopter*. Nesse trabalho o *quadcopter* foi representado por três subsistemas interligados, sendo o primeiro um subsistema que descreve a relação dinâmica entre a posição horizontal da aeronave e os ângulos de *pitch* e *roll*, o segundo subsistema descreve o comportamento dinâmico da posição vertical e do ângulo de *yaw*, e o por último, o terceiro subsistema descreve a dinâmica dos quatro conjuntos motor e hélice. O controlador desenvolvido baseou-se na teoria de estabilidade de *Lyapunov*, e foi testado em simulação e em laboratório. Em simulação, o controlador proposto conseguiu controlar a trajetória e o ângulo de *yaw*, na verificação experimental foram controlados com sucesso a posição vertical e o ângulo de *yaw*.

Em 2013 Dong *et al.* [8] descreveram a dinâmica do *quadcopter* através de conceitos da aerodinâmica e desenvolveu um conjunto de controladores PID para realizar o controle da orientação, altitude e trajetória. Para efetuarem os testes dos controladores, utilizaram o *quadcopter Hummingbird*, Figura 2. 9, da *Ascending Technologies, GmbH*, que permite desenvolver os algoritmos em Matlab/Simulink e corrê-los na aeronave. Com os testes de trajetória realizados, verificaram que os controladores propostos seguiram as referências com um erro máximo de 0,5 m e com um *overshoot* máximo de 15 %.



Figura 2. 9 - *Quadcopter Hummingbird* [8].

Em 2015 Tosun *et al.* [9] desenvolveram uma técnica de controle baseada no LQR (*Linear Quadratic Regulator*) para controlar a posição em x, y e z e os ângulos de *roll*, *pitch* e *yaw*. Para testar o controlador, utilizaram o *quadcopter Qball-X4* que tem uma interface MATLAB/Simulink, Figura 2. 10. Os ganhos do controlador foram escolhidos tendo em conta a relação de compromisso entre a velocidade de resposta e o esforço computacional do controle.



Figura 2. 10 - *Quadcopter Qball-X4* [9].

Em 2017 Kuantama *et al.* [10] comparam duas técnicas de controle de orientação e posição do *quadcopter*, PID e Fuzzy-PID. Em simulação, controlando os ângulos *roll* e *pitch* e efetuando uma trajetória circular, os melhores resultados foram obtidos com o controlador baseado no Fuzzy-PID. No controle com PID, apenas foi necessário

utilizar as componentes proporcional e derivativa para ter um voo estável, e com isto, na trajetória circular o desvio máximo foi de 0,12 m. No controlador Fuzzy-PID, o desvio máximo foi de 0,05 m. Na presença de perturbações como o vento, também verificaram que o Fuzzy-PID teve melhor resposta do que o PID normal.

2.2.3 Drones comerciais

Foi no ano de 2006 que pela primeira vez uma agência governamental nos Estados Unidos, a *Federal Aviation Administration (FAA)*, emitiu uma licença para *drones* comerciais. Nesse mesmo ano, os *drones* começaram a ser utilizados por instituições governamentais para a vigilância de fronteiras, assistência em catástrofes e no combate a incêndios. Nas empresas, os primeiros *drones* foram utilizados para inspecionar oleodutos e pulverizar pesticidas em explorações agrícolas [11].

O primeiro *drone* comercial a suscitar o interesse dos consumidores comuns foi o *Parrot AR.Drone*, Figura 2. 11. O motivo que tornou este *drone* popular foi o facto de poder ser controlado por um *smartphone* IOS ou Android. Este equipamento utilizava Wi-Fi para receber o sinal de comando do *smartphone*, e tinha capacidade para 12 minutos de voo a uma velocidade de 5 m/s. Para controlar a orientação do *quadcopter*, este equipamento utiliza em conjunto: um acelerómetro de 3 eixos, um giroscópio de 2 eixos e um outro giroscópio de um eixo (*yaw*). Para o controlo de posição (apenas na vertical) o *drone* está equipado com um altímetro (sensor baseado em ultra-sons) com um alcance máximo de 6 m. Os motores utilizados são motores *brushless* de 15 W. As principais funcionalidades deste *drone* eram a capacidade de fotografar e filmar tanto na vertical como na horizontal (possui duas câmaras, uma na parte da frente e outra na parte de baixo) e a capacidade de automaticamente aterrar em segurança quando a ligação entre o *drone* e o *smartphone* era interrompida [12].



Figura 2. 11 - *Parrot AR.Drone* [13].

Em 2013 a *Amazon* anunciou o projeto *Amazon Prime Air*, ainda em desenvolvimento, que tem como objetivo utilizar *quadcopters* para efetuar a entrega de mercadorias (até aproximadamente 2,3 kg de peso) aos clientes em menos de 30 minutos [14]. A Figura 2. 12 apresenta um *quadcopter* que está a ser desenvolvido pela *Amazon*.



Figura 2. 12 - Um dos quadcopters desenvolvidos no *Amazon Prime air* [14].

Atualmente a marca de *drones* comerciais com maior fatia de mercado é a DJI. O modelo mais comercializado até à data de hoje é o *DJI Phantom 4*, Figura 2. 13 [15]. As principais características deste *drone* são: tempo de voo de aproximadamente 28 minutos; velocidade máxima de subida de 6 m/s; velocidade máxima de descida de 4 m/s; velocidade máxima horizontal de 20 m/s; tolerância a vento com velocidades até 10 m/s; controlo de posição através de GPS; controlo de posição através do sistema de captura de imagens [16].



Figura 2. 13 - *DJI Phantom 4* [17].

Outra área em que está a surgir o interesse na utilização dos *drones* é no transporte de pessoas. Apesar de ainda não haver um *drone* a ser comercializado capaz de transportar uma pessoa, algumas empresas já estão a investir nesta área de *drones*. A empresa *Beijing Yi-Hang Creation Science & Technologie Co* está a desenvolver um *drone*, Figura 2. 14, capaz de transportar um passageiro até 100 kg.



Figura 2. 14 - *Ehang 184* [18].

O *Ehang 184* apesar de ter uma estrutura semelhante à de um *quadcopter*, este em cada um dos braços possui dois motores, ou seja, no total são oito motores. Esta configuração permite ter uma aeronave com a agilidade de um *quadcopter* e a redundância de um *octocopter* [19]. O fabricante indica que terá um tempo de voo de

23 minutos, e uma distância máxima de viagem de aproximadamente 16 Km. O passageiro não terá que controlar a aeronave, pois está equipada com um sistema de voo autónomo. A legislação relativa a veículos aéreos não tripulados de muitos países obriga que todos os voos têm de ser feitos com linha de vista entre a aeronave e o comando, colocando assim um entrave para a utilização deste tipo de veículos no transporte de pessoas [19].

2.3 Arquitetura típica de um *quadcopter*

Para compreender melhor a constituição de um *quadcopter* está representado na Figura 2. 15 o diagrama com os constituintes principais de um *quadcopter*.

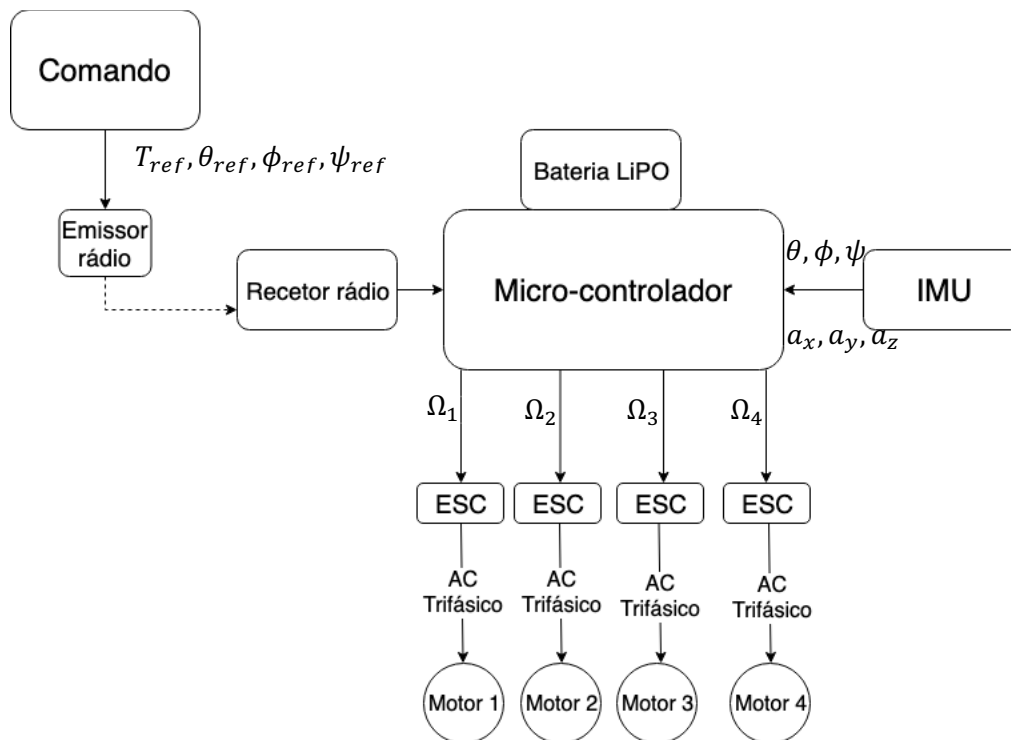


Figura 2. 15 - Diagrama da arquitetura de um *quadcopter*.

O microcontrolador é a unidade principal do *quadcopter*, porque recebe os sinais de comando do utilizador ($T_{ref}, \theta_{ref}, \phi_{ref}, \psi_{ref}$) e os valores dos sensores de movimento ($\theta, \phi, \psi, a_x, a_y, a_z$) para calcular as velocidades de rotação dos motores ($\Omega_1, \Omega_2, \Omega_3, \Omega_4$).

O comando comunica com o microcontrolador através de um sinal rádio e é neste que o utilizador comanda a orientação e posição do *quadcopter*. Os sinais produzidos pelo comando controlam a posição vertical do *drone*, T_{ref} , consoante o valor desta variável, todos os motores aceleram ou abrandam de forma a que o *quadcopter* possa ascender, descer ou ficar na mesma posição. O utilizador com o comando também define a orientação (*roll*, *pitch* e *yaw*) do *quadcopter* através das variáveis θ_{ref}, ϕ_{ref} e ψ_{ref} . Através do comando da orientação do *quadcopter* o utilizador consegue controlar a posição do *quadcopter* no espaço.

O IMU (*Inertial Measurement Unit*), Figura 2. 16, é o componente que contém os vários sensores responsáveis por medir a orientação e a posição do *quadcopter*. Através de um giroscópio e de um acelerómetro, o IMU mede os ângulos *roll* (ϕ), *pitch* (θ) e *yaw* (ψ) e a aceleração nos três eixos (a_x , a_y e a_z). IMU mais sofisticados podem ter outros sensores como altímetros, barómetros e magnetómetros para poder seguir com maior detalhe a posição e orientação da aeronave.

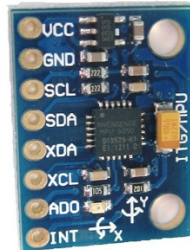


Figura 2. 16 - IMU com acelerómetro e giroscópio, MPU6050 [20].

Tipicamente neste tipo de aeronaves os motores utilizados são do tipo BLDC (*brushless*) porque não necessitam de manutenção (não utilizam escovas), são leves e conseguem alcançar velocidades de rotação elevadas. Como são motores trifásicos síncronos (semelhantes aos motores de ímanes permanentes), é necessário um inversor trifásico para transformar a corrente contínua em alternada, para fazer mover os motores. No *quadcopter* os ESC (*Electronic Speed Controllers*), Figura 2. 17, são responsáveis por regular a velocidade de rotação dos motores, convertendo a corrente contínua da bateria LiPO (*Lithium Polimer*) em corrente alternada trifásica.

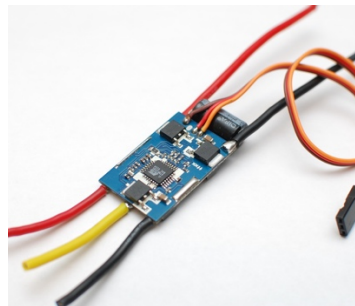


Figura 2. 17 - ESC de 35 A.

2.4 Sensores

Para poder realizar o controlo de voo é necessário monitorizar várias grandezas que descrevem a orientação e posição da aeronave. Normalmente recorre-se a sensores baseados em sistemas micro eletromecânicos (MEMS) por serem baratos, eficientes e rápidos [3].

2.4.1 Acelerómetro de três eixos

Um acelerómetro de três eixos tem como objetivo medir a aceleração nos eixos x , y e z . A Figura 2. 18 ilustra o funcionamento de um acelerómetro que mede a aceleração em dois eixos.

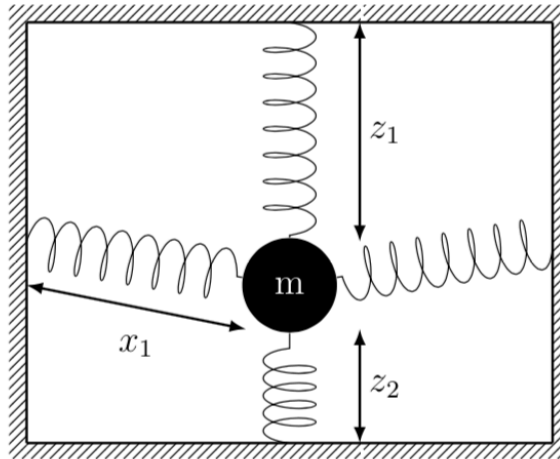


Figura 2. 18 - Acelerómetro de molas [3].

Observando a Figura 2. 18, o acelerómetro é composto por uma massa suspensa através de molas. A aceleração que atua sobre essa massa é dada pela deformação nas molas. Tanto as molas em x como em z sofrem deformação devido à aceleração da gravidade, mas apenas em z a deformação é diferente para cada uma das molas enquanto que em x as molas têm a mesma deformação. Em repouso, a força gravítica é a única força a atuar sobre a massa, assim a aceleração (no eixo do z) medida pelo o acelerómetro pode ser dada da seguinte forma [3]:

$$F = m.A; F = k.z;$$

$$a_z = \frac{k.(z_1 - z_2)}{2.m}, \quad (2.5)$$

em que m é a massa sujeita à aceleração A , k é o coeficiente da mola, z é a extensão da mola, a_z é a aceleração medida no eixo do z , e z_1 e z_2 são a extensão das molas no eixo z .

É possível determinar dois ângulos de Euler através da aceleração medida por um acelerómetro de três eixos. As expressões (2. 6) e (2. 7) convertem a aceleração num referencial cartesiano para os ângulos θ e ϕ [3]:

$$\theta = \text{atan}\left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}}\right) \frac{180}{\pi}; \quad (2.6)$$

$$\phi = \text{atan2}(a_y, a_z) \frac{180}{\pi}, \quad (2.7)$$

em que a_x , a_y e a_z são as acelerações medidas pelo acelerómetro em cada eixo.

No anexo B está descrita com mais detalhe a estimação da orientação através de um acelerómetro de três eixos.

Não é possível estimar o ângulo ψ através de um acelerómetro de três eixos porque um movimento em ψ não provoca nenhuma alteração na aceleração dos três eixos cartesianos [3].

Apenas se consegue estimar a orientação de um corpo com um acelerómetro quando a aceleração gravítica (g) é a única força a atuar sobre esse corpo. Quando uma aceleração externa atua sobre o acelerómetro, a orientação deste torna-se desconhecida. Para exemplificar esta situação, supõe-se que sobre o acelerómetro está a atuar uma aceleração duas vezes maior à da gravidade e no sentido oposto a esta, $a_z = -1g, a_x = 0g, a_y = 0g$. Substituindo estes valores de aceleração na expressão (2. 7), obtém-se $\phi = 180^\circ$, ou seja o *quadcopter* estaria invertido quando na realidade apenas tinha uma aceleração positiva no eixo do z [3].

Outro aspeto relevante dos acelerómetros é que estes são sensíveis a vibrações com altas frequências. Isto não seria um problema no caso de outras aplicações, mas como os *quadcopters* possuem motores que atingem altas velocidades de rotação, os acelerómetros podem ser afetados pelas vibrações causadas pelos motores. A aplicação de filtros passa-baixos poderia diminuir o ruído nas medições do acelerómetro, mas também provocaria um atraso que poderia causar instabilidade no voo [3].

2.4.2 Giroscópio de três eixos

Utilizando apenas um acelerómetro de três eixos para medir a orientação de um *quadcopter*, e como não é possível medir o ângulo ψ com esse sensor, o binário total da aeronave ficará desequilibrado devido à falta de controlo na rotação *yaw*. O desequilíbrio no binário total do sistema faz com que o *quadcopter* comece a rodar sobre si, de forma descontrolada. Devido a esta limitação do acelerómetro, é necessário recorrer a um giroscópio de três eixos que já possibilita a monitorização de todos os ângulos de Euler.

Ao contrário dos acelerómetros, os giroscópios medem rotações. Os primeiros giroscópios consistiam de massas giratórias no interior de uma estrutura *gimbal*, Figura 2. 19.



Figura 2. 19 - Giroscópio clássico [21].

O funcionamento do giroscópio baseia-se no princípio de conservação do momento angular. O disco no interior do giroscópio roda livremente e mantém a rotação do seu eixo independentemente da orientação da estrutura exterior. Devido à conservação do momento angular, o eixo de rotação do disco mantém-se com a mesma orientação e caso exista alguma força externa para reorientar o eixo de rotação, uma força resistiva irá opor-se [21]. Para medir eletricamente mudanças na rotação, potenciômetros são ligados à estrutura exterior do giroscópio [3].

Os giroscópios MEMS utilizados nos *drones* não têm um disco giratório devido à dificuldade em obter momentos angulares elevados em espaços microscópicos. Devido a esta limitação, estes giroscópios baseiam-se noutro fenómeno físico, o efeito *coriolis*. Em vez de medirem os ângulos de orientação como nos giroscópios clássicos, estes giroscópios medem a velocidade angular, logo para se obterem os ângulos de Euler com este tipo de sensor é necessário integrar as medições [3]:

$$\theta = \int_0^t \dot{\theta} dt; \quad \phi = \int_0^t \dot{\phi} dt; \quad \psi = \int_0^t \dot{\psi} dt. \quad (2. 8)$$

Como resultado da integração das leituras do giroscópio, o erro estacionário relativo à orientação acumula-se ao longo do tempo. Esta acumulação do erro provoca um desvio nos valores dos ângulos. Devido a isto, os giroscópios têm de ser recalibrados frequentemente ou então filtrados por um filtro passa-alto [3].

2.4.3 Fusão de sensores

Nos *quadcopters* é comum combinar a informação de vários sensores para estimar a orientação do *quadcopter*. Os IMU mais simples utilizam a informação proveniente de um giroscópio e de um acelerómetro para calcular os ângulos de Euler. Existem vários algoritmos de fusão de dados, mas consoante a complexidade, a capacidade computacional disponível e a velocidade dos controladores, nem todos os algoritmos são adequados para o voo de *quadcopters*. O filtro complementar, Figura 2. 20, permite obter a orientação agregando os dados do giroscópio e do acelerómetro. Devido ao acumular do erro por parte do giroscópio, no filtro complementar os valores

dos ângulos do giroscópio têm mais interesse a curto prazo, enquanto que o impacto dos valores dos ângulos do acelerómetro é maior a longo prazo. Os dados de cada sensor têm um peso diferente nos valores dos ângulos finais. Os valores destes pesos são regulados consoante o desempenho que se pretende ter no filtro [22].

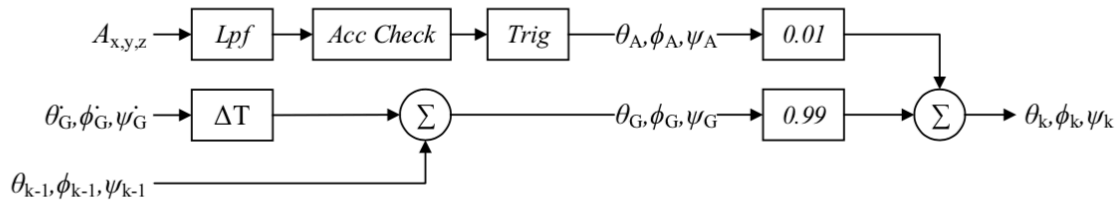


Figura 2. 20 - Filtro complementar que atua sobre os dados do acelerómetro e do giroscópio [3].

Na Figura 2. 20, $A_{x,y,z}$ é a aceleração dada pelo acelerómetro e $\dot{\theta}_G$, $\dot{\phi}_G$ e $\dot{\psi}_G$ são as velocidades de rotação medidas pelo giroscópio. Como se explicou em 2.4.1, devido ao ruído no acelerómetro, provocado pelas vibrações de alta frequência, no filtro complementar é aplicado um filtro passa baixo aos dados do acelerómetro. Depois de filtrados, os dados são analisados e convertidos em ângulos através das expressões (2. 6) e (2. 7). Os dados do giroscópio são integrados antes de serem combinados com os do acelerómetro [22].

No filtro complementar da Figura 2. 20, o ângulo ψ é obtido apenas com os dados do giroscópio porque com os dados do acelerómetro não é possível calcular este ângulo. Com isto, devido ao acumular do erro por parte do giroscópio, o valor medido deste ângulo ao longo do tempo irá desviar-se do valor real. Uma forma de diminuir este efeito seria adicionar um magnetómetro ao filtro complementar [3].

2.5 Motores

Os dois tipos de motores que tipicamente são utilizados nos *quadcopters* são os motores DC e os motores BLDC, Figura 2. 21.



Figura 2. 21 - Motor: (a) DC; (b) BLDC.

A primeira diferença entre estes dois motores é indicada no nome BLDC, *Brushless* DC, sem escovas. Apesar de ser dominado de *brushless* DC, o motor BLDC não é um motor de corrente contínua como o motor DC.

A configuração do motor DC está ilustrada na Figura 2. 22. O motor DC é constituído por uma armadura bobinada que se comporta como um eletroímã. O sentido da corrente na armadura é invertido duas vezes por ciclo através de um comutador mecânico. Assim os polos do eletroímã vão-se atraindo e repelindo em relação ao ímã permanente na parte exterior do motor [23].

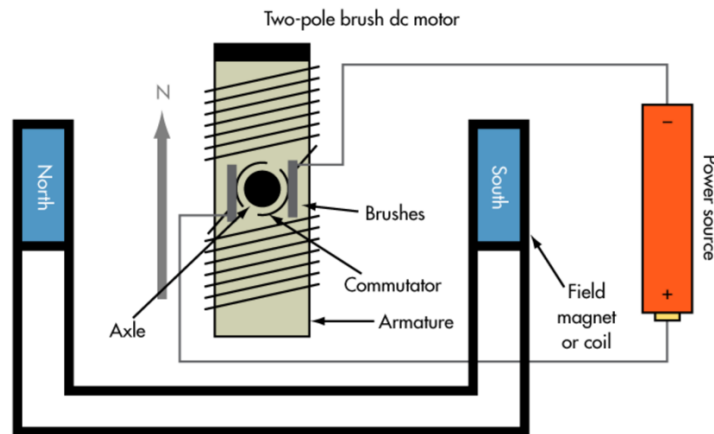


Figura 2. 22 - Diagrama de funcionamento do motor DC [24].

Como se pode visualizar pela Figura 2. 23, o motor BLDC é um motor trifásico em que o rotor é composto por ímãs permanentes. O comportamento e controlo do motor BLDC é semelhante ao do motor AC síncrono [24].

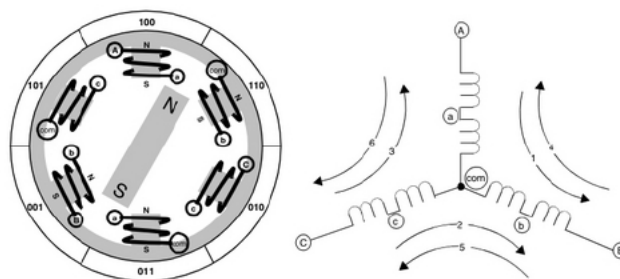


Figura 2. 23 - Diagrama da estrutura do motor BLDC [25].

Entre estes dois tipos de motores o mais utilizado em *drones* é o BLDC devido às vantagens que este tem sobre o motor DC. As principais vantagens são: maior binário por peso e eficiência, maior fiabilidade, menor ruído e maior tempo de vida [25].

Existem dois tipos de motores BLDC, um trapezoidal e outro sinusoidal. No tipo trapezoidal, a força eletromotriz induzida nos enrolamentos do estator tem forma trapezoidal e as fases têm de ser alimentadas por correntes com forma de onda quase quadrática. No tipo sinusoidal a força eletromotriz induzida é uma senoide, e as correntes nas fases também devem ser sinusoidais. A forma de onda da força eletromotriz induzida está relacionada com a forma dos ímãs do rotor e com a distribuição dos enrolamentos no estator. O motor sinusoidal necessita de um sensor de posição de alta resolução, porque a posição do rotor tem de ser conhecida em todos os instantes para funcionar corretamente. Como no motor trapezoidal é possível

realizar o controlo sem o sensor de posição, este motor é mais utilizado no *drones*, porque o custo associado e a complexidade é menor relativamente aos motores sinusoidais [26].

O funcionamento do motor BLDC trifásico é a duas fases, isto é, dependendo da posição do rotor, as duas fases que produzem o maior binário estão ligadas enquanto que a terceira está desligada. Na Figura 2. 24 estão ilustradas as formas de ondas ideais para as forças eletromotrizes e as correntes de cada fase. Também é possível observar que a cada 60° é alterado o valor do sensor de posição. A comutação das correntes em cada fase é feita através de uma ponte trifásica, Figura 2. 25. Os interruptores da ponte são representados por transístores bipolares de junção, mas tipicamente são utilizados MOSFET [26].

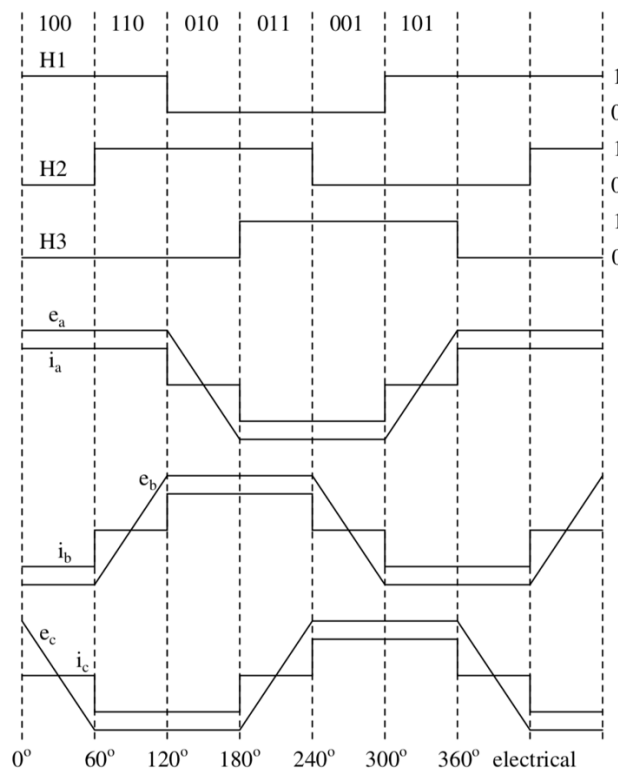


Figura 2. 24 - Sinais do sensor de posição, forças eletromotrizes e as correntes nas fases [26].

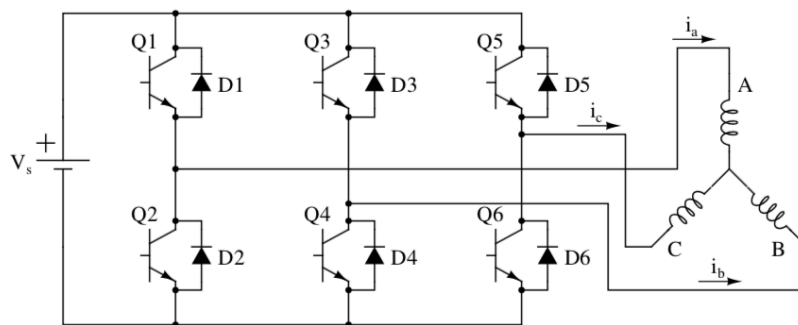


Figura 2. 25 - Inversor trifásico simplificado [26].

A Tabela 2. 1 apresenta a sequência de comutação dos interruptores, os sinais do sensor de posição e o sentido das correntes em cada fase.

Tabela 2. 1 - Sequência de comutação [26]

Intervalo de comutação	Número da sequência	Sensor de posição			Interruptores Fechados		Corrente nas fases		
		H1	H2	H3			A	B	C
0° - 60°	0	1	0	0	Q1	Q4	+	-	off
60° - 120°	1	1	1	0	Q1	Q6	+	off	-
120° - 180°	2	0	1	0	Q3	Q6	off	+	-
180° - 240°	3	0	1	1	Q3	Q2	-	+	off
240° - 300°	4	0	0	1	Q5	Q2	-	off	+
300° - 360°	5	1	0	1	Q5	Q4	off	-	+

O sensor de posição do rotor é constituído por três sensores de efeito de *Hall* que detetam a presença de pequenos ímanes colocados no veio do motor [26]. Na Figura 2. 26 está representada a secção transversal de um motor BLDC com um sensor de posição do rotor.

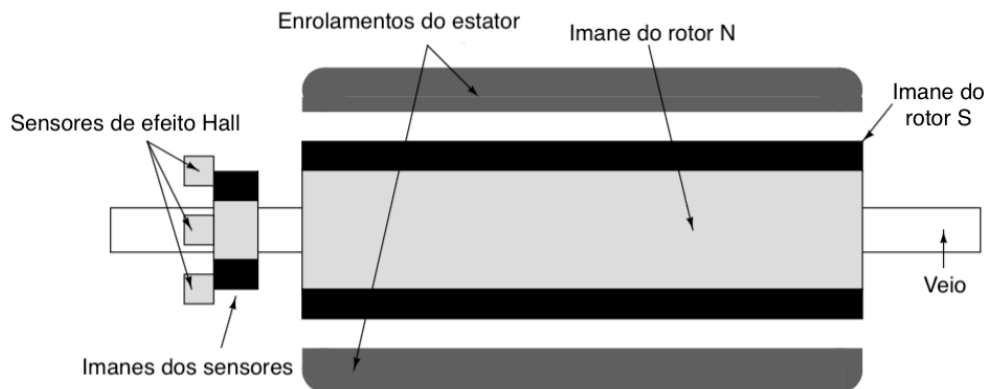


Figura 2. 26 - Secção transversal de um motor BLDC [26].

O sensor de posição do rotor permite realizar o arranque e fornecer a sequência correta de comutação do inversor trifásico. Estes sensores aumentam o custo e o tamanho do motor e também são necessários arranjos mecânicos para poderem ser montados. Os sensores de posição, particularmente os de efeito de *Hall*, são sensíveis à temperatura, limitando o funcionamento do motor a temperaturas abaixo dos 75 °C. Também existem aplicações em que devido às características dos motores utilizados, não é possível montar os sensores de posição do rotor. Devido a estas contrariedades associadas ao sensor de posicionamento, o controlo sem sensor dos motores BLDC tem sido o preferido em várias aplicações [27].

Existem duas técnicas de controlo sem sensores de posição do rotor, uma que recorre à força eletromotriz induzida para determinar o momento ótimo de comutação, e outra que através das tensões e correntes nas fases consegue estimar o ângulo do rotor. A segunda técnica como tem uma computação mais complexa, requer a utilização de DSP para realizar o controlo, aumentando assim o custo do sistema. Por esse motivo

normalmente opta-se pela técnica da detecção da força eletromotriz para efetuar o controlo do motor BLDC [27].

Como no motor trifásico a cada intervalo de 60° duas fases conduzem e outra está desligada, na fase desligada é induzida uma força eletromotriz. Ao final de cada intervalo de 60° , a corrente é comutada, e para se obter uma maior quantidade de binário por ampere, a corrente tem de estar em fase com a força eletromotriz induzida [27]. O instante de comutação é dado pela posição do rotor, e como a forma da força eletromotriz induzida revela a posição do rotor, também é possível determinar o instante de comutação através desta força. Através da detecção da passagem por zero da força eletromotriz induzida é possível determinar o instante ideal de comutação. A fase que está em aberto é a fase utilizada para detetar o cruzamento por zero da força eletromotriz. Na Figura 2. 27 estão representados dois esquemas de detecção do cruzamento por zero da força eletromotriz.

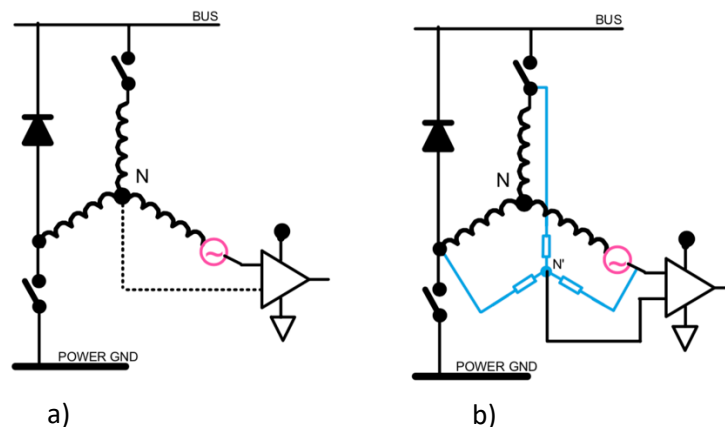


Figura 2. 27 – Detecção do cruzamento por zero com: a) ponto neutro do motor; b) ponto de neutro virtual [27].

Na maioria dos casos o ponto neutro do motor não está disponível, sendo na prática o método do ponto de neutro virtual o mais utilizado. O neutro virtual é criado através de resistências cujos valores têm de ser escolhidos de forma a não exercerem carga sobre o motor.

A baixas velocidades de rotação, a força eletromotriz induzida é baixa, o que diminui a precisão da comutação. Quanto maior for a incerteza no instante de comutação, menor será a eficiência energética do sistema. Por isso, normalmente no arranque do motor é utilizado um método em malha aberta até que seja atingida uma certa velocidade que permita ter forças eletromotrizes induzidas nas fases semelhantes às da Figura 2. 24. Alcançada essa velocidade, o controlo passa a ser feito através da detecção da passagem por zero da força eletromotriz induzida.

Em [28] está descrito em mais pormenor este método da deteção da passagem por zero, bem como outros métodos de controlo do motor BLDC como o da determinação do ângulo do rotor através das correntes e das tensões nas fases.

2.6 Telecomando

Tipicamente um *quadcopter* é controlado através de um comando com um rádio transmissor. Atualmente a maioria dos rádios transmissores utilizados nesta área operam na banda dos 2,4 GHz. Para maiores alcances, existem rádio transmissores a operarem com frequências mais baixas, 433 MHz e 900 MHz. Também existem *drones* comerciais que são controlados por *smartphone* através de Wi-Fi na banda dos 5 GHz.

O sistema de 2,4 GHz é o padrão de controlo rádio depois de novos protocolos terem sido criados, que introduziram a tecnologia de salto em frequência. Este sistema procura automaticamente por um canal disponível de forma a não interferir com outros pilotos, permitindo que haja várias aeronaves a voar simultaneamente [29].

Uma frequência maior (como 2,4 GHz) tem a vantagem de ter uma antena mais pequena em relação às frequências mais baixas, mas também um alcance menor.

A Figura 2. 28 apresenta um exemplo de um comando de um *quadcopter* com o rádio transmissor e o respetivo recetor.



Figura 2. 28 - Componentes do telecomando: a) comando com transmissor rádio; b) recetor rádio.

Normalmente os comandos de *quadcopters* têm dois manípulos que permitem comandar o acelerador e os movimentos de *roll*, *pitch* e *yaw*. O recetor rádio tem vários canais e cada um fica associado a uma ação do comando. No caso do *quadcopter* em que se tem um comando com dois manípulos a comandar quatro variáveis, o recetor necessita no mínimo de quatro canais. Na Figura 2. 29 é apresentada uma configuração comum dos comandos utilizados para controlar *drones*.



Figura 2. 29 - Configuração dos manípulos e respetivas variáveis controláveis.

2.7 Controlo

Num *quadcopter* o objetivo principal do controlo é conseguir controlar a orientação e a altura da aeronave:

$$\begin{aligned} \dot{\theta} &= 0; \quad \dot{\phi} = 0; \quad \dot{\psi} = 0; \\ \theta &\rightarrow \theta_{ref}; \quad \phi \rightarrow \phi_{ref}; \quad \psi \rightarrow \psi_{ref}; \quad T \rightarrow T_{ref}. \end{aligned} \quad (2. 9)$$

Na expressão (2. 9), θ_{ref} , ϕ_{ref} e ψ_{ref} são os valores de comando da orientação e T_{ref} o acelerador comandado pelo utilizador para fazer subir ou descer o *quadcopter*. $\dot{\theta}$, $\dot{\phi}$ e $\dot{\psi}$ são as velocidades de rotação do *quadcopter*.

Com os comandos θ_{ref} , ϕ_{ref} , ψ_{ref} e T_{ref} é possível manobrar o *quadcopter* em qualquer direção num espaço tridimensional. Nestes quatro graus de liberdade dados ao utilizador para operar a aeronave, a altitude em vários *drones* comerciais é controlada pelo próprio utilizador. Para controlar, o utilizador tem de observar a altura do *quadcopter* e tem de ajustar manualmente o acelerador (T_{ref}) para que a altura em relação ao solo seja aproximadamente constante. O utilizador também consegue controlar a posição no espaço, sendo que para tal tem de observar a posição em x e y e ajustar θ e ϕ de acordo com a posição que pretende alcançar.

Sem a observação do utilizador, para controlar a posição do *drone* no espaço, apenas com os quatro graus de liberdade da expressão (2. 9) não é possível ter um voo autónomo estável. Para tal é necessário um segundo objetivo no controlo que permita controlar a posição x , y e z do *quadcopter*:

$$\begin{aligned} \dot{x} &= 0; \quad \dot{y} = 0; \quad \dot{z} = 0; \\ x &\rightarrow x_{ref}; \quad y \rightarrow y_{ref}; \quad z \rightarrow z_{ref}. \end{aligned} \quad (2. 10)$$

em que x_{ref} , y_{ref} e z_{ref} são as coordenadas de referência. Apenas com um giroscópio e um acelerómetro não é possível efetuar este controlo e ter assim um sistema de voo

autónomo. Para tal são necessários sensores adicionais como GPS (*Global Positioning System*), LIDAR (*Light Detection And Ranging*) e sonar [3].

2.8 Legislação

Em Portugal a entidade responsável pela regulamentação dos voos de *drones* é a Autoridade Nacional da Aviação Civil, ANAC. A 14 de dezembro de 2016 saiu em Diário da República, o regulamento n.º 1093/2016 que aprova as condições de operação aplicáveis à utilização do espaço aéreo pelos sistemas de aeronaves civis pilotadas remotamente (*drones*) [30].

O regulamento da ANAC impõe regras gerais de operação tais como: aeronaves pilotadas remotamente apenas podem efetuar voos em condições de linha de vista até 120 m acima da superfície, à exceção de aeronaves brinquedo (aeronaves com peso inferior a 0,250 kg), que não devem exceder os 30 m de altura; o piloto remoto deve dar sempre prioridade de passagem às aeronaves tripuladas e afastar-se das mesmas sempre que, qualquer que seja o motivo, as aeronaves tripuladas estejam excepcionalmente a voar a uma altura próxima da aeronave pilotada remotamente; as aeronaves pilotadas remotamente estão proibidas de sobrevoar multidões de pessoas (mais do que 12 pessoas), salvo se expressamente autorizado pela ANAC; as aeronaves pilotadas remotamente têm de ter luzes de identificação ligadas independentemente se tratar de um voo diurno ou noturno. O regulamento também traça regras específicas para voos nas proximidades de infraestruturas aeroportuárias civis, como também nas proximidades nas áreas de jurisdição militar [31].

Nas proximidades de áreas restritas como aeroportos são definidas zonas que condicionam de diferentes formas o voo dos *drones*. Na Figura 2. 30 estão representadas as áreas de proteção operacional do aeroporto da Madeira, estabelecidas pela ANAC.

Aeroporto da Madeira

Áreas de Proteção Operacional associadas ao aeroporto

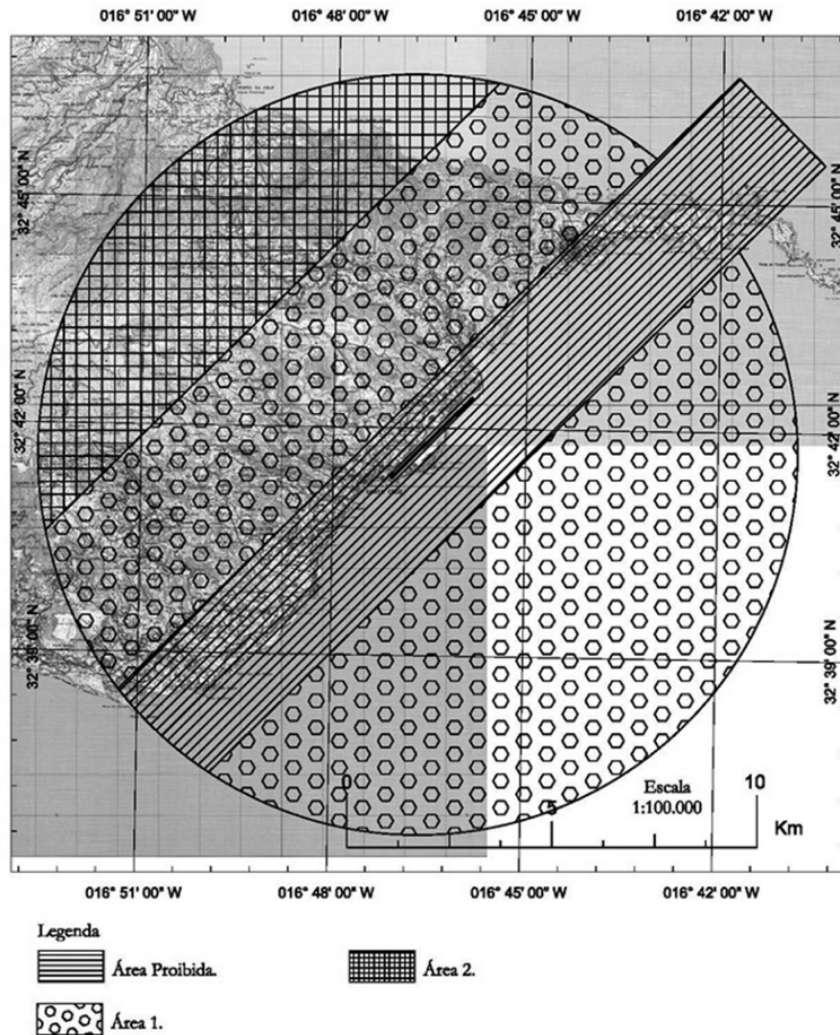


Figura 2. 30 - Definição das áreas de proteção operacional do aeroporto da Madeira [31].

Na área proibida são proibidos os voos das aeronaves pilotadas remotamente, exceto quando os voos são efetuados sob a responsabilidade do operador do aeroporto ou quando os voos são efetuados com base num protocolo acordado entre o prestador de serviços de tráfego aéreo e o operador responsável pelos voos. Na área 1 são permitidos voos de *drones* até 30 m acima da superfície, ou até à altura máxima do obstáculo (natural ou artificial) num raio de 75 m com centro na aeronave quando a altura desse obstáculo ultrapassa os 30 m. Na área 2 são permitidos voos de *drones* até uma altura máxima de 60 m, ou no caso da existência de um obstáculo com uma altura superior de 60 m, à semelhança da área 1, é permitido o voo até à altura desse obstáculo num raio de 75 m centrado na aeronave. A operação acima das alturas definidas para as áreas 1 e 2 carece de uma autorização por parte da ANAC [31].

3 Modelos e controlo do *drone*

Neste capítulo é descrito o modelo matemático do *quadcopter* como também técnicas de controlo da orientação e da posição. O modelo do *drone* descrito neste capítulo foca-se essencialmente nos *drones* com estrutura em forma de $+$.

A dinâmica do *quadcopter* neste capítulo tem por base as equações de Newton-Euler que descrevem a dinâmica rotacional e translacional de um corpo rígido. Para poder descrever a dinâmica do *quadcopter* assumiu-se que este tem uma estrutura simétrica e rígida, que o centro de gravidade do *quadcopter* coincide com a origem do referencial do corpo e que as hélices são rígidas.

3.1 Cinemática do *quadcopter*

Antes de começar a descrever o modelo matemático do *drone*, é necessário definir os referenciais das coordenadas. Na Figura 3. 1 estão representados os dois referenciais, o referencial fixo da Terra, com os eixos a apontar para Norte, Este e para baixo, o referencial do corpo com origem no centro de massa do *quadcopter*, com o eixo x na direção do motor 1, o eixo y na direção do motor 2 e o eixo z a apontar para baixo.

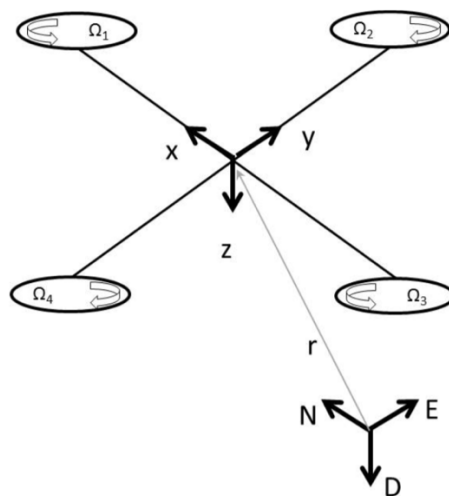


Figura 3. 1 - Referenciais do *quadcopter*: referencial do corpo e referencial fixo [32].

A distância entre o referencial fixo e o referencial do corpo descreve a posição do centro de massa do *drone* no espaço e é dada por:

$$r = [x \ y \ z]^T. \quad (3. 1)$$

A matriz de rotação R da expressão (2. 4) permite passar as grandezas do referencial do corpo para o referencial fixo. Essa matriz é importante porque a força gravítica é representada no referencial fixo enquanto que as forças geradas pelas hélices são representadas no referencial do corpo. A matriz inversa que permite passar do referencial fixo para o referencial do corpo é dada por [32]:

$$R_r = \begin{bmatrix} 1 & 0 & -\text{sen}(\theta) \\ 0 & \cos(\phi) & \text{sen}(\phi)\cos(\theta) \\ 0 & -\text{sen}(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix}. \quad (3.2)$$

A velocidade angular do *quadcopter* medida pelo IMU é relativa ao referencial fixo, logo para se ter as velocidades angulares no referencial do corpo recorre-se à matriz de rotação inversa:

$$\omega = R_r \dot{\eta} \quad (3.3)$$

em que $\omega = [p \ q \ r]^T$ são as velocidades angulares sobre os eixos x , y e z do referencial do corpo, e $\dot{\eta} = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$ as taxas de variação dos ângulos de Euler que são medidas no referencial fixo.

O movimento do *quadcopter* pode ser subdividido em dois subsistemas, subsistema rotacional (orientação em ϕ , θ e ψ) e subsistema translacional (altura e posição, em z e em x e y).

3.1.1 Equações do movimento rotacional

As equações que descrevem a rotação do *quadcopter* têm origem no referencial do corpo com a fórmula geral de Newton-Euler [32]:

$$J\dot{\omega} + \omega \times J\omega + M_G = M_B \quad (3.4)$$

em que J é a matriz de inércia do *quadcopter*, ω a velocidade angular do corpo, M_G o momento giroscópico devido à inércia dos motores e M_B os momentos a atuarem sobre o *quadcopter* no referencial do corpo.

Os primeiros dois termos da expressão (3.4), $J\dot{\omega}$ e $\omega \times J\omega$ representam a mudança do momento angular no referencial do corpo [33]. O momento giroscópico M_G , devido à inércia dos motores, é definido da seguinte maneira [32]:

$$M_G = \omega \times [0 \ 0 \ J_r \Omega_r]^T \quad (3.5)$$

em que J_r é a inércia dos motores e Ω_r a velocidade relativa dos motores:

$$\Omega_r = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4. \quad (3.6)$$

O momento giroscópico de um motor é um efeito físico em que o binário giroscópico tenta alinhar o eixo de rotação do motor com o eixo vertical do referencial fixo [32].

A matriz de inércia do *quadcopter* no referencial do corpo é uma matriz diagonal devido ao facto da estrutura do *drone* ser simétrica. A matriz de inércia é expressa por:

$$J = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (3.7)$$

onde I_{xx} , I_{yy} e I_{zz} são os momentos de inércia em cada um dos eixos do referencial do corpo.

O último termo da expressão (3.4), M_B , apresenta os momentos a atuarem sobre o *quadcopter*, sendo necessário definir dois fenômenos aerodinâmicos produzidos pela rotação dos motores: a força de elevação e o momento aerodinâmico. As expressões seguintes descrevem a força de elevação e o momento produzido por um motor de um *quadcopter* [32]:

$$F_i = \frac{1}{2} \rho A C_T r^2 \Omega_i^2; \quad (3.8)$$

$$M_i = \frac{1}{2} \rho A C_D r^2 \Omega_i^2. \quad (3.9)$$

Nas expressões (3.8) e (3.9), ρ é a densidade do ar, A é a área das hélices, r é o raio das hélices, C_T e C_D são os coeficientes aerodinâmicos das hélices e Ω_i é a velocidade angular de um motor. As forças e momentos aerodinâmicos estão dependentes da geometria das hélices e da densidade do ar. Como a altura máxima que o *drone* pode alcançar é limitada, normalmente a densidade do ar é considerada constante. Com isto, as expressões (3.8) e (3.9) são simplificadas da seguinte forma:

$$F_i = K_f \Omega_i^2; \quad (3.10)$$

$$M_i = K_M \Omega_i^2. \quad (3.11)$$

As constantes aerodinâmicas K_f e K_M podem ser determinadas experimentalmente, ou através das dimensões geométricas das hélices.

Na Figura 3.2 estão representadas todas as forças e momentos aerodinâmicos num *quadcopter* gerados pelos motores. Cada motor produz uma força de elevação F_i e um momento M_i com direção oposta ao sentido de rotação do motor correspondente.

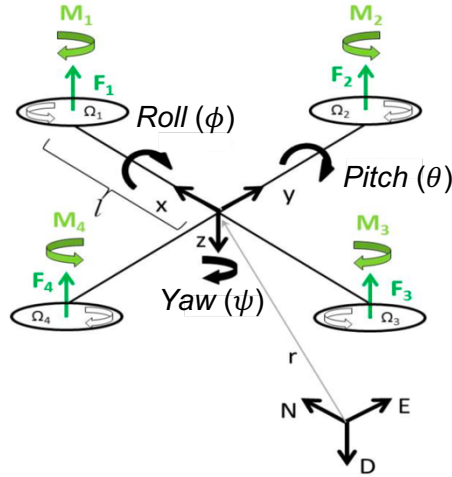


Figura 3. 2 - Forças e momentos aerodinâmicos no *quadcopter* [32].

No eixo do x do *quadcopter*, os momentos são provocados pelas forças de elevação dos motores 2 e 4. Considerando l a distância entre o centro de massa e um motor, utilizando a regra da mão direita, o momento total no eixo do x é dado por:

$$M_x = -F_2 l + F_4 l = l K_f (-\Omega_2^2 + \Omega_4^2). \quad (3. 12)$$

Utilizando o mesmo procedimento para o eixo do y , o total dos momentos nesse eixo é produzido pelos motores 1 e 3:

$$M_y = F_1 l - F_3 l = l K_f (\Omega_1^2 - \Omega_3^2). \quad (3. 13)$$

No caso do eixo do z , o momento não é causado pelas forças de elevação dos motores, mas sim pelos momentos devido à rotação dos motores, expressão (3. 11). Assim o momento no eixo do z é expresso por [32]:

$$M_z = M_1 - M_2 + M_3 - M_4 = K_M (\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2). \quad (3. 14)$$

Combinando as expressões (3. 12), (3. 13) e (3. 14) obtém-se o momento total a atuar sobre o *quadcopter*:

$$M_B = \begin{bmatrix} l K_f (-\Omega_2^2 + \Omega_4^2) \\ l K_f (\Omega_1^2 - \Omega_3^2) \\ K_M (\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{bmatrix}. \quad (3. 15)$$

As expressões (3. 12), (3. 13) e (3. 15) são válidas quando a estrutura do *quadcopter* tem forma de \dagger . Em forma de \times , Figura 3. 3, em que o eixo do x está entre o motor 1 e 2 e o eixo do y entre o motor 2 e 3, todos os motores têm influência sobre os momentos M_x e M_y .

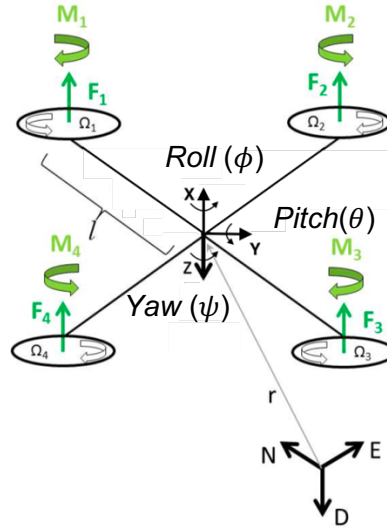


Figura 3. 3 - Posição dos eixos do referencial do corpo na estrutura em forma de x [32].

Na estrutura em forma de **x** os momentos M_x e M_y ficam [32]:

$$\begin{aligned} M_{x_x} &= F_1 l \text{sen}(45^\circ) - F_2 l \text{sen}(45^\circ) - F_3 l \text{sen}(45^\circ) + F_4 l \text{sen}(45^\circ) & (3. 16) \\ &= l \text{sen}(45^\circ) K_f (\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2); \end{aligned}$$

$$\begin{aligned} M_{y_x} &= F_1 l \text{sen}(45^\circ) + F_2 l \text{sen}(45^\circ) - F_3 l \text{sen}(45^\circ) - F_4 l \text{sen}(45^\circ) & (3. 17) \\ &= l \text{sen}(45^\circ) K_f (\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2). \end{aligned}$$

O $\text{sen}(45^\circ)$ das duas expressões anteriores deve-se ao ângulo de 45° entre os eixos do referencial e os braços do *quadcopter*. Com isto o total dos momentos na estrutura em **x** é dado por:

$$M_{B_x} = \begin{bmatrix} l \text{sen}(45^\circ) K_f (\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ l \text{sen}(45^\circ) K_f (\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2) \\ K_M (\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{bmatrix}. \quad (3. 18)$$

3.1.2 Equações do movimento translacional

As equações que descrevem o movimento de translação do *quadcopter* baseiam-se na segunda lei de Newton e têm origem no referencial fixo da Terra [32]:

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + RF_B. \quad (3. 19)$$

Da expressão (3. 19), $r = [x \ y \ z]^T$ é a distância entre a origem do referencial fixo e o centro de massa do *quadcopter*, m é a massa do *quadcopter*, g é a aceleração gravítica ($9,81 \text{ m/s}^2$) e F_B são as forças não gravitacionais a atuarem sobre o *quadcopter*.

Na situação em que o *quadcopter* se encontra numa orientação horizontal com os ângulos ϕ e θ a zero, as forças de elevação dos motores são as únicas forças não gravitacionais a agirem sobre a aeronave. Neste contexto, a partir da expressão (3. 10) as forças não gravitacionais são expressas da seguinte forma:

$$F_B = \begin{bmatrix} 0 \\ 0 \\ -K_f(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \end{bmatrix}. \quad (3. 20)$$

Na expressão anterior as primeiras duas linhas são zero porque o *quadcopter* está numa posição horizontal, logo nessas direções as forças de elevação dos motores não têm efeito. Na terceira linha, a força de elevação é igual ao somatório de cada uma das forças produzidas por cada motor. Esta força tem sinal negativo porque, como se definiu na Figura 3. 1, o eixo z está a apontar para baixo no sentido oposto a esta força.

Na expressão (3. 19) F_B é multiplicado pela matriz de rotação R para passar as forças de elevação do referencial do corpo para o referencial fixo. Desta forma a expressão (3. 19) pode ser aplicada qualquer que seja a orientação do *quadcopter* [32].

3.1.3 Efeitos aerodinâmicos

Até agora na descrição da dinâmica do *quadcopter*, os efeitos aerodinâmicos a atuarem sobre o corpo do *quadcopter* foram desprezados. No entanto, para se ter um modelo mais próximo da realidade é necessário ter em conta estes efeitos, nomeadamente as forças e os momentos produzidos pela fricção do ar [32].

A fricção do ar no *quadcopter* devido ao seu movimento produz uma força de oposição. À medida que a velocidade aumenta, esta força de oposição ao movimento também aumenta. As forças devido à fricção do ar podem ser aproximadas pela seguinte expressão [32]:

$$F_a = K_t \dot{r}. \quad (3. 21)$$

onde \dot{r} é a derivada do vetor de posição e K_t é uma matriz constante denominada de matriz dos coeficientes de translação aerodinâmica [32]. Com esta força resistiva adicional, a expressão (3. 19) do movimento de translação fica:

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + RF_B - F_a. \quad (3. 22)$$

À semelhança da força resistiva devido à fricção do ar, esse efeito aerodinâmico também causa um momento sobre o *quadcopter*, que pode ser expresso por [32]:

$$M_a = K_r \dot{\eta}. \quad (3. 23)$$

Na expressão anterior K_r é uma matriz constante, designada de matriz dos coeficientes de rotação aerodinâmica, e $\dot{\eta}$ é a derivada dos ângulos de Euler [32]. Com este efeito aerodinâmico adicional, a expressão (3. 4) do movimento rotacional fica:

$$J\dot{\omega} + \omega \times J\omega + \omega \times [0 \ 0 \ J_r\Omega_r]^T = M_B - M_a. \quad (3. 24)$$

3.2 Modelo em variáveis de estado

Para simplificar o controlo de orientação e posição, transforma-se o modelo matemático do *quadcopter* em variáveis de estado.

O vetor das variáveis de estado do *quadcopter* é definido da seguinte forma:

$$X = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9 \ x_{10} \ x_{11} \ x_{12}]^T. \quad (3. 25)$$

O vetor das variáveis de estado define a orientação, posição e velocidades angulares e lineares do *quadcopter*:

$$X = [\phi \ \dot{\phi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi} \ z \ \dot{z} \ x \ \dot{x} \ y \ \dot{y}]^T. \quad (3. 26)$$

3.2.1 Vetor das entradas de controlo

O vetor das entradas de controlo consiste em quatro variáveis, responsáveis por controlar o movimento de subida e descida, o movimento de *roll*, o movimento de *pitch* e o movimento de *yaw*. O vetor de entrada é definido como:

$$U = [U_1 \ U_2 \ U_3 \ U_4], \quad (3. 27)$$

em que:

$$U_1 = K_f(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2); \quad (3. 28)$$

$$U_2 = K_f(-\Omega_2^2 + \Omega_4^2); \quad (3. 29)$$

$$U_3 = K_f(\Omega_1^2 - \Omega_3^2); \quad (3. 30)$$

$$U_4 = K_M(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2). \quad (3. 31)$$

Na forma matricial, as variáveis de entrada ficam:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} K_f & K_f & K_f & K_f \\ 0 & -K_f & 0 & K_f \\ K_f & 0 & -K_f & 0 \\ K_M & -K_M & K_M & -K_M \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \quad (3. 32)$$

O vetor de entrada U das expressões (3. 28) a (3. 32) é adequado para o controlo de *quadcopters* com uma estrutura em $+$, Figura 3. 2, em que o eixo do x está alinhado com o motor 1 e o eixo do y está alinhado com o motor 2. No caso de uma estrutura

em x , como na Figura 3. 3, o eixo do x passa a estar entre o motor 1 e 2, e o eixo do y entre o motor 2 e o motor 3.

Na forma em x as entradas de controlo ficam diferentes porque nesse caso todos os motores têm influência nos momentos M_{x_x} e M_{y_x} do *quadcopter*. Assim para a estrutura em x , os elementos U_2 e U_3 do vetor de entrada U ficam:

$$U_{2_x} = K_f(\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2); \quad (3. 33)$$

$$U_{3_x} = K_f(\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2). \quad (3. 34)$$

Na forma matricial o vetor de entrada de controlo para a estrutura em x é dado da seguinte forma:

$$\begin{bmatrix} U_{1_x} \\ U_{2_x} \\ U_{3_x} \\ U_{4_x} \end{bmatrix} = \begin{bmatrix} K_f & K_f & K_f & K_f \\ K_f & -K_f & -K_f & K_f \\ K_f & K_f & -K_f & -K_f \\ K_M & -K_M & K_M & -K_M \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix}. \quad (3. 35)$$

Do vetor de controlo U , U_1 representa as forças produzidas por todos os motores responsáveis por controlar a altitude, ou seja, as variáveis de estado z e \dot{z} . A componente U_2 , dependendo do tipo de estrutura, engloba as forças dos motores que produzem o momento M_x , ou seja, U_2 é responsável pelo controlo das variáveis ϕ e $\dot{\phi}$. U_3 , como U_2 , depende do tipo de estrutura e engloba as forças dos motores que provocam o momento M_y . Assim U_3 controla as variáveis θ e $\dot{\theta}$. A componente U_4 contem os binários de todos os motores que originam o momento M_z , e desta forma U_4 controla as variáveis ψ e $\dot{\psi}$.

A forma como é definido o vetor U permite desacoplar o sistema rotacional porque assim U_1 controla a altitude, U_2 controla o ângulo de *roll*, U_3 controla o ângulo de *pitch* e U_4 controla o ângulo de *yaw* [32].

3.2.2 Equação do movimento rotacional

Utilizando a equação (3. 15) com as componentes do vetor de entrada U_2 , U_3 e U_4 , o total dos momentos a atuarem sobre o *quadcopter* fica:

$$M_B = \begin{bmatrix} lU_2 \\ lU_3 \\ U_4 \end{bmatrix}. \quad (3. 36)$$

Através da equação do movimento rotacional, (3. 4), com a expressão (3. 36) e expandido cada termo com a sua definição previamente explicada ao longo deste capítulo, obtém-se o seguinte [32]:

$$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ J_r \Omega_r \end{bmatrix} =$$

$$\begin{bmatrix} I_{xx} \ddot{\phi} \\ I_{yy} \ddot{\theta} \\ I_{zz} \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\theta} I_{zz} \dot{\psi} - \dot{\psi} I_{yy} \dot{\theta} \\ \dot{\psi} I_{xx} \dot{\phi} - \dot{\phi} I_{zz} \dot{\psi} \\ \dot{\phi} I_{yy} \dot{\theta} - \dot{\theta} I_{xx} \dot{\phi} \end{bmatrix} + \begin{bmatrix} \dot{\theta} J_r \Omega_r \\ -\dot{\phi} J_r \Omega_r \\ 0 \end{bmatrix} = \begin{bmatrix} l U_2 \\ l U_3 \\ U_4 \end{bmatrix}. \quad (3.37)$$

Reescrevendo a expressão anterior em função das acelerações angulares, fica-se com:

$$\ddot{\phi} = \frac{l}{I_{xx}} U_2 - \frac{J_r}{I_{xx}} \dot{\theta} \Omega_r + \frac{I_{yy}}{I_{xx}} \dot{\psi} \dot{\theta} - \frac{I_{zz}}{I_{xx}} \dot{\theta} \dot{\psi}; \quad (3.38)$$

$$\ddot{\theta} = \frac{l}{I_{yy}} U_3 - \frac{J_r}{I_{yy}} \dot{\phi} \Omega_r + \frac{I_{zz}}{I_{yy}} \dot{\phi} \dot{\psi} - \frac{I_{xx}}{I_{yy}} \dot{\psi} \dot{\phi}; \quad (3.39)$$

$$\ddot{\psi} = \frac{1}{I_{zz}} U_4 + \frac{I_{xx}}{I_{zz}} \dot{\theta} \dot{\phi} - \frac{I_{yy}}{I_{zz}} \dot{\phi} \dot{\theta}. \quad (3.40)$$

De forma a simplificar, são feitas as seguintes mudanças de variáveis:

$$\begin{aligned} a_1 &= \frac{I_{yy} - I_{zz}}{I_{xx}}; & a_5 &= \frac{I_{xx} - I_{yy}}{I_{zz}}; \\ a_2 &= \frac{J_r}{I_{xx}}; & b_1 &= \frac{l}{I_{xx}}; \\ a_3 &= \frac{I_{zz} - I_{xx}}{I_{yy}}; & b_2 &= \frac{l}{I_{yy}}; \\ a_4 &= \frac{J_r}{I_{yy}}; & b_3 &= \frac{l}{I_{zz}}. \end{aligned}$$

Com as mudanças de variáveis feitas, as expressões (3.38) a (3.40) podem ser reescritas de forma simplificada em função das variáveis de estado [32]:

$$\dot{x}_2 = b_1 U_2 - a_2 x_4 \Omega_r + a_1 x_4 x_6; \quad (3.41)$$

$$\dot{x}_4 = b_2 U_3 - a_4 x_2 \Omega_r + a_3 x_2 x_6; \quad (3.42)$$

$$\dot{x}_6 = b_3 U_4 + a_5 x_2 x_4. \quad (3.43)$$

3.2.3 Equação do movimento translacional

Substituindo vetor das entradas de controlo U , (3.28) a (3.31), na expressão das forças não gravitacionais a atuarem sobre o *quadcopter* (3.20), obtém-se:

$$F_B = \begin{bmatrix} 0 \\ 0 \\ -U_1 \end{bmatrix}. \quad (3.44)$$

Expandindo os termos da expressão (3.19) e substituindo o termo das forças não gravitacionais pelo da expressão (3.44), a equação do movimento translacional fica:

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + \begin{bmatrix} c\psi c\theta & c\psi s\phi s\theta & s\phi s\psi + c\phi c\psi s\theta \\ c\theta s\psi & c\theta c\psi + s\phi s\psi s\theta & c\phi s\psi s\theta - c\psi s\theta \\ -s\theta & c\theta s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -U_1 \end{bmatrix};$$

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + \begin{bmatrix} (s\phi s\psi + c\phi c\psi s\theta)(-U_1) \\ (c\phi s\psi s\theta - c\psi s\theta)(-U_1) \\ (c\phi c\theta)(-U_1) \end{bmatrix}. \quad (3.45)$$

Reescrevendo a expressão anterior de forma a isolar as acelerações lineares:

$$\ddot{x} = \frac{-U_1}{m} (\sin(\phi) \sin(\psi) + \cos(\phi) \cos(\psi) \sin(\theta)); \quad (3.46)$$

$$\ddot{y} = \frac{-U_1}{m} (\cos(\phi) \sin(\psi) \sin(\theta) - \cos(\psi) \sin(\phi)); \quad (3.47)$$

$$\ddot{z} = g - \frac{U_1}{m} (\cos(\phi) \cos(\theta)). \quad (3.48)$$

Utilizando o vetor X das variáveis de estado, as acelerações lineares ficam:

$$\ddot{x} = \frac{-U_1}{m} (\sin(x_1) \sin(x_5) + \cos(x_1) \cos(x_5) \sin(x_3)); \quad (3.49)$$

$$\ddot{y} = \frac{-U_1}{m} (\cos(x_1) \sin(x_5) \sin(x_3) - \cos(x_5) \sin(x_1)); \quad (3.50)$$

$$\ddot{z} = g - \frac{U_1}{m} (\cos(x_1) \cos(x_3)). \quad (3.51)$$

3.2.4 Representação em variáveis de estado

Com o vetor X das variáveis de estado (expressões (3.25) e (3.26)) e as expressões das acelerações angulares e lineares (expressões (3.41) a (3.43) e (3.49) a (3.51) respetivamente), o modelo matemático do *quadcopter* pode ser escrito em função das variáveis de estado [32]:

$$\dot{x}_1 = \dot{\phi} = x_2;$$

$$\dot{x}_2 = \ddot{\phi} = x_4 x_6 a_1 - x_4 \Omega_r a_2 + b_1 U_2;$$

$$\dot{x}_3 = \dot{\theta} = x_4;$$

$$\dot{x}_4 = \ddot{\theta} = x_2 x_6 a_3 + x_2 \Omega_r a_4 + b_2 U_3;$$

$$\begin{aligned}
\dot{x}_5 &= \dot{\psi} = x_6; \\
\dot{x}_6 &= \ddot{\psi} = x_2 x_4 a_5 + b_3 U_4; \\
\dot{x}_7 &= \dot{z} = x_8; \\
\dot{x}_8 &= \ddot{z} = g - \frac{U_1}{m} (\cos(x_1) \cos(x_3)); \\
\dot{x}_9 &= \dot{x} = x_{10}; \\
\dot{x}_{10} &= \ddot{x} = \frac{-U_1}{m} (\sin(x_1) \sin(x_5) + \cos(x_1) \sin(x_3) \cos(x_5)); \\
\dot{x}_{11} &= \dot{y} = x_{12}; \\
\dot{x}_{12} &= \ddot{y} = \frac{U_1}{m} (\sin(x_1) \cos(x_5) - \cos(x_1) \sin(x_3) \sin(x_5)).
\end{aligned}$$

$$f(X, U) = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} = \begin{bmatrix} x_2 \\ x_4 x_6 a_1 - x_4 \Omega_r a_2 + b_1 U_2 \\ x_4 \\ x_2 x_6 a_3 + x_2 \Omega_r a_4 + b_2 U_3 \\ x_6 \\ x_2 x_4 a_5 + b_3 U_4 \\ x_8 \\ g - \frac{U_1}{m} (\cos(x_1) \cos(x_3)) \\ x_{10} \\ \frac{-U_1}{m} (\sin(x_1) \sin(x_5) + \cos(x_1) \sin(x_3) \cos(x_5)) \\ x_{12} \\ \frac{U_1}{m} (\sin(x_1) \cos(x_5) - \cos(x_1) \sin(x_3) \sin(x_5)) \end{bmatrix} \quad (3.52)$$

3.3 Técnicas de controlo

Nesta secção serão descritas as técnicas para o controlo da altura e orientação do *quadcopter*. As técnicas utilizadas foram: PID, Fuzzy-PID e *backstepping*.

Na Figura 3.4 estão representados os controladores em conjunto com o sistema do *quadcopter*. O controlador de posição pode ser o próprio utilizador, que ao analisar o espaço e a posição da aeronave, define a orientação do *drone* através dos comandos ϕ_{ref} , θ_{ref} e ψ_{ref} para atingir a posição desejada.

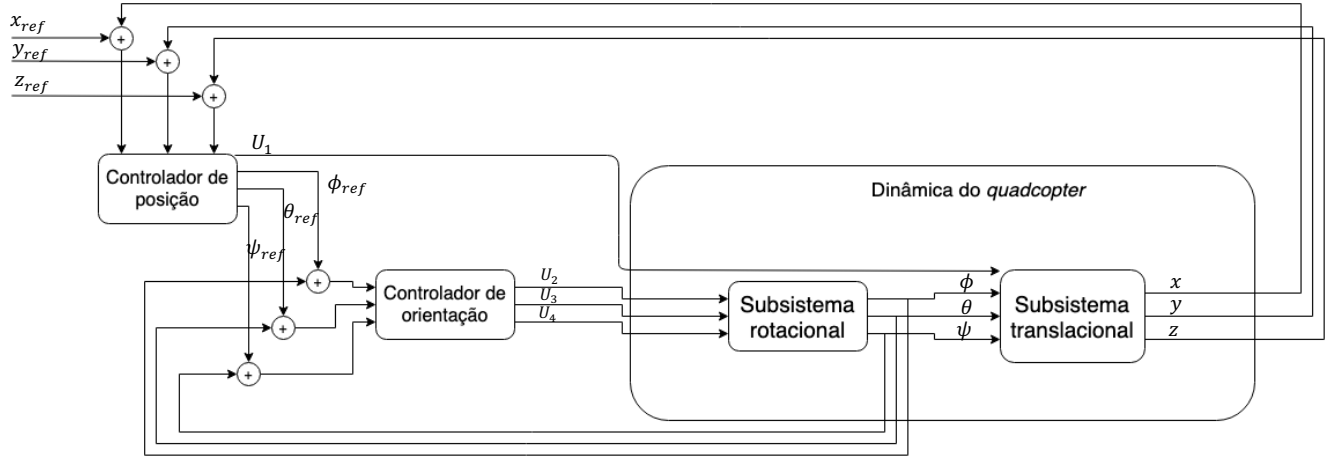


Figura 3. 4 - Diagrama do sistema completo com os controladores.

Ao contrário da orientação, que é controlada através de U_2 , U_3 e U_4 , o controle da posição do *quadcopter* é feito através dos ângulos ϕ e θ . Através das equações do movimento translacional (3. 46) e (3. 47) é possível determinar os ângulos de referência ϕ_{ref} e θ_{ref} :

$$\ddot{x} = -\frac{U_1}{m} (\sin(\phi_{ref}) \sin(\psi) + \cos(\phi_{ref}) \sin(\theta_{ref}) \cos(\psi)); \quad (3. 53)$$

$$\ddot{y} = -\frac{U_1}{m} (\cos(\phi_{ref}) \sin(\theta_{ref}) \sin(\psi) - \sin(\phi_{ref}) \cos(\psi)). \quad (3. 54)$$

Como o *quadcopter* opera com pouca inclinação, isto é, valores de ϕ e θ baixos, é possível simplificar as expressões (3. 53) e (3. 54) assumindo que para pequenos ângulos:

$$\cos(\phi_{ref}) \cong 1, \cos(\theta_{ref}) \cong 1; \quad (3. 55)$$

$$\sin(\phi_{ref}) \cong \phi_{ref}, \sin(\theta_{ref}) \cong \theta_{ref}. \quad (3. 56)$$

Com as simplificações anteriores, as equações (3. 53) e (3. 54) ficam:

$$\ddot{x} = -\frac{U_1}{m} (\phi_{ref} \sin(\psi) + \theta_{ref} \cos(\psi)); \quad (3. 57)$$

$$\ddot{y} = -\frac{U_1}{m} (\theta_{ref} \sin(\psi) - \phi_{ref} \cos(\psi)). \quad (3. 58)$$

Na forma matricial, as expressões anteriores ficam:

$$\begin{bmatrix} -\sin(\psi) & -\cos(\psi) \\ \cos(\psi) & -\sin(\psi) \end{bmatrix} \begin{bmatrix} \phi_{ref} \\ \theta_{ref} \end{bmatrix} = \frac{m}{U_1} \begin{bmatrix} \ddot{x}_{ref} \\ \ddot{y}_{ref} \end{bmatrix}. \quad (3. 59)$$

Invertendo a equação (3. 59) obtêm-se os ângulos de referência em função das posições de referência:

$$\begin{aligned}
\begin{bmatrix} \phi_{ref} \\ \theta_{ref} \end{bmatrix} &= \begin{bmatrix} -\sin(\psi) & -\cos(\psi) \\ \cos(\psi) & -\sin(\psi) \end{bmatrix}^{-1} \frac{m}{U_1} \begin{bmatrix} x_{ref}'' \\ y_{ref}'' \end{bmatrix}; \\
&= \frac{m}{U_1} \begin{bmatrix} -\sin(\psi) & \cos(\psi) \\ -\cos(\psi) & -\sin(\psi) \end{bmatrix} \begin{bmatrix} x_{ref}'' \\ y_{ref}'' \end{bmatrix}; \\
&= \frac{m}{U_1} \begin{bmatrix} -x_{ref}'' \sin(\psi) + y_{ref}'' \cos(\psi) \\ -x_{ref}'' \cos(\psi) - y_{ref}'' \sin(\psi) \end{bmatrix}.
\end{aligned} \tag{3.60}$$

Para que a expressão anterior seja válida, devido às aproximações feitas, os ângulos ϕ_{ref} e θ_{ref} têm de estar limitados entre -20° e 20° .

3.3.1 Controlador PID

O controlador PID (Proporcional, Integral e Derivativo) é uma técnica de controlo em malha fechada que ajusta a saída através da observação do erro entre a grandeza desejada e a grandeza medida. O controlador PID é caracterizado pelos coeficientes K_p , K_i e K_d , sendo a saída do controlador $u(t)$ em função do erro $e(t)$ dada por [34]:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt}. \tag{3.61}$$

Aplicando a transformada de Laplace à expressão (3.61), a função transferência do controlador PID é a seguinte [34]:

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s. \tag{3.62}$$

A Tabela 3.1 descreve os efeitos de cada ganho do controlador PID na resposta do sistema a uma entrada em degrau.

Tabela 3.1 - Efeito dos ganhos do controlador PID na resposta a um degrau [34].

Ganho PID	Percentagem de <i>overshoot</i>	Tempo de estabilização	Erro em estado estacionário
aumento de K_p	aumenta	pouco impacto	diminui
aumento de K_i	aumenta	aumenta	nulo
aumento de K_d	diminui	diminui	sem impacto

O compensador PID para controlar a altitude do *quadcopter* gera a entrada U_1 . A expressão para o controlador PID da altitude é a seguinte:

$$U_1 = K_{pz}(z_{ref} - z) + K_{iz} \int (z_{ref} - z) dt + k_{dz}(z_{ref} - \dot{z}). \tag{3.63}$$

As expressões (3.64) a (3.66) são as expressões dos compensadores responsáveis por controlarem a orientação do *quadcopter*. Através dos ângulos de referência

$(\phi_{ref}, \theta_{ref}, \psi_{ref})$ e dos ângulos medidos (ϕ, θ, ψ) , são produzidas as respectivas entradas de controlo (U_2, U_3, U_4) .

$$U_2 = K_{p_\phi}(\phi_{ref} - \phi) + K_{i_\phi} \int (\phi_{ref} - \phi)dt + k_{d_\phi}(\dot{\phi}_{ref} - \dot{\phi}); \quad (3.64)$$

$$U_3 = K_{p_\theta}(\theta_{ref} - \theta) + K_{i_\theta} \int (\theta_{ref} - \theta)dt + k_{d_\theta}(\dot{\theta}_{ref} - \dot{\theta}); \quad (3.65)$$

$$U_4 = K_{p_\psi}(\psi_{ref} - \psi) + K_{i_\psi} \int (\psi_{ref} - \psi)dt + k_{d_\psi}(\dot{\psi}_{ref} - \dot{\psi}). \quad (3.66)$$

Para o controlo de posição, ao contrário do de orientação e de altura, o compensador produz uma aceleração de referência em vez de uma entrada de controlo:

$$x_{ref}'' = K_{p_x}(x_{ref} - x) + K_{i_x} \int (x_{ref} - x)dt + k_{d_x}(\dot{x}_{ref} - \dot{x}); \quad (3.67)$$

$$y_{ref}'' = K_{p_y}(y_{ref} - y) + K_{i_y} \int (y_{ref} - y)dt + k_{d_y}(\dot{y}_{ref} - \dot{y}). \quad (3.68)$$

Com as acelerações de referências calculadas, através da expressão (3.60) obtêm-se os ângulos de referência ϕ_{ref} e θ_{ref} de forma a controlar a orientação para que o *quadcopter* consiga alcançar a posição comandada.

O ajuste dos coeficientes dos controladores PID pode ser feito manualmente. Em [35] é descrito um processo iterativo de ajuste manual do PID. Inicialmente é atribuído um valor baixo, 0,5 ou menos, a K_p e 0 a K_i e K_d . Analisando a resposta do sistema a um degrau, aumenta-se o valor de K_p até ao ponto em que oscilações comecem a surgir. Nesse ponto, reduz-se para metade o valor de K_p . Depois começa-se a aumentar o valor de K_i para tornar a resposta mais rápida, mas se ocorrerem oscilações diminui-se um pouco o valor de K_i . Se for necessário, aumenta-se o valor de K_d para tornar mais rápida a resposta a perturbações. À medida que se aumenta K_i e K_d , pode ser necessário diminuir um pouco K_p .

Outro método de ajuste de PID é o método Ziegler-Nichols em que consiste inicialmente colocar a 0 os ganhos K_i e K_d e aumentar o ganho proporcional K_p até que o sistema alcance a fronteira da instabilidade. Nesse ponto, K_p é denominado por K_U e o período das oscilações é denominado de T_U . Com os valores de K_U e T_U , através da Tabela 3.2, são calculados os ganhos do controlador PID [34].

Tabela 3. 2 - Método de Ziegler - Nichols para o afinamento do controlador PID

Tipo de controlador	K_p	K_i	K_d
Proporcional (P) $G_c(s) = K_p$	$0,5K_U$	–	–
Proporcional – integral (PI) $G_c(s) = K_p + \frac{K_i}{s}$	$0,45K_U$	$\frac{0,54K_U}{T_U}$	–
Proporcional – Integral – Derivativo (PID) $G_c(s) = K_p + \frac{K_i}{s} + K_d s$	$0,6K_U$	$\frac{1,2K_U}{T_U}$	$\frac{0,6K_U T_U}{8}$

O Simulink do MATLAB também dispõe de um conjunto de ferramentas que permitem ajustar os coeficientes do PID de acordo com a resposta pretendida. Com o “*Simulink Design Optimization*” definem-se as características da resposta do sistema a uma entrada em degrau, e através de um processo iterativo são calculados os valores de PID de forma a se obter a resposta definida [36].

3.3.2 Controlador Fuzzy-PID

Os controladores PID convencionais são muito utilizados, mas na presença de não linearidades, de atrasos temporais, de perturbações e de mudanças dos parâmetros dos sistemas, o desempenho do sistema de controlo fica aquém do desejado [37]. A técnica de controlo Fuzzy-PID combina a teoria do controlo PID com a teoria de controlo Fuzzy. O Fuzzy-PID pretende estabelecer duas funções contínuas entre os parâmetros do PID e os valores do erro absoluto e da taxa de variação do erro. Esta técnica é caracterizada por ter um mecanismo de auto-ajuste, que automaticamente afina os parâmetros de controlo para obter uma saída de controlo ótima. A saída de controlo ótima, expressão (3. 69), é obtida ajustando os valores dos coeficientes $K_{P-Fuzzy}$, $K_{I-Fuzzy}$ e $K_{D-Fuzzy}$ tendo em consideração o valor absoluto do erro e a sua taxa de variação [10].

$$U_{Fuzzy-PID} = K_{P-Fuzzy}e(t) + K_{I-Fuzzy} \int_0^t e(\tau)d\tau + K_{D-Fuzzy} \frac{de(t)}{dt} \quad (3. 69)$$

No controlador Fuzzy são definidas um conjunto de regras no formato *if-then*, que quando aplicadas às entradas (valor absoluto e taxa de variação do erro), ajustam os coeficientes da expressão (3. 69) [10].

Na Figura 3. 5 está representado o diagrama de blocos do controlador Fuzzy-PID para o controlo da orientação do *quadcopter*. O erro gerado pela orientação de referência e a orientação medida pelo IMU é analisado pelo controlador Fuzzy, e a taxa de variação do erro também é analisada. Através dessa análise, os valores dos coeficientes $K_{P-Fuzzy}$, $K_{I-Fuzzy}$ e $K_{D-Fuzzy}$ são ajustados tendo em consideração um

conjunto de regras *if-then* do controlador Fuzzy. Definidos os coeficientes, o compensador PID produz a saída de controle $U_{Fuzzy-PID}$.

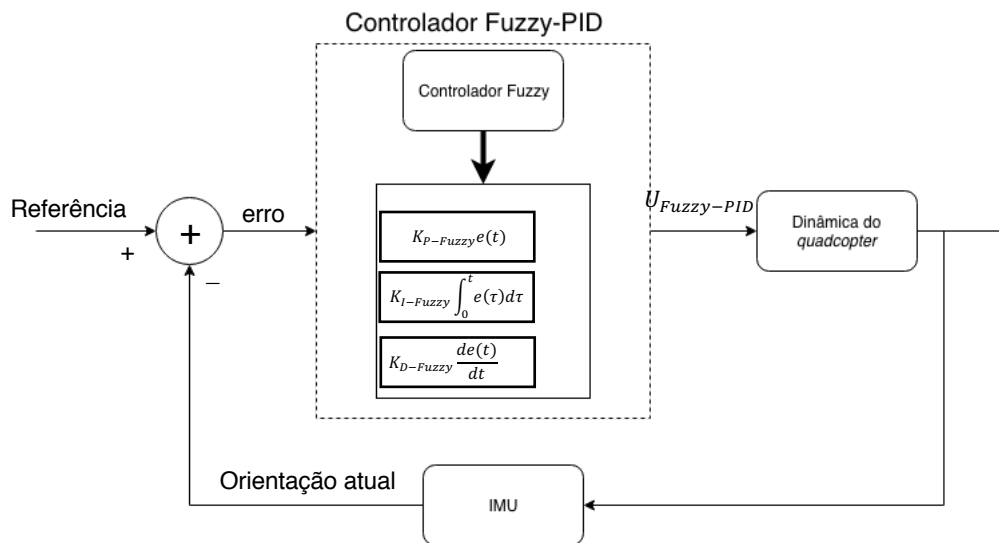


Figura 3. 5 – Diagrama de blocos do controlador Fuzzy-PID para o controle da orientação do quadcopter.

A entrada do controlador Fuzzy-PID é dada pelo sinal de erro e pela sua derivada. Consoante o valor destas duas variáveis de entrada, são atribuídos diferentes níveis linguísticos. O sinal de erro está limitado de -10 a 10 e os níveis linguísticos são: (NB) negativo grande; (NS) negativo pequeno; (ZE) zero; (PS) positivo pequeno e (PB) positivo grande. A derivada do erro tem os mesmos níveis linguísticos que o erro, mas está limitada entre -5 e 5. Assim qualquer decisão de controle é tomada consoante o nível linguístico do erro e da sua derivada [38]. Na Figura 3. 6 e Figura 3. 7 estão representadas as funções triangulares de associação do sinal de erro e da derivada do sinal de erro. Cada triângulo das funções corresponde a um conjunto fuzzy associado a um termo linguístico, e estão limitados entre 0 e 1, indicado assim o grau de associação da variável de entrada ao nível linguístico em questão.

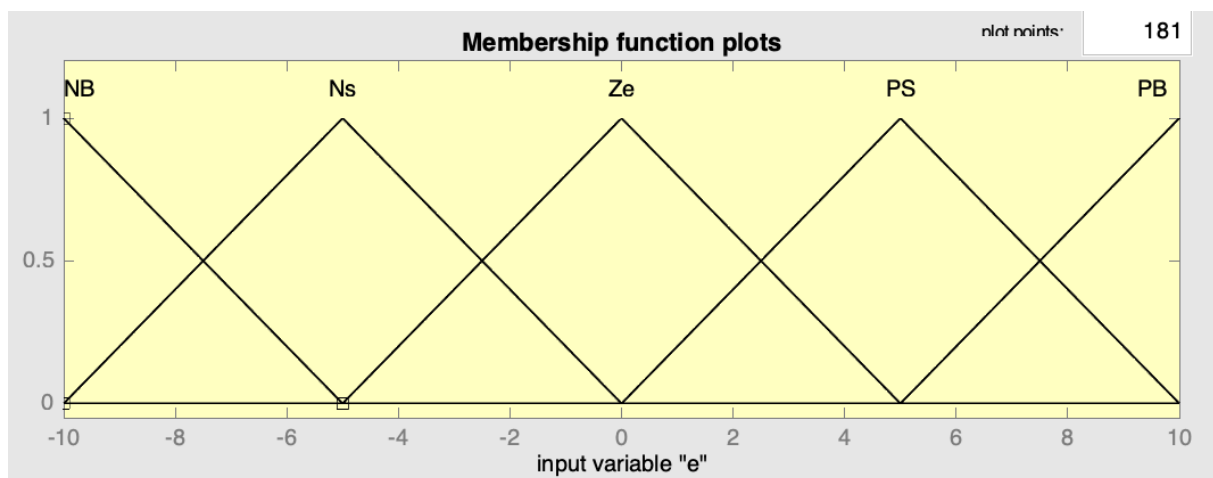


Figura 3. 6 - Função de associação da variável de entrada, erro.

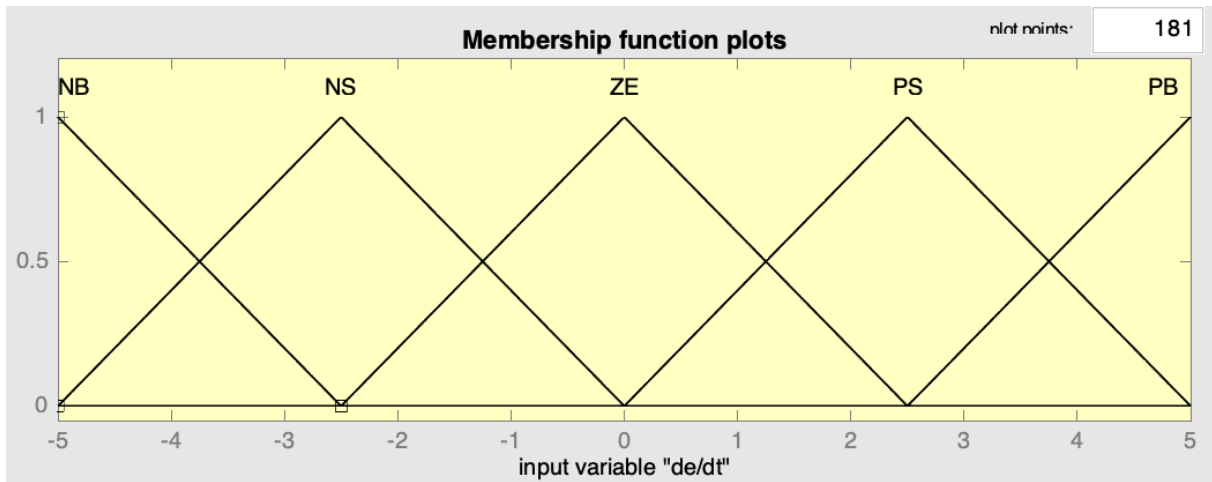


Figura 3. 7 - Função de associação da variável de entrada, derivada do erro.

As saídas do controlador Fuzzy são os ganhos proporcional, integral e derivativo do compensador PID. Estes ganhos também são definidos por funções de associação triangulares, Figura 3. 8 a Figura 3. 10. Os níveis linguísticos para o ganho proporcional são: (ZE) zero; (PS) positivo pequeno; (PM) positivo médio e (PL) positivo largo. Para os ganhos integral e derivativo os níveis linguísticos são: (ZE) zero; (PS) positivo pequeno e (PB) positivo grande. Os limites dos ganhos são de 0 a 10 para o proporcional, 0 a 1,5 para o integral e 0 a 5 para o derivativo [38].

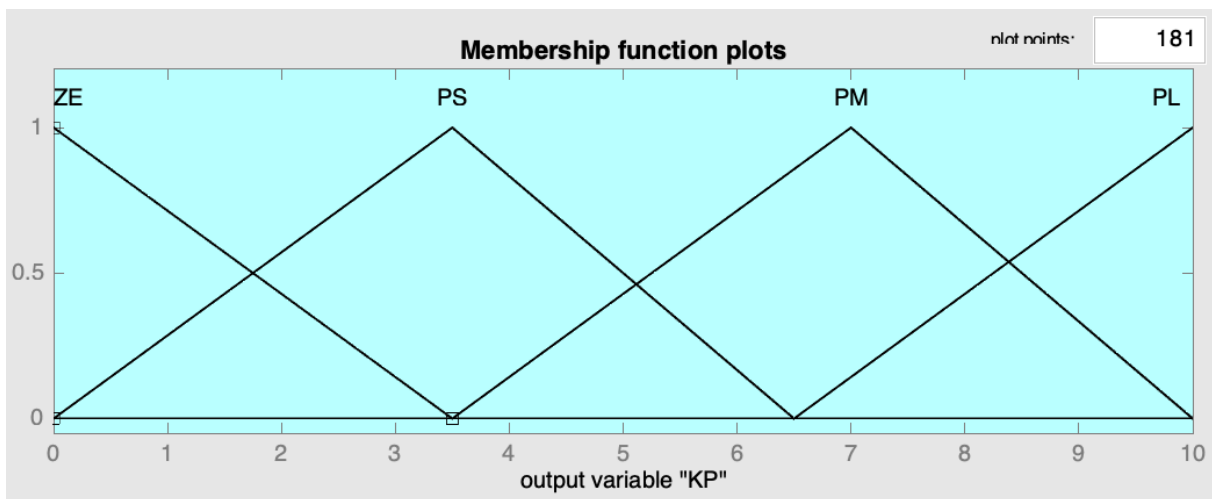


Figura 3. 8 - Função de associação da variável de saída, ganho proporcional.

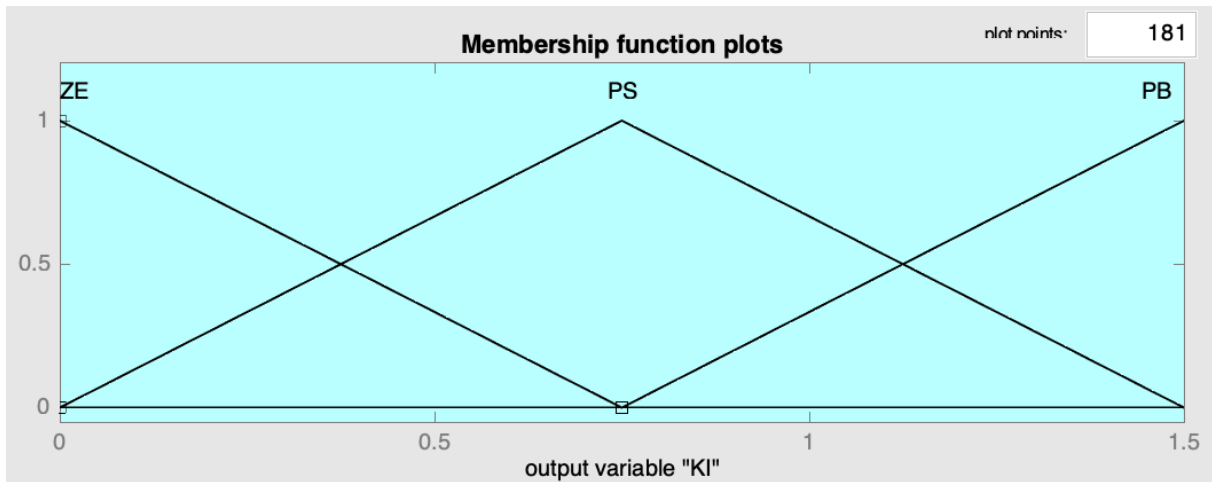


Figura 3. 9 - Função de associação da variável de saída, ganho integral.

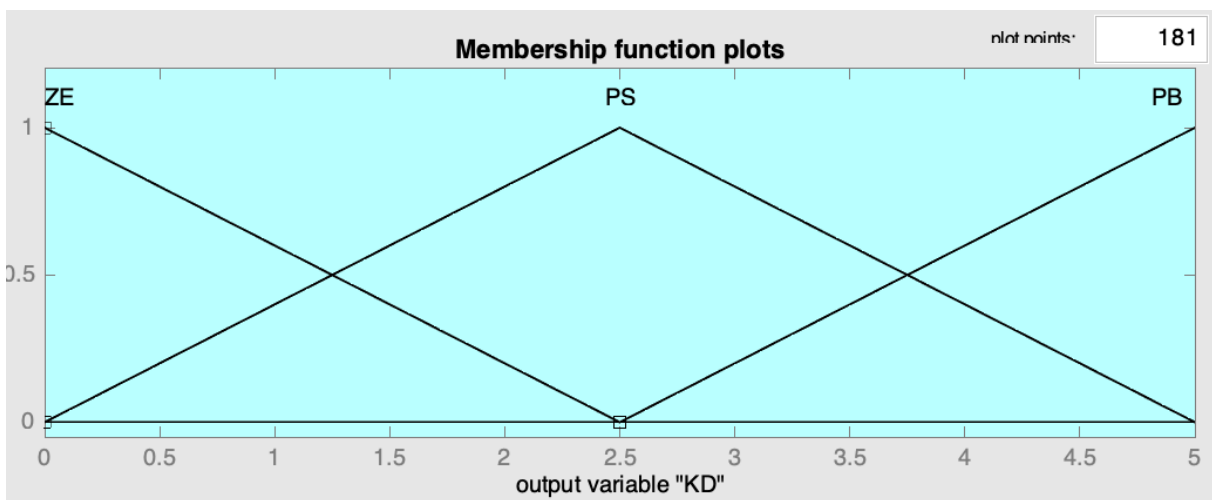


Figura 3. 10 - Função de associação da variável de saída, ganho derivativo.

As variáveis de saída do controlador são determinadas através das regras *if-then* da Tabela 3. 3 [38].

Tabela 3. 3 - Regras de inferência do controlador Fuzzy

Regra	if	then
1	$(e == PB) \ \&\& \ (de / dt == NB)$	$(K_p = ZE) \ \&\& \ (K_i = ZE) \ \&\& \ (K_d = PB)$
2	$(e == PB) \ \&\& \ (de / dt == NS)$	$(K_p = PS) \ \&\& \ (K_i = ZE) \ \&\& \ (K_d = PB)$
3	$(e == PB) \ \&\& \ (de / dt == ZE)$	$(K_p = PM) \ \&\& \ (K_i = PS) \ \&\& \ (K_d = PB)$
4	$(e == PB) \ \&\& \ (de / dt == PS)$	$(K_p = PL) \ \&\& \ (K_i = PS) \ \&\& \ (K_d = PS)$
5	$(e == PB) \ \&\& \ (de / dt == PB)$	$(K_p = PS) \ \&\& \ (K_i = PB) \ \&\& \ (K_d = PS)$
6	$(e == NB) \ \&\& \ (de / dt == PB)$	$(K_p = PL) \ \&\& \ (K_i = PB) \ \&\& \ (K_d = ZE)$
7	$(e == NB) \ \&\& \ (de / dt == PS)$	$(K_p = PM) \ \&\& \ (K_i = PB) \ \&\& \ (K_d = ZE)$
8	$(e == NB) \ \&\& \ (de / dt == ZE)$	$(K_p = PM) \ \&\& \ (K_i = PS) \ \&\& \ (K_d = ZE)$

Tabela 3. 3 (Continuação)

Regra	if	then
9	$(e == NB) \ \&\& \ (de / dt == NS)$	$(K_p = PS) \ \&\& \ (K_i = ZE) \ \&\& \ (K_d = PS)$
10	$(e == NB) \ \&\& \ (de / dt == NB)$	$(K_p = ZE) \ \&\& \ (K_i = ZE) \ \&\& \ (K_d = PS)$
11	$(e == ZE) \ \&\& \ (de / dt == NB)$	$(K_p = PS) \ \&\& \ (K_i = ZE) \ \&\& \ (K_d = PB)$
12	$(e == ZE) \ \&\& \ (de / dt == PB)$	$(K_p = PM) \ \&\& \ (K_i = PB) \ \&\& \ (K_d = PB)$
13	$(e == ZE) \ \&\& \ (de / dt == NS)$	$(K_p = PM) \ \&\& \ (K_i = PS) \ \&\& \ (K_d = PS)$
14	$(e == ZE) \ \&\& \ (de / dt == PS)$	$(K_p = PM) \ \&\& \ (K_i = PS) \ \&\& \ (K_d = PS)$
15	$(e == NS) \ \&\& \ (de / dt == ZE)$	$(K_p = PS) \ \&\& \ (K_i = PS) \ \&\& \ (K_d = PS)$
16	$(e == NS) \ \&\& \ (de / dt == NB)$	$(K_p = PS) \ \&\& \ (K_i = ZE) \ \&\& \ (K_d = PB)$
17	$(e == NS) \ \&\& \ (de / dt == NS)$	$(K_p = PS) \ \&\& \ (K_i = PS) \ \&\& \ (K_d = PS)$
18	$(e == NS) \ \&\& \ (de / dt == PB)$	$(K_p = PM) \ \&\& \ (K_i = PB) \ \&\& \ (K_d = PB)$
19	$(e == NS) \ \&\& \ (de / dt == PS)$	$(K_p = PM) \ \&\& \ (K_i = PS) \ \&\& \ (K_d = PS)$
20	$(e == PS) \ \&\& \ (de / dt == ZE)$	$(K_p = PM) \ \&\& \ (K_i = PS) \ \&\& \ (K_d = PS)$
21	$(e == PS) \ \&\& \ (de / dt == PB)$	$(K_p = PS) \ \&\& \ (K_i = PB) \ \&\& \ (K_d = PB)$
22	$(e == PS) \ \&\& \ (de / dt == PS)$	$(K_p = PS) \ \&\& \ (K_i = PS) \ \&\& \ (K_d = PS)$
23	$(e == PS) \ \&\& \ (de / dt == NB)$	$(K_p = PS) \ \&\& \ (K_i = ZE) \ \&\& \ (K_d = PB)$
24	$(e == PS) \ \&\& \ (de / dt == NS)$	$(K_p = PS) \ \&\& \ (K_i = PS) \ \&\& \ (K_d = PS)$

Os valores dos ganhos são obtidos através de um processo denominado de defuzzificação, que transforma os dados produzidos pelas regras de inferência em valores concretos [39].

Na Figura 3. 11 está representado o diagrama de blocos do sistema do controlador Fuzzy-PID.

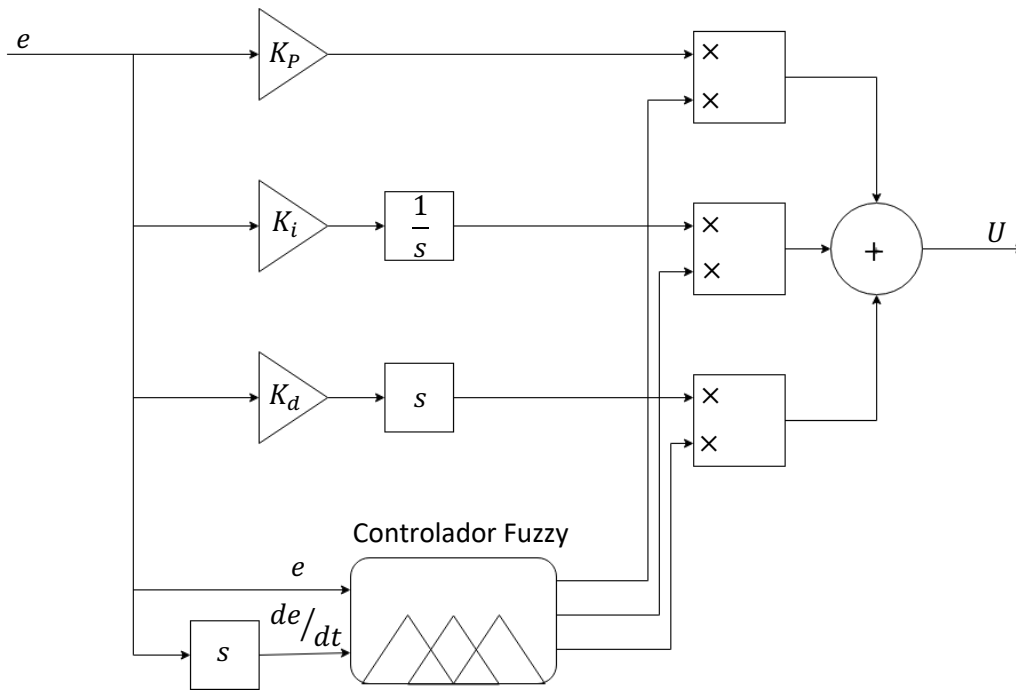


Figura 3. 11 - Diagrama de blocos do controlador Fuzzy-PID.

3.3.3 Controlador *Backstepping*

A técnica de controlo *backstepping* permite controlar a orientação e a altitude do *quadcopter*. Esta técnica é baseada na teoria da estabilidade de Lyapunov.

A teoria de Lyapunov é utilizada para obter conclusões sobre trajetórias de um sistema do tipo $\dot{x} = f(x)$ sem descobrir as trajetórias, isto é, sem resolver a equação diferencial. Um típico teorema de Lyapunov tem a seguinte estrutura [40]:

- Se existe uma função $V : \mathbf{R}^n \rightarrow \mathbf{R}$ que satisfaz algumas condições em V e \dot{V} ;
- Então, as trajetórias do sistema satisfazem alguma propriedade.

Um exemplo de um teorema de Lyapunov é o teorema que demonstra que as trajetórias do sistema são limitadas. Este teorema supõe que existe uma função V que satisfaz as seguintes condições:

- Todos os subconjuntos de V são limitados;
- $\dot{V}(z) \leq 0$ para todo z .

Assim todas as trajetórias estão limitadas, isto é, para cada trajetória x existe R de tal forma que $\|x(t)\| \leq R$ para todo $t \geq 0$. Neste caso V é denominada de função de Lyapunov, o que comprova que as trajetórias estão limitadas [40].

3.3.3.1 Controlo de *roll*

Para controlar o ângulo de *roll* (ϕ) através da técnica de *backstep*, primeiro é necessário ter em consideração o erro entre o valor do ângulo de referência e o valor atual do ângulo:

$$e_1 = x_{1_{ref}} - x_1. \quad (3.70)$$

Considerando a seguinte função positiva definida de Lyapunov [41]:

$$V_1 = \frac{1}{2} e_1^2. \quad (3.71)$$

Tendo em conta o modelo em variáveis de estado de (3.52) $\dot{x}_1 = x_2$, a derivada da função de Lyapunov V_1 fica:

$$\dot{V}_1 = e_1 \dot{e}_1 = e_1 (\dot{x}_{1_{ref}} - x_2). \quad (3.72)$$

Para que o sistema seja assintoticamente estável, a derivada da função de Lyapunov tem de ser negativa semi-definida. Para que tal aconteça, escolhe-se uma função de limitação $W_1(e) = c_1 e_1^2$ para limitar \dot{V}_1 [41]:

$$\dot{V}_1 = e_1 (\dot{x}_{1_{ref}} - x_2) \leq -c_1 e_1^2, \quad c_1 > 0, \quad (3.73)$$

em que c_1 é uma constante positiva. Para satisfazer a desigualdade da expressão anterior, define-se a entrada de controlo virtual como sendo [41]:

$$x_{2_{ref}} = \dot{x}_{1_{ref}} + c_1 e_1. \quad (3.74)$$

Definindo um novo erro, como sendo a diferença entre o valor atual do estado x_2 e o valor de referência:

$$e_2 = x_2 - x_{2_{ref}} = x_2 - \dot{x}_{1_{ref}} - c_1 e_1, \quad (3.75)$$

a derivada da função de Lyapunov pode ser reescrita em função dos erros e_1 e e_2 :

$$\begin{aligned} \dot{V}_1 &= e_1 \dot{e}_1 \\ &= e_1 (\dot{x}_{1_{ref}} - x_2) \\ &= e_1 (\dot{x}_{1_{ref}} - (e_2 + \dot{x}_{1_{ref}} + c_1 e_1)) \\ &= -e_1 e_2 - c_1 e_1^2. \end{aligned} \quad (3.76)$$

A presença do termo $e_1 e_2$ não garante que a derivada da função de Lyapunov seja negativa e semi-definida. Para tal, na próxima iteração do algoritmo de *backstepping*, adiciona-se à função V_1 um termo quadrático na variável de erro e_2 [41]:

$$V_2 = V_1 + \frac{1}{2}e_2^2. \quad (3.77)$$

A derivada de V_2 fica:

$$\begin{aligned} \dot{V}_2 &= \dot{V}_1 + e_2\dot{e}_2 \\ &= -e_1e_2 - c_1e_1^2 + e_2(\dot{x}_2 - \ddot{x}_{1ref} - c_1\dot{e}_1). \end{aligned} \quad (3.78)$$

Do modelo das variáveis de estado (3.52), \dot{x}_2 é dado em função da entrada de controle U_2 :

$$\dot{x}_2 = x_4x_6a_1 - x_4\Omega_r a_2 + b_1U_2. \quad (3.79)$$

Escolhendo a função positiva definida de limite $W_2(e) = -c_1e_1^2 - c_2e_2^2$, onde c_2 é uma constante positiva, e substituindo \dot{x}_2 de (3.79) em (3.78) obtém-se a seguinte desigualdade [32]:

$$\begin{aligned} \dot{V}_2 &= -e_1e_2 - c_1e_1^2 + e_2(x_4x_6a_1 - x_4\Omega_r a_2 + b_1U_2 - \ddot{x}_{1ref} - c_1\dot{e}_1) \\ &\leq -c_1e_1^2 - c_2e_2^2. \end{aligned} \quad (3.80)$$

Resolvendo a desigualdade anterior, a saída de controle U_2 pode ser expressa da seguinte forma:

$$U_2 = \frac{1}{b_1}(-c_2e_2 + e_1 - x_4x_6a_1 + x_4\Omega_r a_2 + \ddot{x}_{1ref} + c_1\dot{x}_{1ref} - c_1x_2). \quad (3.81)$$

3.3.3.2 Controle de *pitch*

O controlador para o ângulo de *pitch* é obtido da mesma forma que o controlador de *roll*. Neste caso o controlador tem origem nos seguintes estados:

$$\dot{x}_3 = x_4; \quad (3.82)$$

$$\dot{x}_4 = x_2x_6a_3 + x_2\Omega_r a_4 + b_2U_3. \quad (3.83)$$

Sendo o erro de *pitch* dado por:

$$e_3 = x_{3ref} - x_3. \quad (3.84)$$

A função de Lyapunov positiva definida é dada por:

$$V_3 = \frac{1}{2}e_3^2, \quad (3.85)$$

sendo a sua derivada:

$$\dot{V}_3 = e_3\dot{e}_3 = e_3(\dot{x}_{3ref} - x_4). \quad (3.86)$$

Com a função de limite $W_3(e) = -c_3e_3^2$, em que c_3 é uma constante positiva, o valor de referência de x_4 é:

$$x_{4ref} = \dot{x}_{3ref} + c_3 e_3. \quad (3.87)$$

E o erro entre o valor de x_4 e a referência é dado por:

$$e_4 = x_4 - \dot{x}_{3ref} - c_3 e_3. \quad (3.88)$$

Reescrevendo a derivada da função de Lyapunov tem-se:

$$\begin{aligned} \dot{V}_3 &= e_3 \dot{e}_3 \\ &= e_3 (\dot{x}_{3ref} - (e_4 + \dot{x}_{3ref} + c_3 e_3)) \\ &= -e_3 e_4 - c_3 e_3^2. \end{aligned} \quad (3.89)$$

Somando à função V_3 um termo quadrático com o erro e_4 , obtém-se outra função de Lyapunov:

$$V_4 = V_3 + \frac{1}{2} e_4^2. \quad (3.90)$$

Definindo uma nova função de limite:

$$W_4(e) = -c_3 e_3^2 - c_4 e_4^2, \quad (3.91)$$

em que c_4 é uma constante positiva, consegue-se obter a seguinte desigualdade:

$$\dot{V}_4 = -c_3 e_3^2 - c_4 e_4^2 + e_4 (\dot{x}_4 - \ddot{x}_{3ref} - c_3 e_3) \leq -c_3 e_3^2 - c_4 e_4^2 \quad (3.92)$$

Da expressão anterior, substitui-se \dot{x}_4 a partir da expressão (3.83) e resolve-se em ordem a U_3 , obtém-se a expressão para a variável de controlo do ângulo de *pitch*:

$$U_3 = \frac{1}{b_2} \left(-c_4 e_4 + e_3 - x_2 x_6 a_3 - x_2 \Omega_r a_4 + \ddot{x}_{3ref} + c_3 \dot{x}_{3ref} - c_3 x_4 \right). \quad (3.93)$$

3.3.3.3 Controlo de yaw

Para o controlador de *yaw*, a expressão para a variável de controlo do ângulo é obtida da mesma forma que para os controladores de *roll* e *pitch*. Neste caso a expressão para a entrada de controlo U_4 fica:

$$U_4 = \frac{1}{b_3} \left(-c_6 e_6 + e_5 - x_2 x_4 a_5 + \ddot{x}_{5ref} + c_5 \dot{x}_{5ref} - c_5 x_6 \right), \quad (3.94)$$

com:

$$e_5 = x_{5ref} - x_5; \quad (3.95)$$

$$e_6 = x_6 - \dot{x}_{5ref} - c_5 e_5. \quad (3.96)$$

Sendo c_5 e c_6 constantes positivas [32].

3.3.3.4 Controlo de altitude

O procedimento para a obtenção do controlador de altitude é o mesmo que o efetuado para os controladores de *roll*, *pitch* e *yaw*. Neste caso a expressão para a variável de controlo U_1 é a seguinte:

$$U_1 = \frac{m}{\cos(x_1) \cos(x_3)} \left(-e_7 + g - \ddot{x}_{7ref} - c_7 \dot{x}_{7ref} + c_7 x_8 + c_8 e_8 \right), \quad (3.97)$$

com:

$$e_7 = x_{7ref} - x_7; \quad (3.98)$$

$$e_8 = x_8 - \dot{x}_{7ref} - c_7 e_7. \quad (3.99)$$

Sendo c_7 e c_8 constantes positivas [32].

Com isto termina-se o terceiro capítulo, onde se abordou a dinâmica do *quadcopter* e as técnicas de controlo que permitem controlar a orientação e a altura da aeronave. No capítulo seguinte descreve-se a implementação em Simulink da dinâmica do *drone* e comparam-se os resultados obtidos entre as várias técnicas de controlo.

4 Simulação do controlo do *drone*

Neste capítulo será explicado como foi implementada em ambiente Simulink, a simulação da dinâmica e do controlo do *quadcopter* descritos no capítulo anterior.

Este capítulo está subdividido em quatro secções, sendo que na primeira são apresentados e descritos os dois modelos desenvolvidos, relativos à dinâmica do *drone*. Na segunda secção é descrita a implementação em Simulink dos diferentes controladores estudados. Na terceira secção são realizadas várias simulações aos modelos, com diferentes técnicas de controlo. Na quarta e última secção são comparados os resultados obtidos para as diferentes técnicas de controlo.

4.1 Implementação das equações do *drone*

Em Simulink implementou-se, em dois modelos distintos, a dinâmica do sistema do *quadcopter* descrita no capítulo 3. No primeiro modelo utilizou-se a função “Level-2 MATLAB S-Function” para implementar em linguagem MATLAB a dinâmica do *drone*, enquanto que no segundo modelo implementou-se o sistema através de um diagrama de blocos.

4.1.1 Modelo em linguagem MATLAB

O modelo em linguagem MATLAB do *quadcopter* é composto por 22 entradas e 12 saídas. Na Tabela 4. 1 e na Tabela 4. 2 estão descritas as entradas e saídas do modelo em linguagem MATLAB, do *quadcopter* implementado no Simulink.

Tabela 4. 1 - Descrição das entradas do modelo em linguagem MATLAB do *quadcopter*.

Entrada	Descrição
Ω_1	Velocidade angular do motor 1
Ω_2	Velocidade angular do motor 2
Ω_3	Velocidade angular do motor 3
Ω_4	Velocidade angular do motor 4
K_f	Constante de elevação das hélices
K_M	Constante de binário das hélices
d	Distância entre um motor e o centro do <i>quadcopter</i> (constante)
g	Aceleração gravítica (constante)
I_{xx}	Inércia do <i>quadcopter</i> em x (constante)
I_{yy}	Inércia do <i>quadcopter</i> em y (constante)
I_{zz}	Inércia do <i>quadcopter</i> em z (constante)
J_r	Inércia do motor (constante)
m	Massa do <i>quadcopter</i> (constante)

Tabela 4. 1 (Continuação)

Entrada	Descrição
p	Velocidade de rotação do <i>drone</i> sobre o eixo x (referencial do corpo)
q	Velocidade de rotação do <i>drone</i> sobre o eixo y (referencial do corpo)
r	Velocidade de rotação do <i>drone</i> sobre o eixo z (referencial do corpo)
ϕ	Ângulo de Euler ϕ (referencial fixo)
θ	Ângulo de Euler θ (referencial fixo)
ψ	Ângulo de Euler ψ (referencial fixo)
U	Velocidade linear do <i>drone</i> em relação ao eixo x (referencial do corpo)
V	Velocidade linear do <i>drone</i> em relação ao eixo y (referencial do corpo)
W	Velocidade linear do <i>drone</i> em relação ao eixo z (referencial do corpo)

Tabela 4. 2 - Descrição das saídas do modelo em linguagem MATLAB do *quadcopter*.

Saída	Descrição
p	Velocidade de rotação do <i>drone</i> sobre o eixo x (referencial do corpo)
q	Velocidade de rotação do <i>drone</i> sobre o eixo y (referencial do corpo)
r	Velocidade de rotação do <i>drone</i> sobre o eixo z (referencial do corpo)
ϕ	Ângulo ϕ (referencial fixo)
θ	Ângulo θ (referencial fixo)
ψ	Ângulo ψ (referencial fixo)
U	Velocidade linear do <i>drone</i> em relação ao eixo x (referencial do corpo)
V	Velocidade linear do <i>drone</i> em relação ao eixo y (referencial do corpo)
W	Velocidade linear do <i>drone</i> em relação ao eixo z (referencial do corpo)
x	Posição do <i>drone</i> no eixo x (referencial fixo)
y	Posição do <i>drone</i> no eixo y (referencial fixo)
z	Posição do <i>drone</i> no eixo z (referencial fixo)

Na Figura 4. 1 está representado o fluxograma que descreve o procedimento efetuado pelo modelo em código do *quadcopter*.

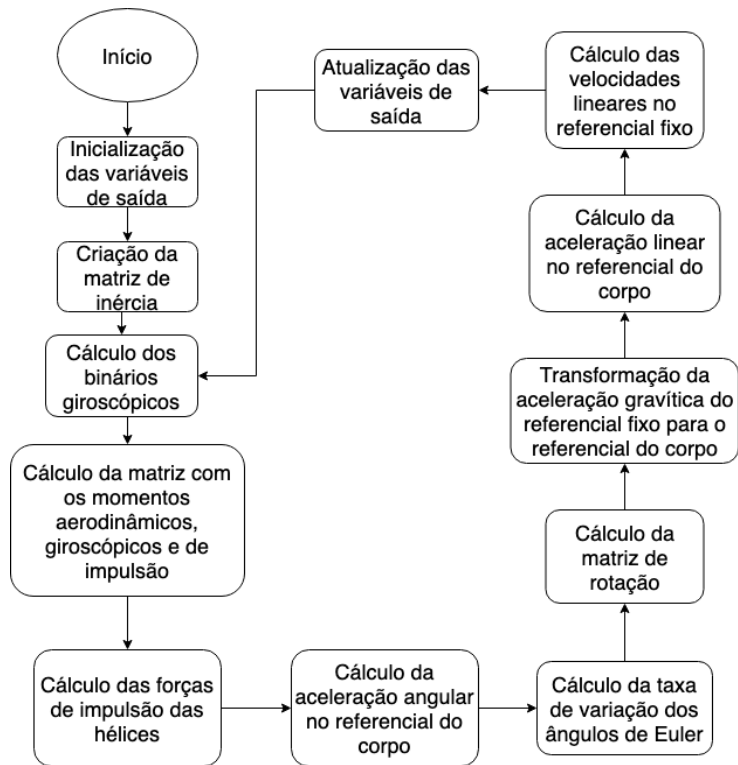


Figura 4. 1 - Fluxograma do modelo do *quadcopter* em linguagem MATLAB.

No anexo C está o código desenvolvido para a simulação da dinâmica do *drone*.

4.1.2 Modelo em diagrama de blocos

Para se implementar, em diagrama de blocos, o modelo do *quadcopter*, utilizaram-se os blocos “Fcn” do Simulink para implementar as equações da dinâmica. Na Figura 4. 2 está representado o diagrama de blocos que simula a dinâmica do *quadcopter*.

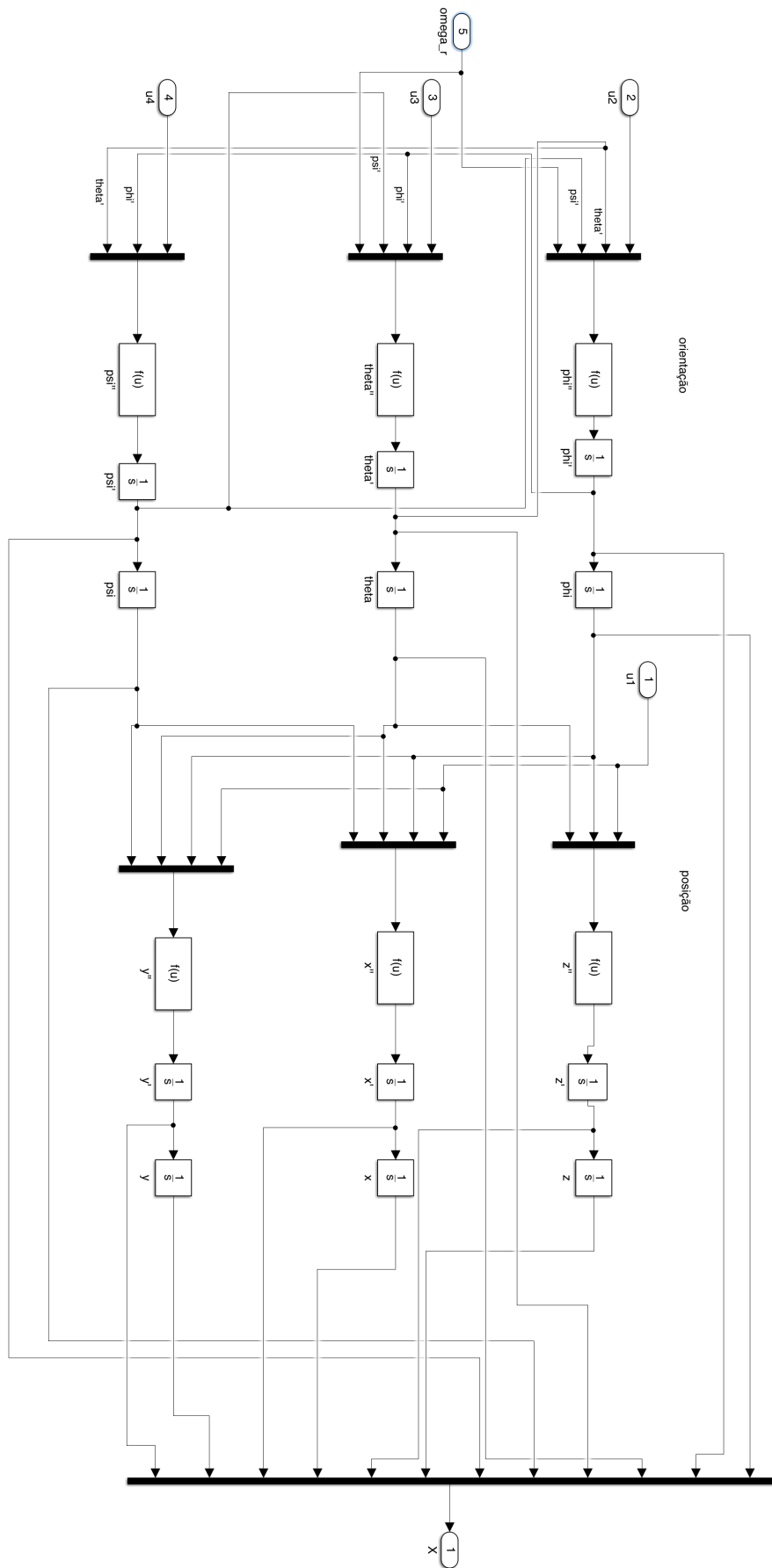


Figura 4. 2 - Diagrama de blocos para simulação da dinâmica do *quadcopter*.

Na Tabela 4. 3 estão indicadas a que equação da dinâmica corresponde cada bloco de funções do diagrama de blocos da simulação.

Tabela 4. 3 - Funções do diagrama e respetivas equações.

Função	Equação
$\ddot{\phi}$	(3. 41)
$\ddot{\theta}$	(3. 42)
$\ddot{\psi}$	(3. 43)
\ddot{z}	(3. 51)
\ddot{x}	(3. 49)
\ddot{y}	(3. 50)

Neste modelo não se colocou um limite inferior na posição vertical do *quadcopter*, ou seja, no início da simulação a aeronave começa suspensa e a sua altura diminui até que a força produzida pela rotação das hélices seja igual ou superior ao peso do *drone*.

4.2 Implementação dos controladores

Nos controladores de altura e orientação implementaram-se as técnicas de controlo descritas no capítulo 3, PID, Fuzzy-PID e *Backstepping*. Os diagramas de blocos em Simulink destes controladores estão presentes no anexo C, secção 3.

A forma do sinal de controlo gerado pelo controlador difere para cada um dos modelos. Enquanto que no modelo em diagrama de blocos o controlador define os sinais U_1 , U_2 , U_3 e U_4 , no modelo em linguagem MATLAB, o controlador define a velocidade angular de cada um dos motores.

4.2.1 Controlador do modelo em linguagem MATLAB

Na Figura 4. 3 está representado o controlador utilizado nas simulações com o modelo em linguagem MATLAB.

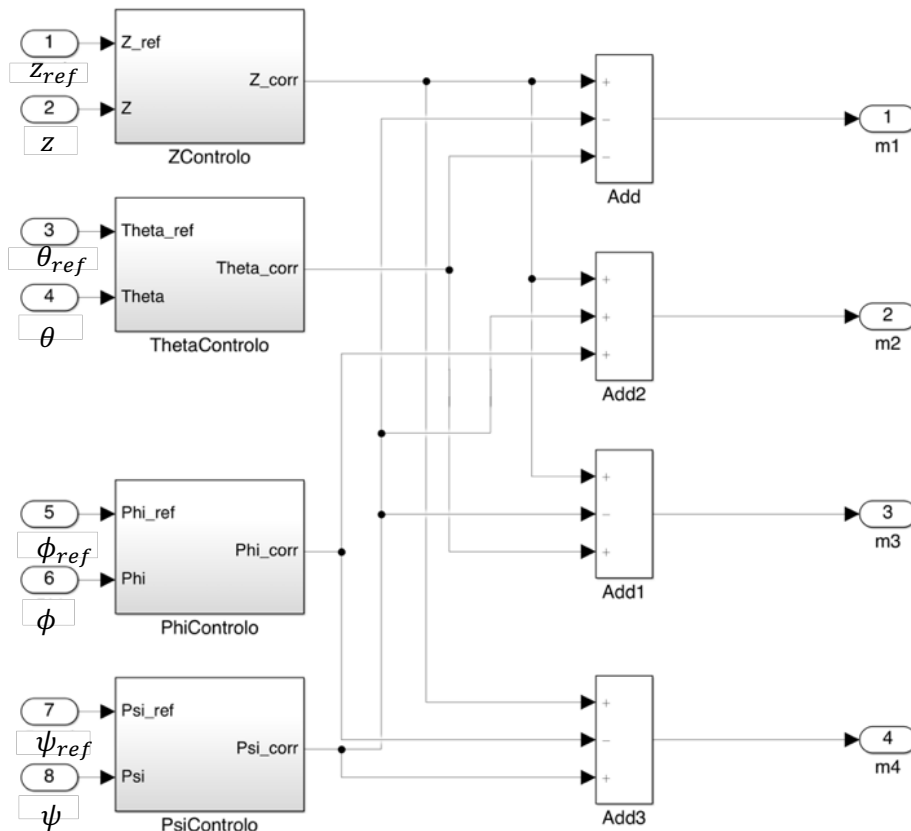


Figura 4. 3 - Controlador para o modelo em linguagem MATLAB.

Neste controlador as correções produzidas pelo controlo da orientação e de altitude são somadas de forma a definir a velocidade dos motores m_1 , m_2 , m_3 e m_4 . Neste caso, os motores 1 e 3 são responsáveis por controlar o ângulo θ e os motores 2 e 4 responsáveis por controlar ϕ . No controlo da altura e do ângulo ψ são necessários todos os motores.

Para que os sinais de saída do controlador da Figura 4. 3 (m_1 , m_2 , m_3 e m_4) possam definir a velocidade de rotação dos motores, primeiro estes têm de ser processados. O diagrama de blocos da Figura 4. 4 mostra como os sinais gerados pelo controlador são transformados em velocidades angulares.

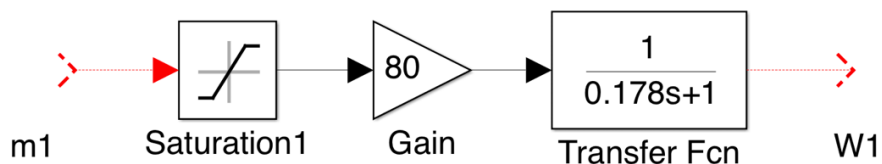


Figura 4. 4 - Diagrama de blocos para a obtenção da velocidade angular dos motores.

Na Figura 4. 4, o sinal gerado pelo controlador (m_1) primeiro passa por um limitador (bloco “Saturation1”) que limita esse sinal entre 0 e 100, depois é multiplicado por um ganho de 80. Com isto a velocidade dos motores fica compreendida entre 0 e 8000 rotações por minuto. O último bloco (“Transfer Fcn”) é um filtro de primeira ordem para impor um tempo de resposta do motor, neste caso o motor tem um tempo de subida

de aproximadamente 4 ms. O tempo de resposta do motor está relacionado com o sinal de comando do regulador de velocidade que opera a uma frequência de 250 Hz.

4.2.2 Controlador do modelo em diagrama de blocos

O controlador de orientação e de altura desenvolvido para o modelo em diagrama de blocos está representado na Figura 4. 5.

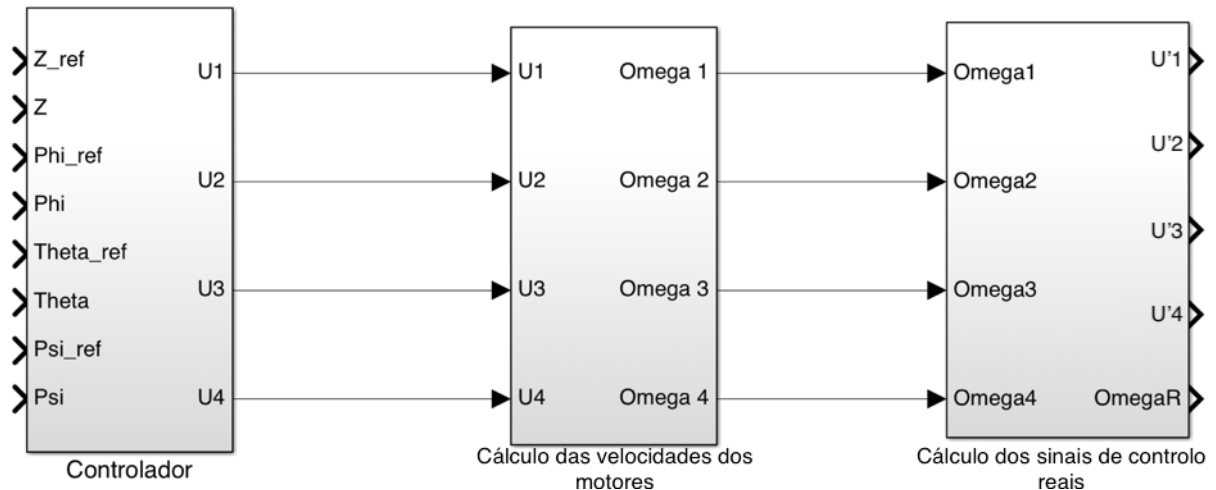


Figura 4. 5 - Controlador de orientação e altitude do modelo em diagrama de blocos.

O controlador gera os sinais de controlo U_1 , U_2 , U_3 e U_4 , sendo estes depois transformados em velocidades angulares, Figura 4. 6.

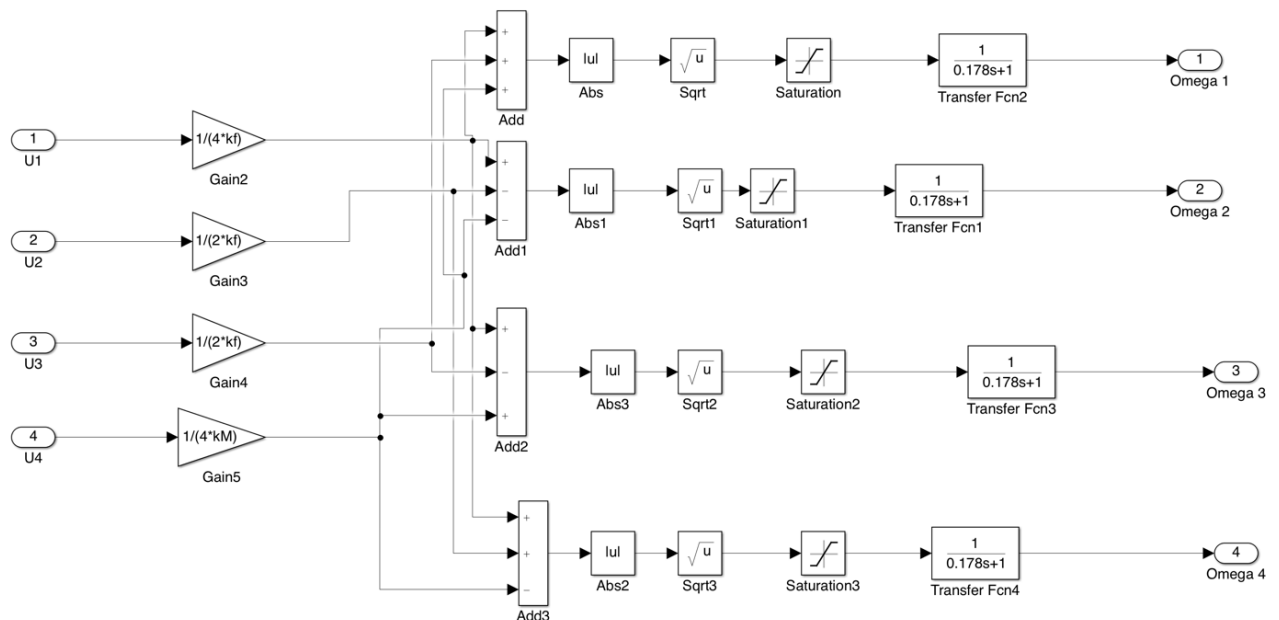


Figura 4. 6 - Diagrama de blocos para o cálculo das velocidades angulares dos motores.

O diagrama de blocos da Figura 4. 6 é o inverso da expressão (3. 32), permitindo assim determinar as velocidades Ω_1 , Ω_2 , Ω_3 e Ω_4 através dos sinais de controlo U_1 , U_2 , U_3 e U_4 . As velocidades dos motores estão também limitadas entre 0 e 8000 rotações e o tempo de resposta também é de aproximadamente 0,4 s.

Com os valores da velocidade dos motores calculados, esses são processados noutro diagrama de blocos, Figura 4. 7, para se determinarem a velocidade relativa dos motores (Ω_r) e os valores reais dos sinais de controlo U'_1 , U'_2 , U'_3 e U'_4 .

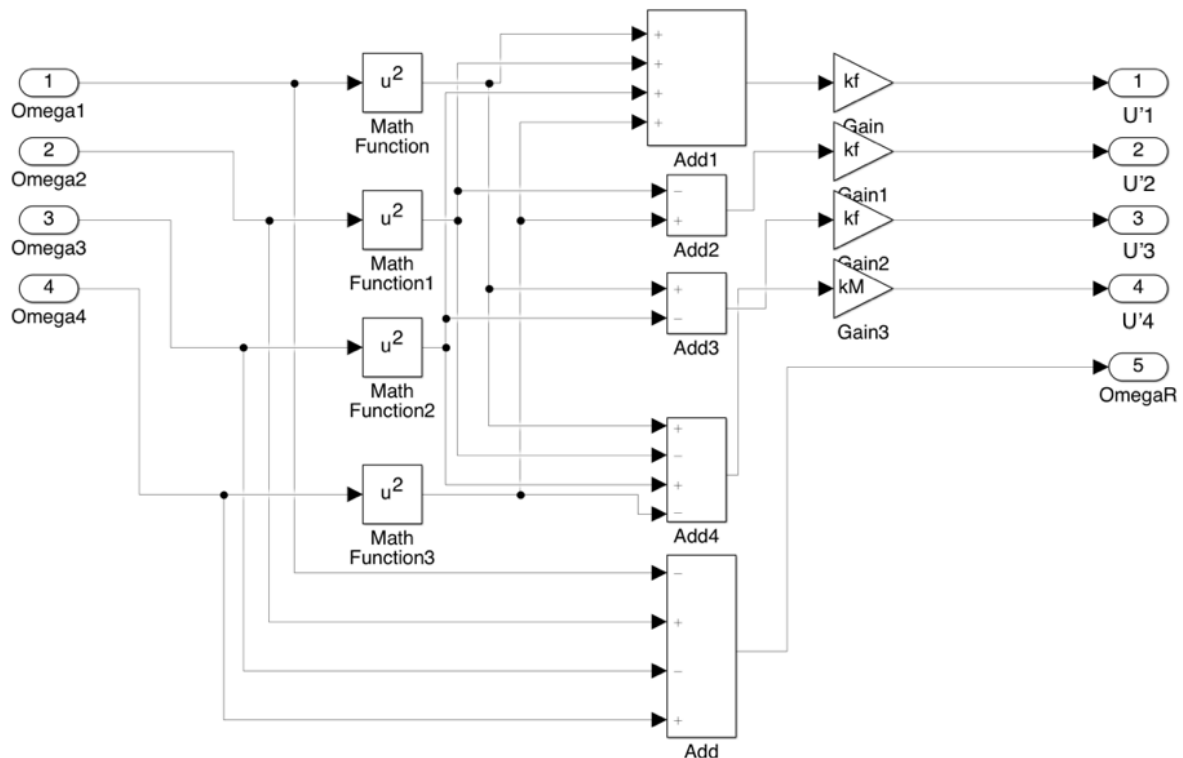


Figura 4. 7 - Diagrama de blocos para o cálculo dos sinais de controlo reais U'_1 , U'_2 , U'_3 , U'_4 e da velocidade relativa Ω_r .

O diagrama da Figura 4. 7 implementa as expressões (3. 6) e (3. 32) para se obterem os valores de Ω_r , U'_1 , U'_2 , U'_3 e U'_4 . A diferença entre o vetor de controlo $[U]$ à saída do controlador e à saída do diagrama de blocos da Figura 4. 7 é que este último já está limitado e inclui o tempo de resposta dos motores.

4.3 Simulação do controlo de altura e orientação

As características do *quadcopter* utilizado na simulação estão indicadas na Tabela 4. 4.

Tabela 4. 4 - Características do *quadcopter* da simulação.

Parâmetros	Valor
Massa	0,927 kg
Distância entre um motor e o centro	0,255 m
Inércia em x	$1,7067 \times 10^{-3} \text{ kg m}^2$
Inércia em y	$1,7067 \times 10^{-3} \text{ kg m}^2$
Inércia em Z	$0,032528 \text{ kg m}^2$
Inércia dos motores	$3,788 \times 10^{-6} \text{ kg m}^2$
Constante de impulsão das hélices	$1,4865 \times 10^{-7} \text{ N}/(\text{rad/s})^2$
Constante de binário das hélices	$2,925 \times 10^{-9} \text{ Nm}/(\text{rad/s})^2$

Como o valor da inércia da aeronave no eixo do y é o mesmo que para o eixo do x , apenas serão apresentados os resultados das simulações para o controlo do ângulo ϕ porque são iguais aos valores obtidos para o controlo do ângulo θ .

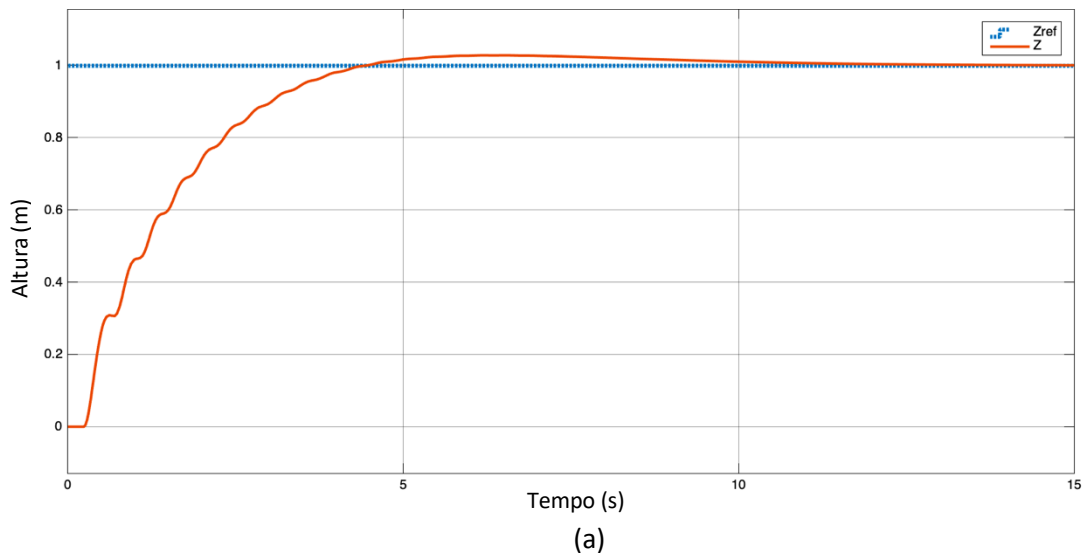
4.3.1 Controlo com a técnica PID

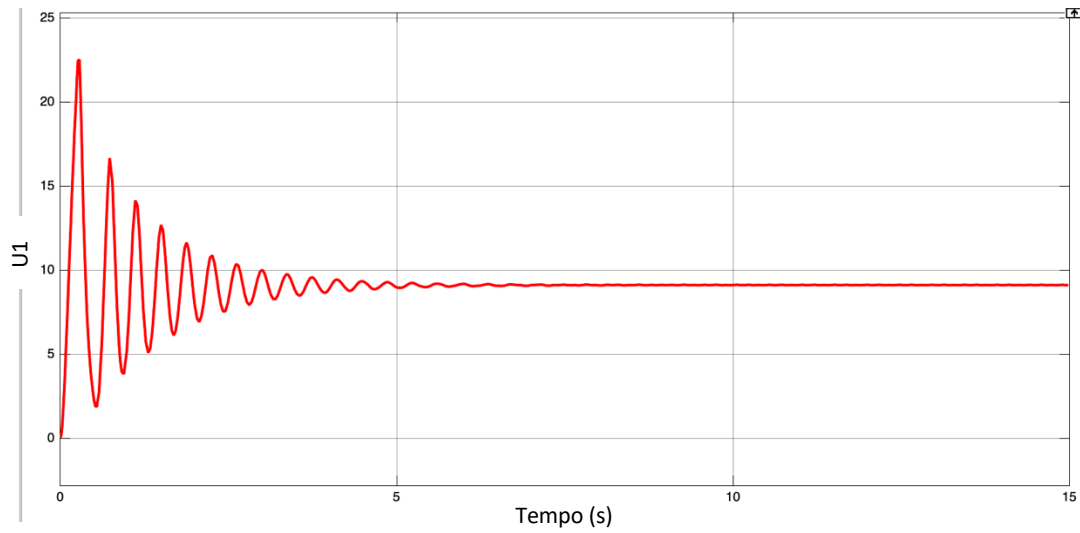
Na técnica PID através da análise da resposta do sistema à entrada em degrau, selecionaram-se os valores dos compensadores PID que tornavam a resposta mais próxima do desejado. Os valores obtidos para os controladores PID utilizados na simulação estão na Tabela 4. 5 e foram obtidos através da *toolbox* “*Simulink Design Optimization*”. Colocaram-se entradas em degrau unitário e os coeficientes PID foram obtidos de acordo com a resposta pretendida: *overshoot* inferior a 10 %, tempo de estabelecimento inferior a 7 s e tempo de subida inferior a 5 s.

Tabela 4. 5 - Valores dos compensadores utilizados na simulação.

Compensador	Controlo			
	ϕ	θ	ψ	z
Proporcional (K_p)	0,0381	0,0381	3,8189	121,6545
Integral (K_i)	$2,237 \times 10^{-20}$	$2,237 \times 10^{-20}$	$1,3275 \times 10^{-16}$	37,8513
Derivativo (K_d)	0,0804	0,0804	10,5879	131,2552

Simulou-se o controlo de altitude durante 15 segundos, com uma altura de referência de 1 m durante toda a simulação, Figura 4. 8.



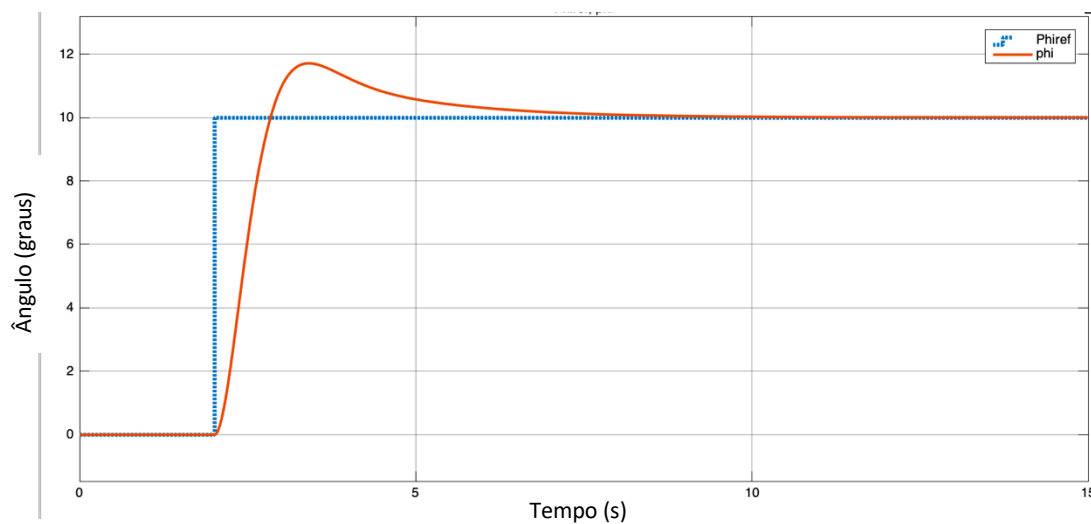


(b)

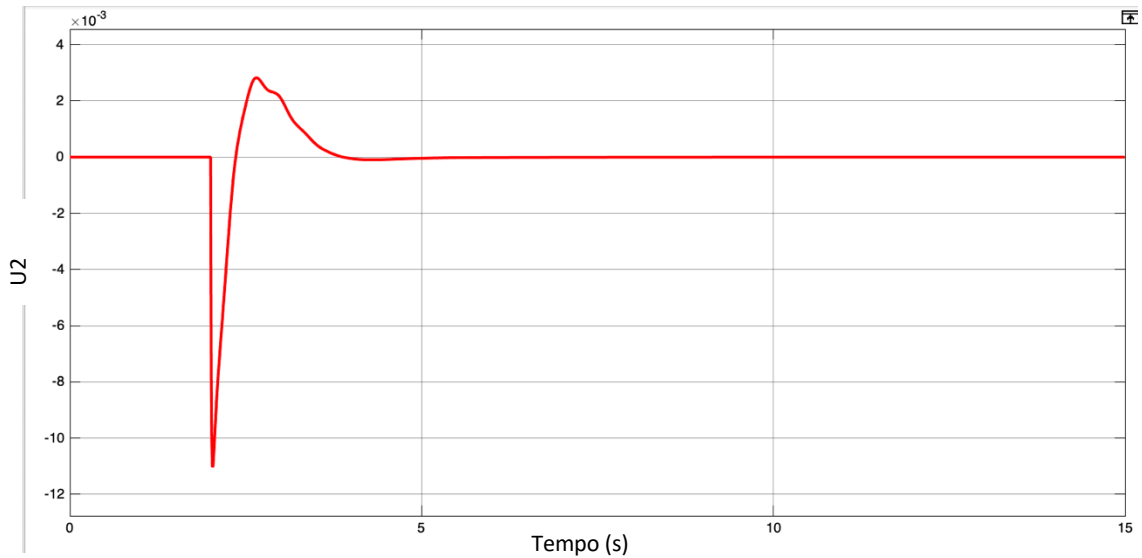
Figura 4. 8 - Controlo de altura com PID: a) altura e altura de referência; b) entrada de controlo U_1 .

Pela simulação do controlo de altura verificou-se que enquanto o *quadcopter* não consegue descolar, a entrada de controlo U_1 cresce rapidamente, até que o *quadcopter* consiga levantar voo. Depois, o valor de U_1 diminui, mas a aeronave continua a subir. As oscilações de U_1 provocam tremor no valor de z , mas este continua a subir até atingir o valor de referência. Quando é alcançado o valor de referência para a altura, U_1 fica com um valor constante de aproximadamente 9,2, situação em que a força de impulsão das hélices é igual ao peso do *quadcopter*.

Para simular o controlo do ângulo ϕ , colocou-se na referência uma entrada em degrau aos 2 s com um valor final de 10° .



(a)



(b)

Figura 4. 9 - Controle PID do ângulo ϕ : a) ângulo ϕ e referência; b) sinal de controle U_2 .

Com a simulação do controle do ângulo ϕ verificou-se que o ângulo converge para o valor de referência. Também se verificou que a variação do ângulo ϕ provoca ligeiras perturbações (na ordem dos 10^{-5}°) no ângulo θ , Figura 4. 10.

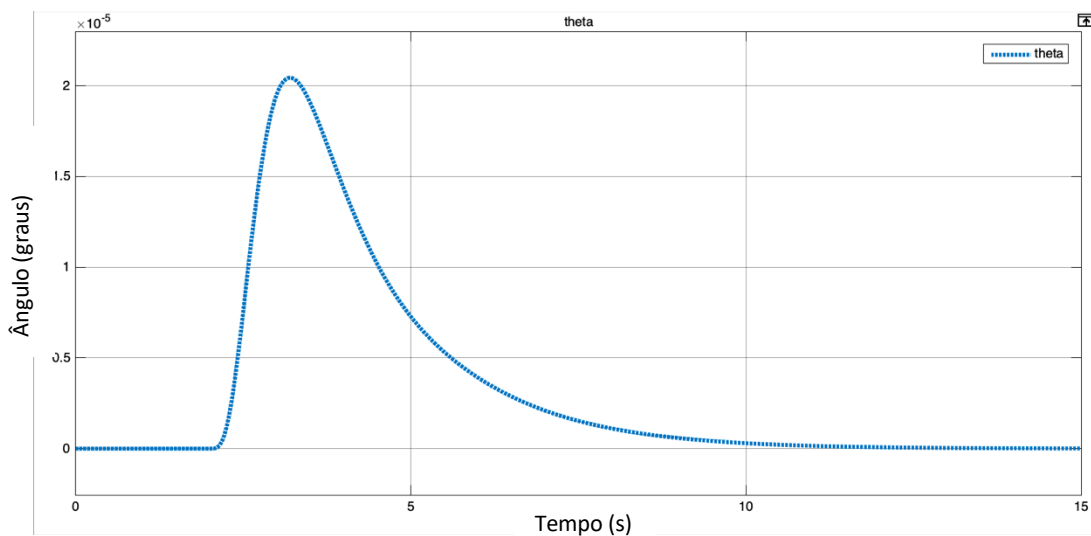


Figura 4. 10 - Ângulo θ durante o controle de ϕ .

A variação do ângulo θ na Figura 4. 10 é explicada através da equação (3. 39) em que diz que $\ddot{\theta}$ sofre a influência de $\dot{\phi}$. Esta variação do ângulo de *pitch* é logo suprimida pelo seu controlador. Na Figura 4. 11 visualiza-se a atuação do sinal de controle U_3 para eliminar as variações do ângulo θ . As mesmas variações também ocorrem para o ângulo ψ sendo eliminadas por U_4 .

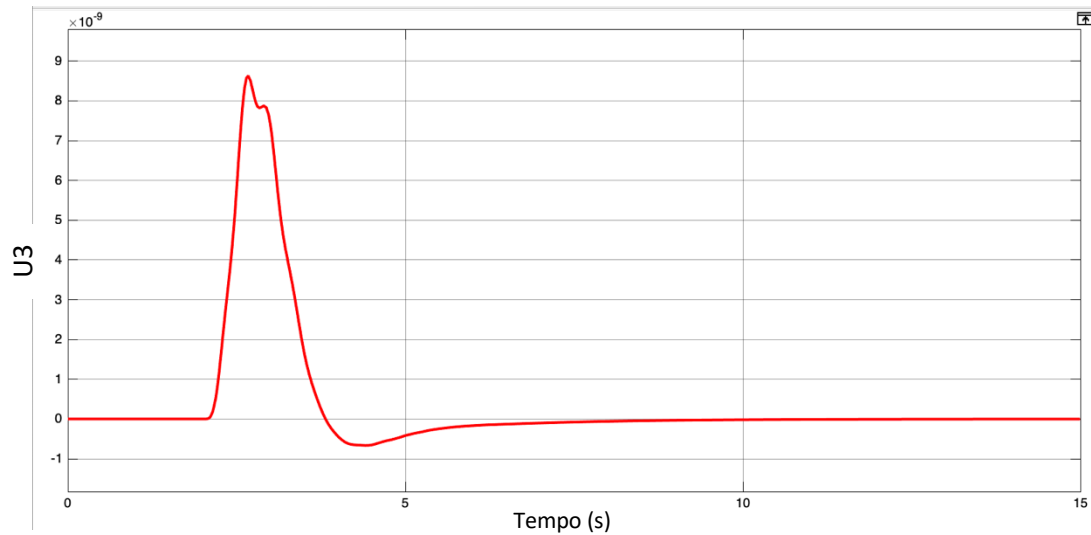
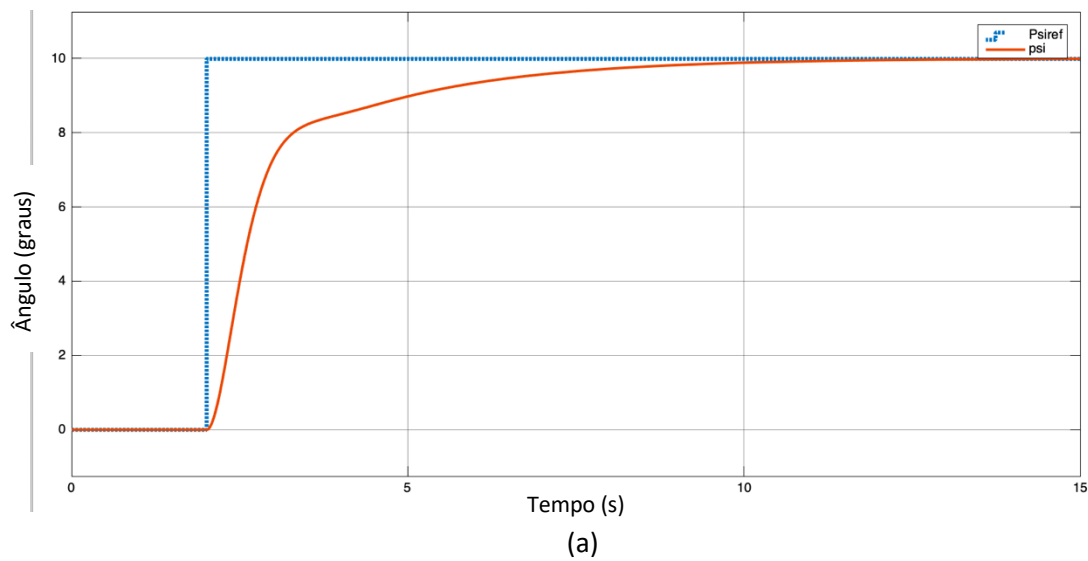
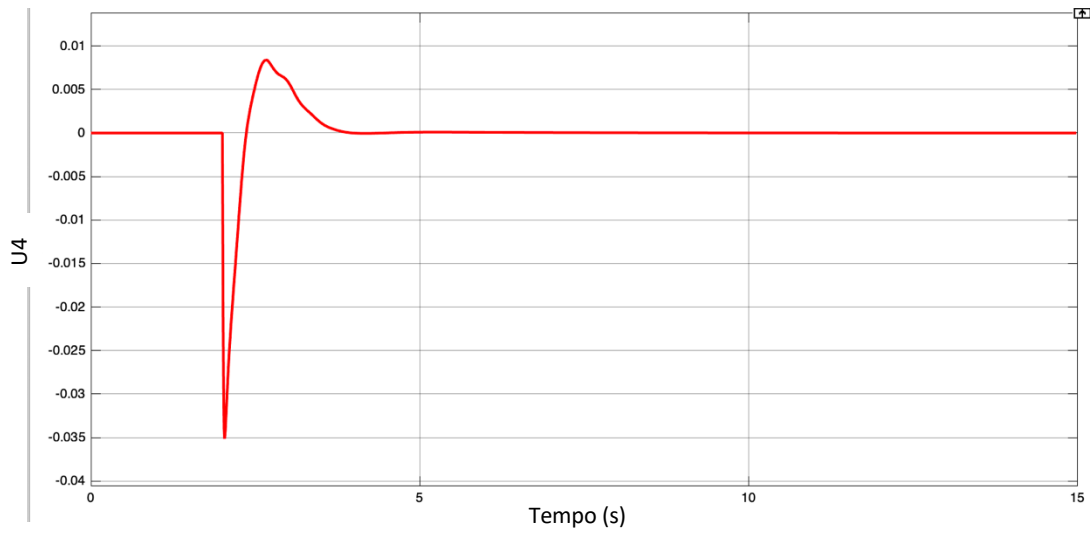


Figura 4. 11 - Sinal U_3 durante a simulação do controle de ϕ .

Para se simular o controle do ângulo ψ efetuou-se o mesmo procedimento que para o ângulo ϕ , ou seja, colocou-se na referência uma entrada em degrau aos 2 s com um valor final de 10° , Figura 4. 12.



(a)



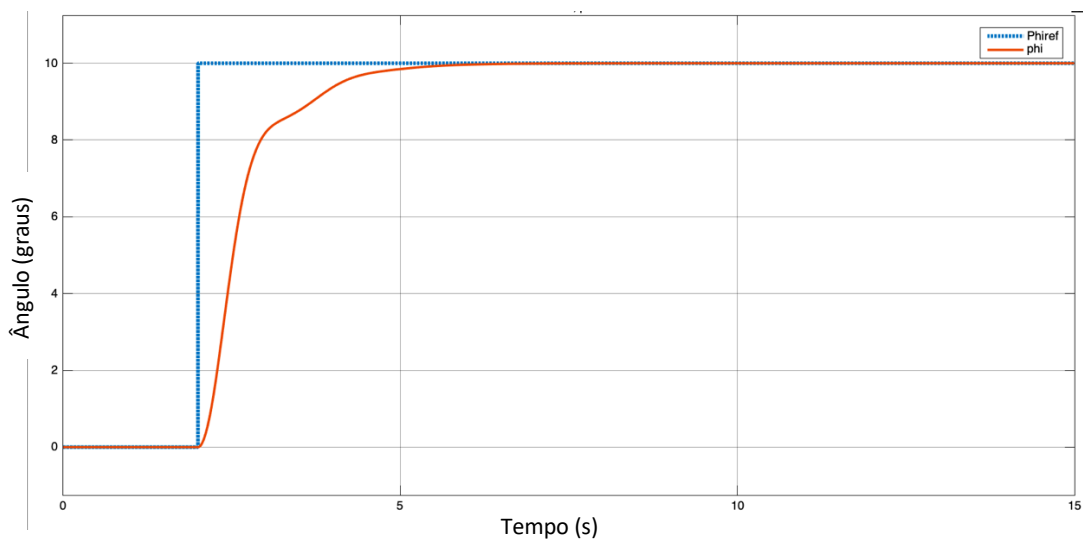
(b)

Figura 4. 12 - Controle PID do ângulo ψ : a) ângulo ψ e referência; b) sinal de controle U_4 .

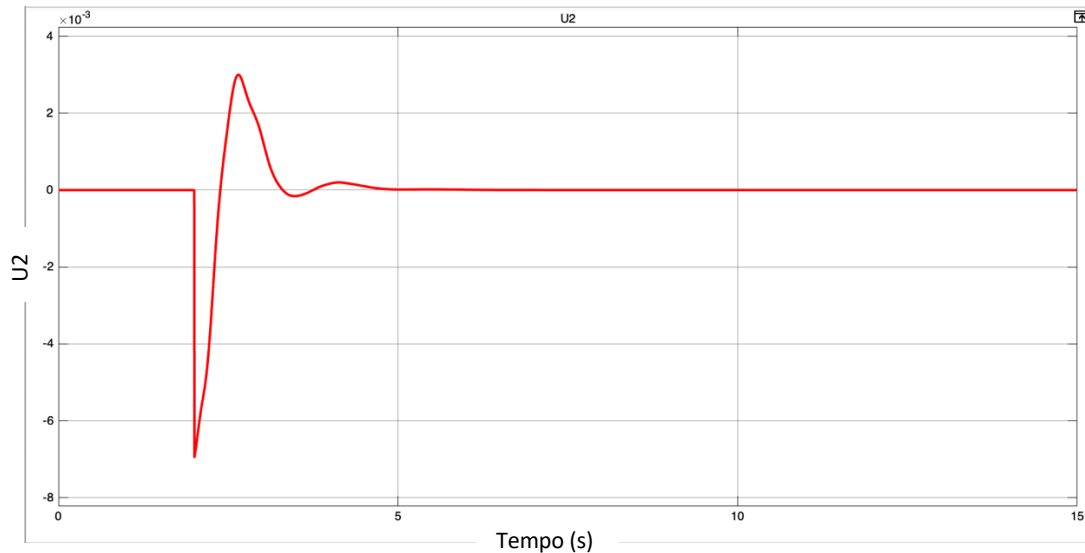
Com a simulação do controle do ângulo ψ , verificou-se que o ângulo converge para o valor de referência.

4.3.2 Controle com a técnica Fuzzy-PID

Utilizou-se a técnica Fuzzy-PID para controlar apenas a orientação do *quadcopter*. Na simulação do controle com esta técnica, colocou-se uma entrada em degrau aos 2 s com valor final de 10° na referência do ângulo ϕ e simulou-se durante 15 s, Figura 4. 13.



(a)

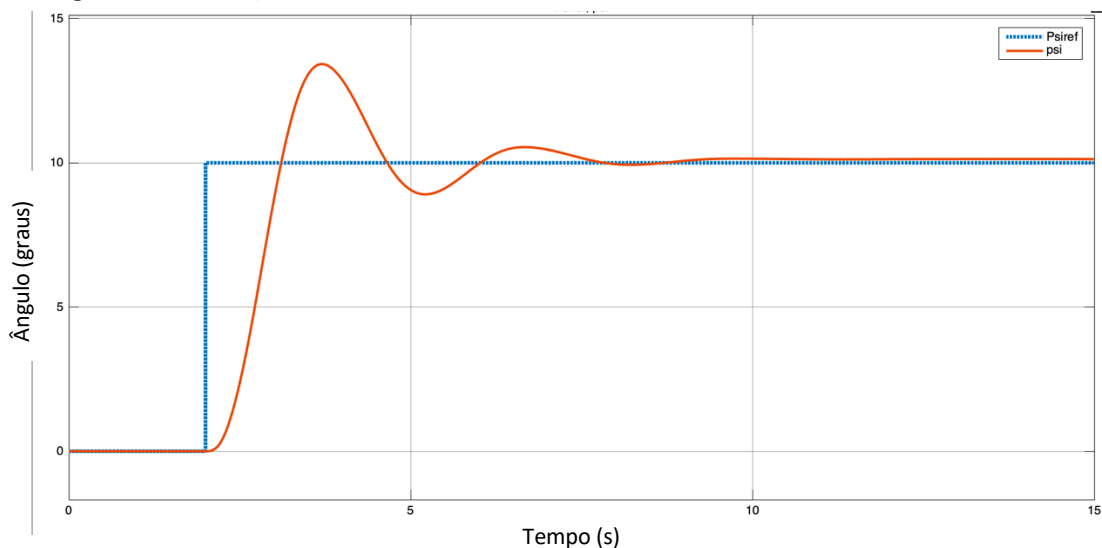


(b)

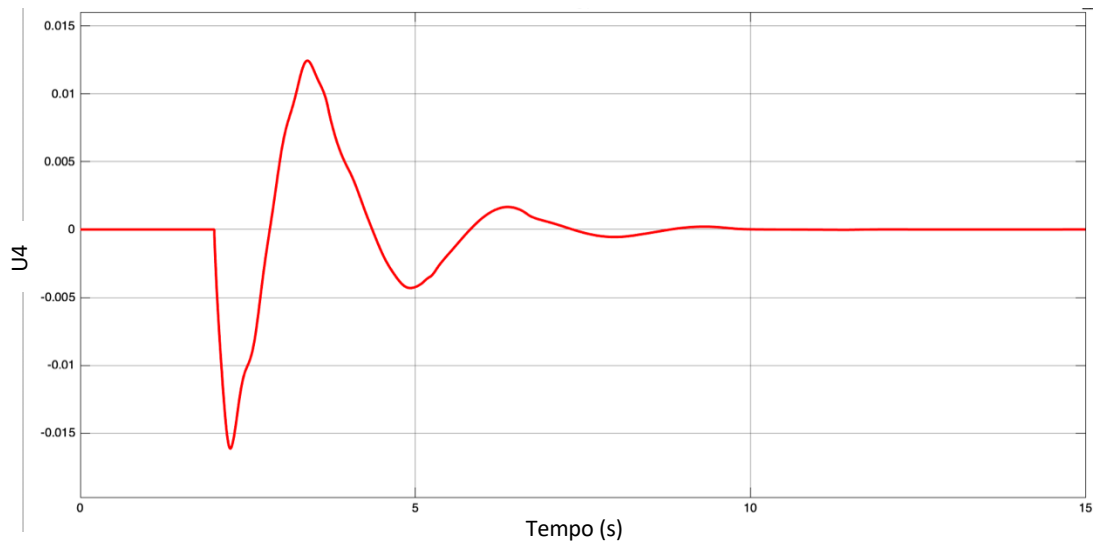
Figura 4. 13 - Controle do ângulo ϕ através da técnica Fuzzy-PID: a) ângulo ϕ e a sua referência; b) sinal de controle U_2 .

No controle de ϕ , os valores iniciais do compensador utilizado foram os mesmos que na técnica PID.

Para a simulação do controle do ângulo ψ , efetuou-se um procedimento semelhante ao anterior. Os valores iniciais do compensador PID foram 3 no proporcional, 0,5 no integral e 3,5 no derivativo. Na Figura 4. 14 estão representados o ângulo ψ e o sinal U_4 ao longo da simulação.



(a)



(b)

Figura 4. 14 - Controle do ângulo ψ com a técnica Fuzzy-PID: a) ângulo ψ e a referência; b) sinal de controle U_4 .

Apesar da lógica Fuzzy ser a mesma para ambos os controladores, a dinâmica e os valores do compensador PID são diferentes para cada ângulo, logo a resposta é diferente para cada controlador.

4.3.3 Controle com a técnica de *Backstepping*

Utilizou-se a técnica de *Backstepping* na simulação para controlar a altura do *quadcopter* e os ângulos ϕ e ψ . As constantes do controlador, identificadas no capítulo 3, foram obtidas através da ferramenta de otimização do Simulink (“*Simulink Design Optimization*”) e estão apresentadas na Tabela 4. 6.

Tabela 4. 6 - Constantes do controlador de *backstepping*

c_1	9,122
c_2	1,3593
c_3	9,122
c_4	1,3592
c_5	0,552
c_6	0,34
c_7	0,552
c_8	0,34

No controle de altura colocou-se como referência 1 m de altitude e simulou-se durante 15 s, Figura 4. 15.

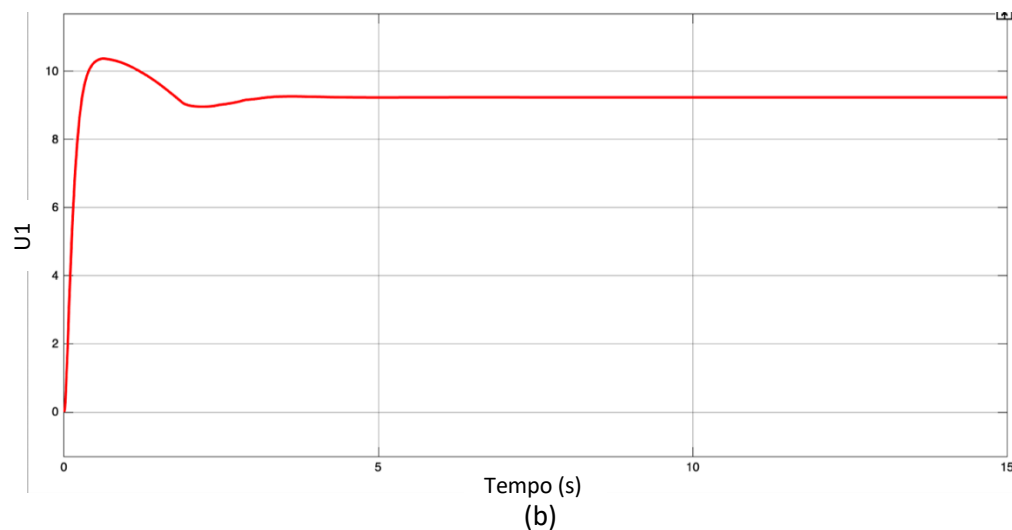
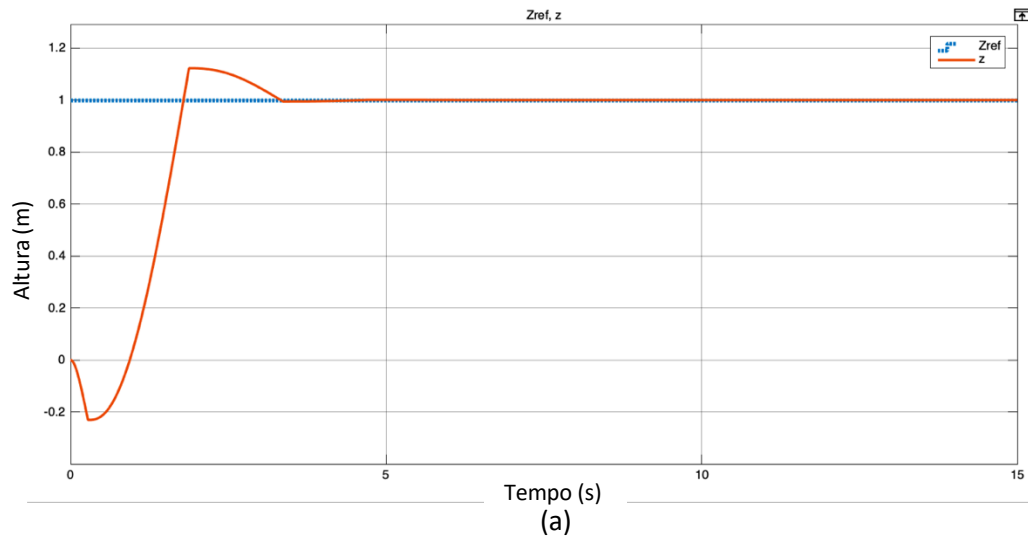
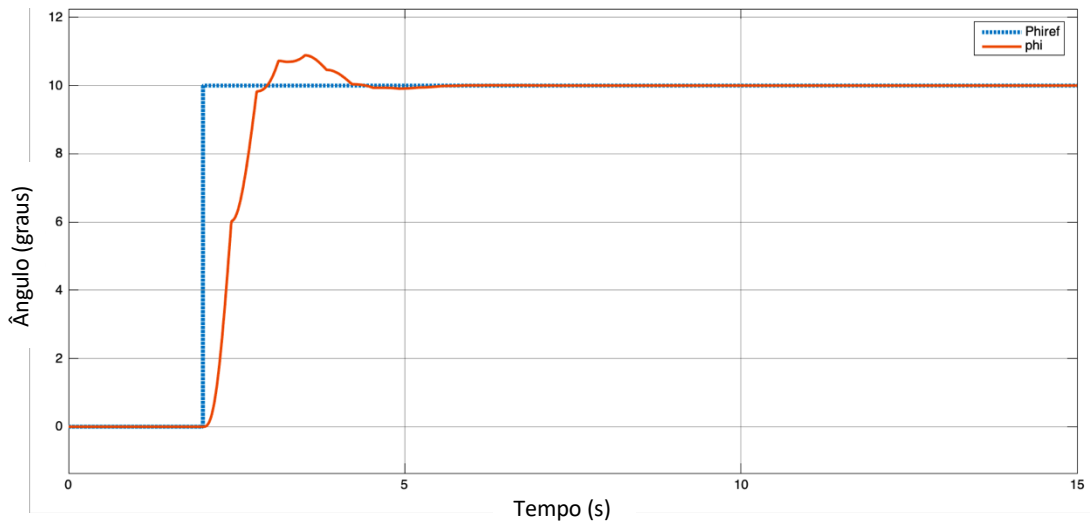


Figura 4. 15 - Controle de altura através da técnica de *backstepping*: a) altura e valor de referência; b) sinal de controle U_1 .

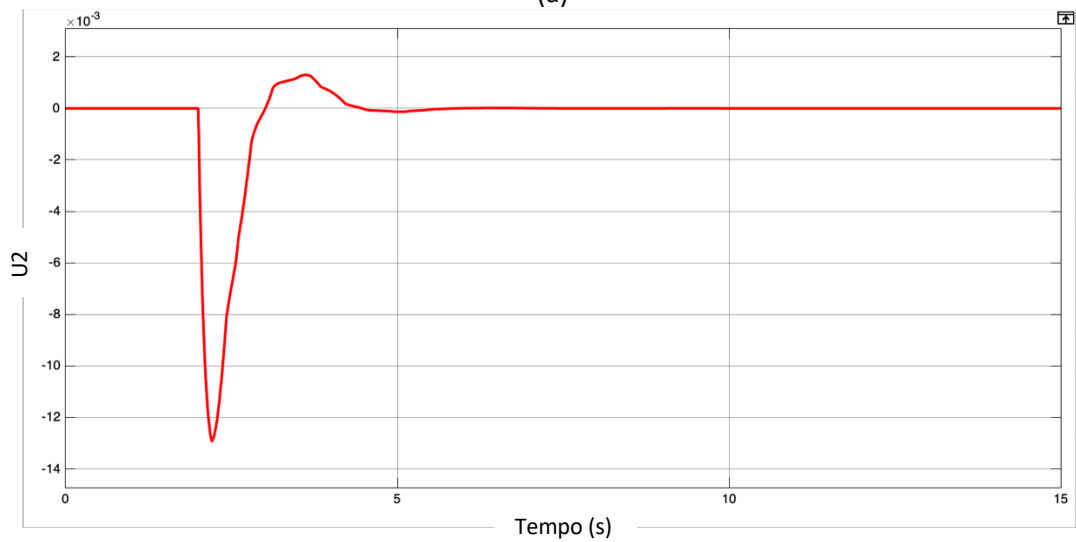
Para o controle com a técnica de *backstepping* utilizou-se o modelo em diagrama de blocos da dinâmica do *quadcopter*, e devido a isso observa-se que na simulação da Figura 4. 15.a) a altura do *drone* nos momentos iniciais é negativa.

Pela Figura 4. 15.a) verifica-se que a altura converge para o valor de referência e que à semelhança da técnica PID o valor do sinal de controle U_1 estabiliza aproximadamente em 9,2, fazendo com que o *quadcopter* fique a pairar no ar à altura de referência.

Para o controle do ângulo ϕ , colocou-se na referência uma entrada em degrau no instante de tempo de 2 s e valor final de 10° , Figura 4. 16.



(a)

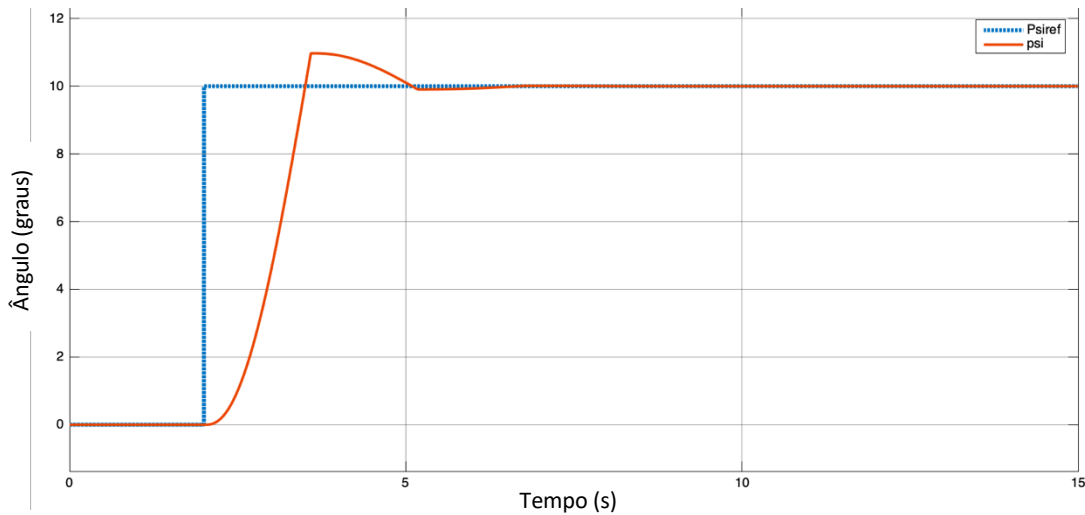


(b)

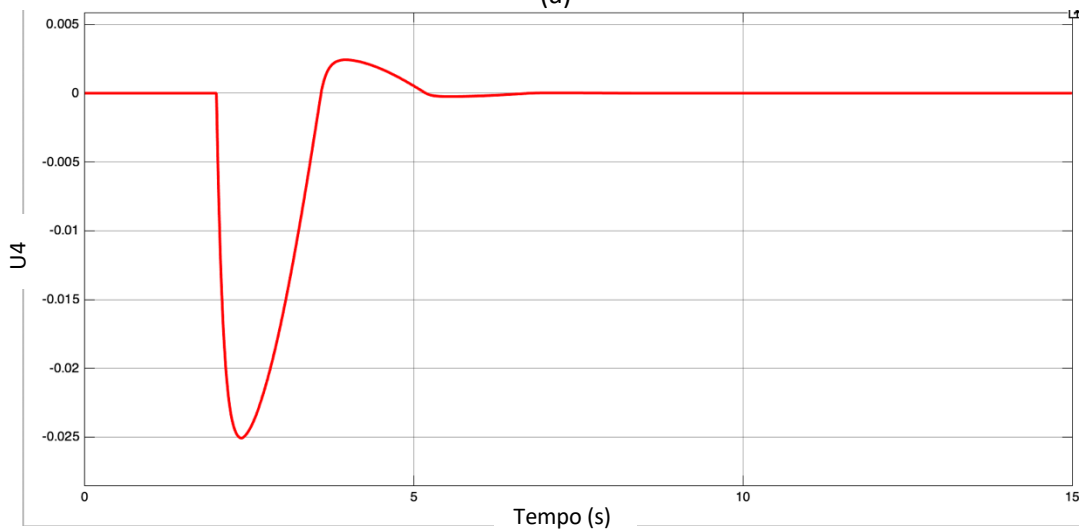
Figura 4. 16 - Controle do ângulo ϕ através da técnica de *backstepping*: a) ângulo ϕ e a sua referência; b) sinal de controle U_2 .

Com a simulação do controle de ϕ verificou-se que o ângulo convergiu para o valor de referência.

Na Figura 4. 17 observa-se a resposta do ângulo ψ a uma entrada em degrau aos 2 s, através da técnica de *backstepping*.



(a)



(b)

Figura 4. 17 - Controle do ângulo ψ através da técnica de *backstepping*: a) ângulo ψ e a sua referência; b) sinal de controle U_4 .

Com a simulação do controle do ângulo ψ , verificou-se que o ângulo convergiu para o valor de referência, 10° .

4.4 Comparação de resultados entre as técnicas

Com todas as técnicas de controle testadas na simulação conseguiu-se controlar a orientação e altura do *quadcopter*. Obtiveram-se respostas diferentes para cada técnica de controle, mas em todas o sistema convergiu para o valor de referência.

No controle de altura, com o controlador *backstepping* obteve-se uma resposta mais rápida, mas com um *overshoot* maior do que a resposta obtida pelo controlador PID. O desempenho destes dois controladores está diretamente ligado aos valores utilizados nos compensadores K_p , K_i e K_d e nas constantes c_7 e c_8 , e como estes foram obtidos através de um processo iterativo, o desempenho dos controladores

ainda pode ser melhorado através de um ajuste mais aprofundado destes valores. Na Tabela 4. 7 apresenta-se as características das respostas dos dois controladores de altura.

Tabela 4. 7 - Características das respostas dos controladores de altura

	PID	Backstepping
<i>Overshoot</i>	2,8 %	12,2 %
Tempo de subida	2,695 s	0,656 s
Tempo de estabelecimento	3,602 s	3,019 s

No controlo dos ângulos ϕ e θ , com a técnica PID obteve-se uma resposta mais rápida do que com o Fuzzy-PID, mas mais lenta do que o *backstepping*. Em termos de *overshoot*, a percentagem foi maior para o PID do que no *backstepping*, e nulo na técnica Fuzzy-PID. Na Tabela 4. 8 apresenta-se a percentagem de *overshoot*, o tempo de subida e o tempo de estabelecimento das diferentes respostas dos controladores dos ângulos ϕ e θ .

Tabela 4. 8 - Características das respostas dos controladores dos ângulos ϕ e θ

	PID	Fuzzy-PID	Backstepping
<i>Overshoot</i>	17,1 %	0 %	8,8 %
Tempo de subida	0,545 s	1,497 s	0,530 s
Tempo de estabelecimento	3,229 s	2,137 s	1,809 s

No controlo do ângulo ψ obteve-se com o PID uma percentagem de *overshoot* nula, mas das três técnicas foi a resposta mais lenta. Com o Fuzzy-PID obteve-se a maior percentagem de *overshoot* mas com tempo de subida mais rápido. Com o *backstepping* obteve-se a resposta com o melhor tempo de estabelecimento. Na Tabela 4. 9 são apresentadas as características das respostas de cada técnica para o controlo do ângulo ψ .

Tabela 4. 9 - Características das respostas dos controladores do ângulo ψ

	PID	Fuzzy-PID	Backstepping
<i>Overshoot</i>	0 %	34,2 %	9,7 %
Tempo de subida	2,858 s	0,648 s	0,914 s
Tempo de estabelecimento	4,697 s	4,898 s	2,605 s

À semelhança do controlo de altura, no controlo da orientação, o desempenho dos controladores pode ser melhorado através do aprofundamento do processo de iterativo no cálculo dos compensadores PID e das constantes das equações do *backstepping*.

Observando-se todas as respostas obtidas na simulação, consegue-se concluir que: a técnica PID é a técnica de controlo mais simples de implementar, porque para cada

variável de controlo diferente esta técnica apenas muda os valores dos compensadores; a técnica Fuzzy-PID no controlo dos ângulos ϕ e θ é a que tem a resposta mais lenta, mas a única com percentagem de *overshoot* nula; a técnica de *backstepping* tem a melhor relação entre tempo de resposta e percentagem de *overshoot* de todas as técnicas simuladas. Se se ajustarem de forma diferente os parâmetros dos compensadores, a resposta também pode mudar.

5 Verificação experimental do *quadcopter*

Neste capítulo descreve-se o procedimento realizado para implementação experimental de um *quadcopter*.

Na primeira secção descreve-se o processador utilizado e o motivo pelo qual se optou por esse microcontrolador. Na segunda secção descreve-se como foi feita a montagem do sistema de controlo remoto e como este funciona. Na terceira secção explica-se o funcionamento do sensor utilizado para monitorizar a orientação da aeronave. Na quarta secção descreve-se o funcionamento dos controladores de velocidade em conjunto com os motores BLDC. Na quinta secção apresenta-se a estrutura utilizada na construção do *quadcopter*. Por último, na sexta secção, explica-se o algoritmo implementado para controlar o voo do *drone*.

5.1 Processador

O microcontrolador utilizado para a implementação do controlador de voo do *drone* foi o Arduino Uno. Este microcontrolador tem o processador ATmega328P de 8 *bits* com uma velocidade de relógio de 16 MHz. Possui 14 portas digitais de entrada/saída (6 das quais podem ser usadas como saída PWM) e 6 entradas analógicas [42].

A programação do Arduino Uno foi feita em linguagem C no Arduino *Software* (IDE). O motivo pelo qual se optou pelo Arduino em vez de outros microcontroladores, deveu-se ao facto do Arduino ser uma plataforma com muitos utilizadores e por consequência disso ter uma grande variedade de bibliotecas de funções que facilitam a programação. Neste projeto em específico recorreu-se a três bibliotecas do Arduino: “Servo.h” para enviar os sinais de PWM para os controladores de velocidade dos motores; “Wire.h” para estabelecer a comunicação entre o microcontrolador e o sensor de orientação do *quadcopter*; “EnableInterrupt.h” para implementar interrupções no microcontrolador e assim poder ler os sinais do recetor rádio.

5.2 Telecomando

Para o controlo remoto do *quadcopter* utilizou-se o transmissor Turnigy 9x de nove canais (Figura 2. 28.a)) e o recetor Turnigy 9x de 8 canais (Figura 2. 28.b)). Este sistema de controlo remoto funciona à frequência de 2,4 GHz em que a potência máxima de transmissão é de 20 dBm e a sensibilidade do recetor é de -105 dBm. A comunicação entre o transmissor e o recetor é feita através do sistema de espalhamento espectral AFHDS 2A (*Automatic Frequency Hoping Digital System*) [43].

A atribuição das funções de cada manípulo do comando foi feita de acordo com a Figura 2. 29. Também se atribuiu a um interruptor do comando, a função de ligar e

desligar os motores do *quadcopter*. Na Tabela 5. 1 estão identificadas as funções de cada canal do sistema de controlo remoto implementado.

Tabela 5. 1 - Funções de cada canal do sistema rádio do *quadcopter*

Canal	Função
CH1	<i>Roll</i>
CH2	<i>Pitch</i>
CH3	Acelerador
CH4	<i>Yaw</i>
CH8	Ligar e Desligar motores

Em cada canal tem-se um sinal periódico quadrático com um período de aproximadamente 20 ms, em que a largura do impulso, compreendida entre os 1 e 2 ms, varia consoante a posição do respetivo manípulo ou interruptor. Na Figura 5. 1 está representado o sinal do canal associado ao botão de ligar e desligar os motores.

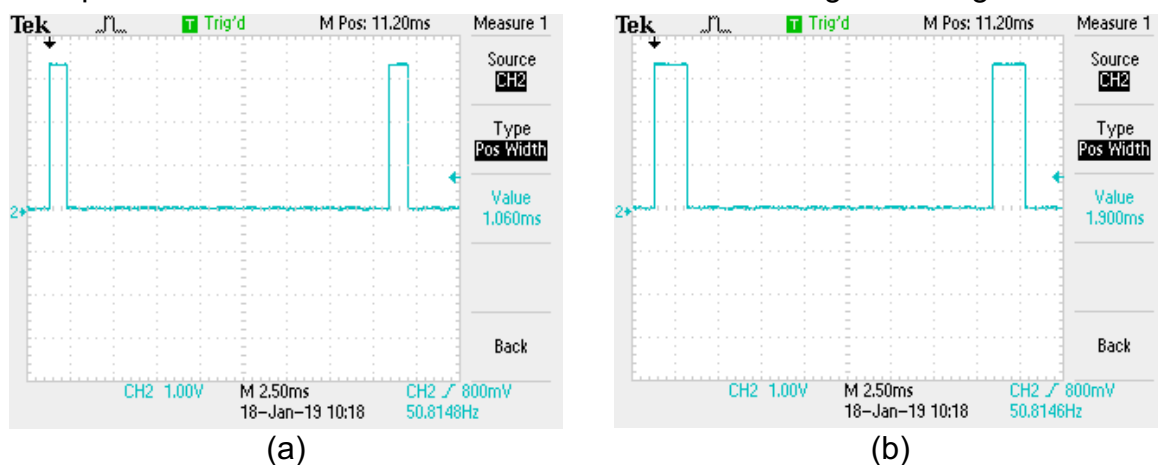


Figura 5. 1 - Sinal do canal 8 do controlo remoto: a) interruptor na posição de desligar os motores; interruptor na posição de ligar os motores.

Na Figura 5. 1 visualiza-se a diferença do sinal do canal 8 para as duas posições do interruptor. Quando este está na posição de desligar os motores, o impulso quadrático tem uma largura de 1,060 ms e quando está na posição de ligar os motores, o impulso quadrático tem uma largura de 1,900 ms.

Cada canal do recetor rádio está ligado a um pino do Arduino, e a cada um dos canais associou-se uma interrupção para se poder determinar a largura do impulso. As interrupções são acionadas tanto na *rising edge* como na *falling edge* da onda quadrática. O fluxograma da rotina da interrupção implementada em Arduino está representado na Figura 5. 2.

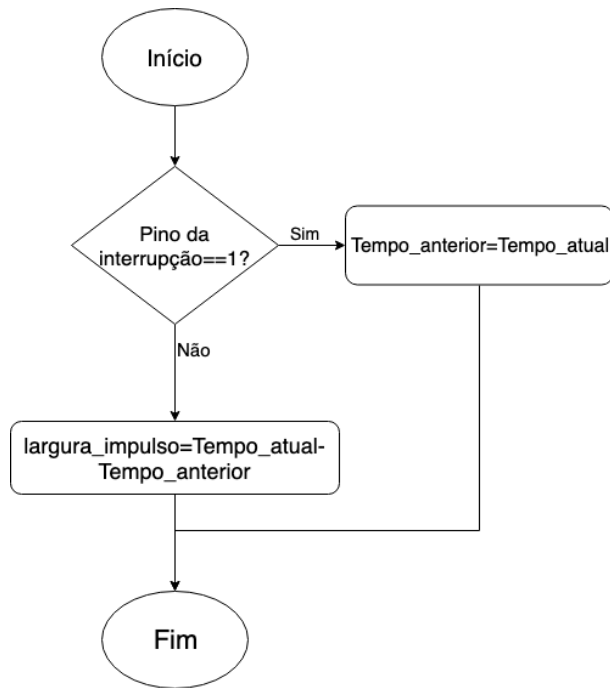


Figura 5. 2 - Rotina da interrupção dos canais rádio.

Quando o telecomando é desligado, os canais do recetor deixam de ter sinal e por isso implementou-se uma funcionalidade no *quadcopter* para desligar os motores quando se perde a ligação com o telecomando, Figura 5. 3.

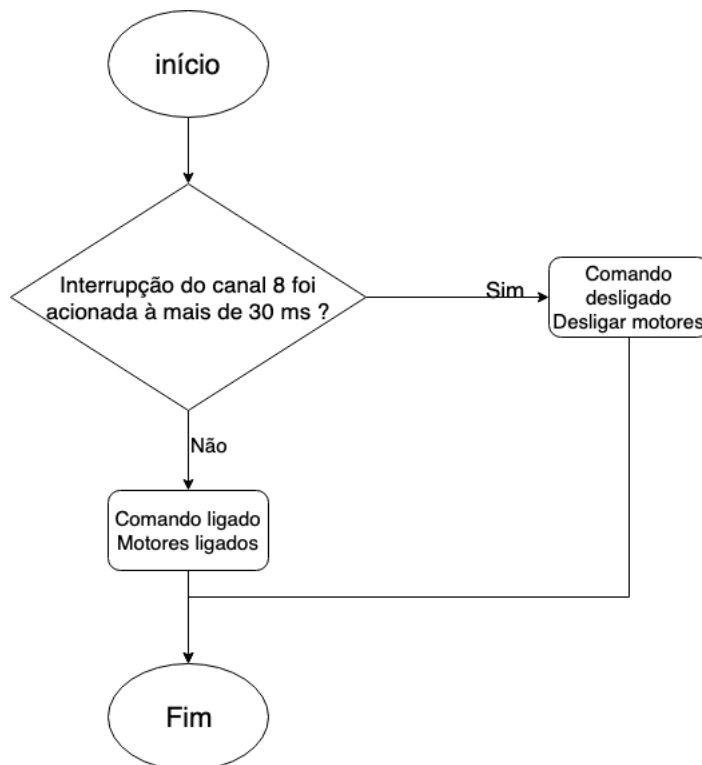


Figura 5. 3 - Rotina do programa principal consoante o estado do telecomando.

Para se implementar a funcionalidade da Figura 5. 3, criou-se uma função que a cada iteração do ciclo principal do programa verifica quanto tempo passou desde que a

interrupção do canal 8 foi ativada pela última vez. Quando a interrupção tiver sido ativada pela última vez à mais de 30 ms, assume-se que se perdeu a ligação rádio.

5.3 Sensores

Para estimar a orientação do *quadcopter* utilizou-se o sensor MPU-6050 da InvenSense, Figura 2. 16. Num único chip este sensor tem um acelerómetro e um giroscópio de três eixos e um termómetro. A interface entre o Arduino e este sensor é realizada através de comunicação I²C. A ligação entre o microcontrolador e o sensor foi estabelecida de acordo com a Figura 5. 4.

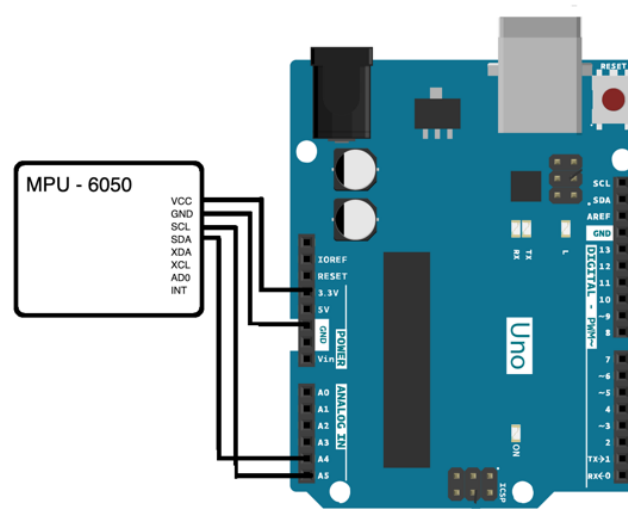


Figura 5. 4 - Ligação do MPU-6050 ao Arduino Uno.

No início do programa principal foi necessário configurar o sensor. Na Figura 5. 5 apresenta-se a configuração inicial do MPU-6050.

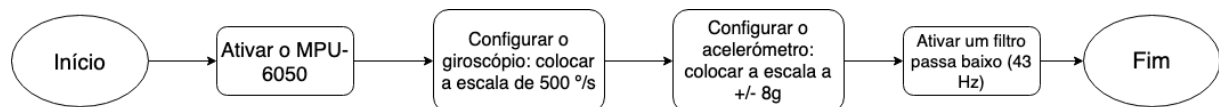


Figura 5. 5 - Configuração inicial do MPU-6050.

A configuração do MPU-6050 é efetuada através da comunicação I²C entre o microcontrolador e o sensor. Para se ativar o sensor, iniciou-se a comunicação I²C para o endereço do sensor, 0x68 em hexadecimal. Depois no registo “PWR_MGMT_1”, com o endereço 0x6B, colocou-se todos os 8 *bits* a 0 terminando de seguida a transmissão. Para se colocar a escala de 500 °/s no giroscópio, começou-se de novo a transmissão com o endereço do sensor, e no registo “GYRO_CONFIG”, no endereço 0x1B, escreveu-se a palavra 00001000 terminando depois a comunicação. Para se configurar a escala de +/- 8 g no acelerómetro do MPU-6050 iniciou-se novamente a comunicação I²C para o endereço do sensor e no

registo “ACCEL_CONFIG”, com endereço 0x1A, escreveu-se a palavra 00010000 e depois terminou-se a ligação. Por fim para se ativar o filtro passa-baixo de 43 Hz no sensor estabeleceu-se outra transmissão com o sensor e no registo “CONFIG”, no endereço 0x1A escreveu-se a palavra 00000011 [44].

Para se obter informação do giroscópio, do acelerómetro e do termómetro, inicia-se a comunicação I²C com o sensor e envia-se na transmissão o *byte* 0x3B. Depois o sensor envia para o microcontrolador 14 *bytes* de informação. A informação enviada pelo sensor corresponde às 7 variáveis medidas (aceleração em x, y e z, temperatura e derivada dos ângulos de Euler), ou seja, 2 *bytes* para cada variável [44].

Depois de configurado o sensor, calibrou-se o giroscópio. Para se calibrar o giroscópio, mediram-se 2000 amostras e calculou-se a média dessas amostras para cada eixo. Depois nas variáveis onde se armazenam os valores atuais do giroscópio subtraíram-se a esses valores, as médias calculadas anteriormente.

5.4 Reguladores de velocidade

A velocidade dos motores é controlada pelos reguladores de velocidade. O sentido de rotação dos motores não é imposto eletronicamente pelos reguladores, mas sim fisicamente através das ligações entre os reguladores e os motores. A troca de uma fase na ligação entre o controlador de velocidade e o motor permite inverter o sentido de rotação. Na Figura 5. 6 apresenta-se o esquemático das ligações efetuadas entre os reguladores e os motores.

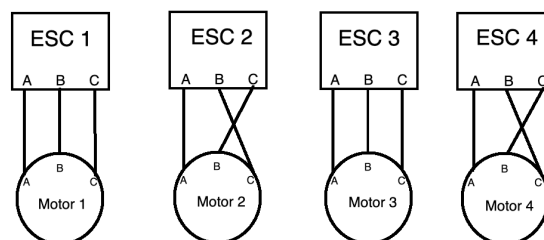


Figura 5. 6 - Ligações dos reguladores de velocidade aos motores.

Com a troca de fases nos motores 2 e 4, inverteu-se o sentido de rotação destes dois motores, ficando o motor 1 e 3 a rodar no sentido dos ponteiros do relógio e os motores 2 e 4 a rodarem no sentido oposto.

Para se utilizarem os controladores de velocidade com o Arduino, e sendo todo o sistema alimentado por uma bateria, ligou-se a alimentação dos controladores à bateria e alimentou-se o Arduino através dos reguladores de tensão dos controladores de velocidade. Por último ligaram-se os pinos de PWM dos controladores 1, 2, 3 e 4 aos pinos de saída 4, 5, 6 e 7 do Arduino. O esquemático destas ligações está apresentado na Figura 5. 7.

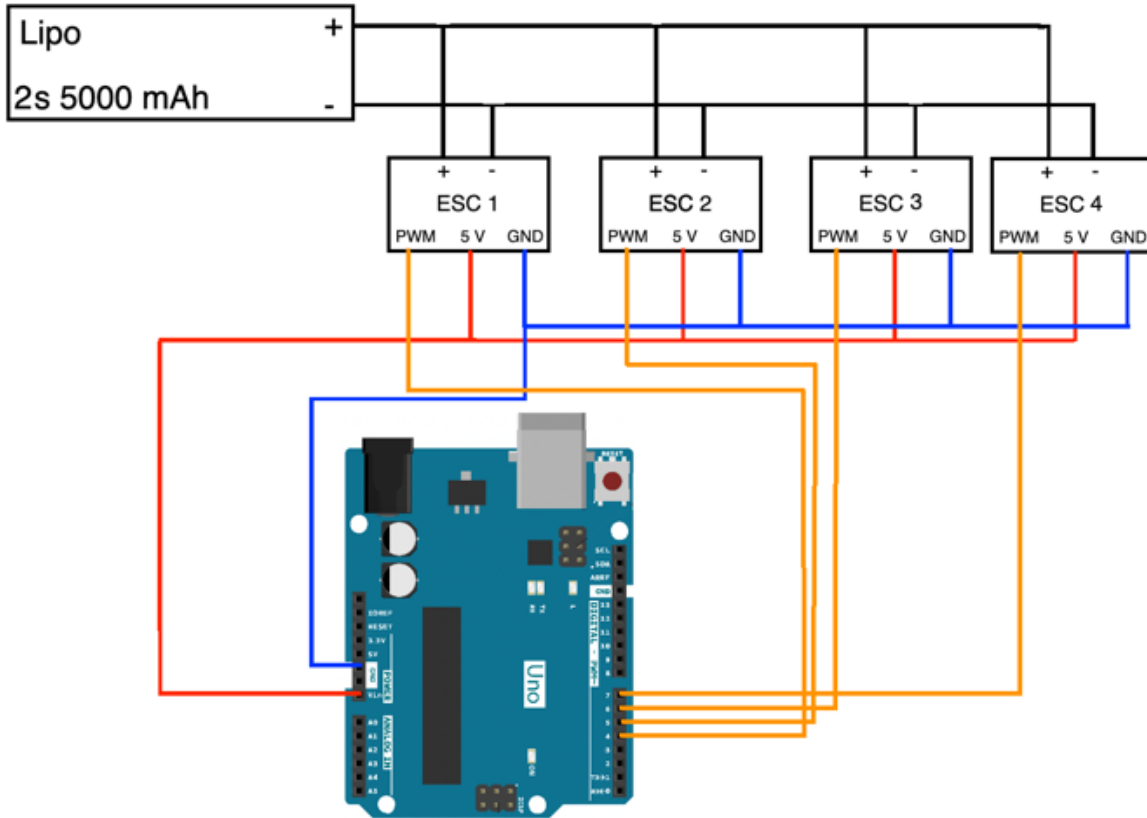


Figura 5. 7 - Ligações estabelecidas entre os controladores de velocidade, microcontrolador e bateria. Os controladores de velocidade utilizados neste projeto, antes de controlarem os motores têm de ser calibrados. O processo de calibração é iniciado no instante em que a alimentação é ligada, Figura 5. 8.

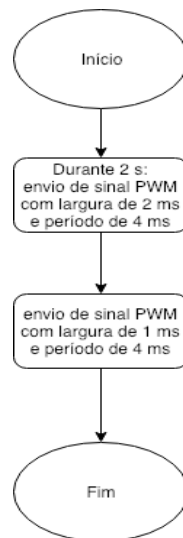


Figura 5. 8 - Fluxograma da calibração dos reguladores de velocidade.

Durante o processo de calibração são estabelecidos os sinais PWM que correspondem às velocidades máxima e mínima que os reguladores conseguem produzir. A velocidade máxima dá-se quando o sinal PWM enviado tem uma largura de 2 ms, e a velocidade mínima (0 rpm) quando a largura é de 1 ms. A velocidade dos

motores é regulada variando a largura do sinal PWM entre 1 e 2 ms. Desde que o período do sinal PWM seja superior a 2 ms os reguladores de velocidade funcionam independentemente do valor escolhido, mas como no programa do controlo de voo a frequência de amostragem é de 250 Hz, definiu-se então como 4 ms o período do sinal de PWM. O período do sinal PWM permite ajustar o tempo de reação dos motores, quanto menor for esse período mais rápido os motores reagem (dentro dos limites que as características eletro-mecânicas dos motores impõem). A Figura 5. 9 apresenta a forma do sinal PWM para fazer parar o motor.

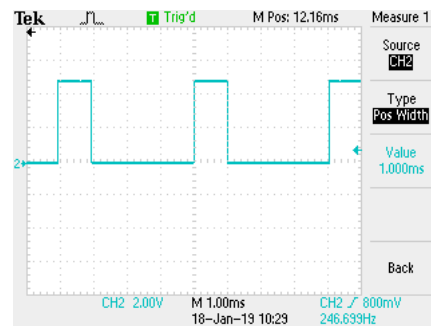


Figura 5. 9 - Sinal PWM enviado para parar o motor.

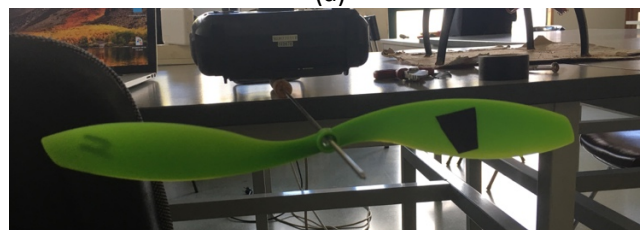
Quando o motor é obstruído, ocorre um curto-circuito e o regulador de velocidade deteta o aumento de corrente. Para eliminar o curto-circuito, o regulador de velocidade reinicializa.

5.5 Estrutura do *quadcopter*

Inicialmente, optou-se por utilizar a estrutura do modelo Arducopter da Figura 2.1 em conjunto com hélices de 12x4,5 polegadas. Observou-se que estas hélices não estavam equilibradas e com fita cola equilibraram-se as mesmas, Figura 5. 10.



(a)



(b)

Figura 5. 10 - Equilíbrio da hélice: a) hélice desequilibrada; b) hélice equilibrada.

Quando as hélices não estão equilibradas, ao rodarem produzem muitas oscilações que introduzem ruído no giroscópio. Ao equilibrar as hélices as oscilações desaparecem.

Apesar de se terem equilibrado as hélices não se conseguiu colocar o *drone* a voar com a estrutura do Arducopter. Como tal substitui-se a estrutura da aeronave por outra mais rígida, Figura 5. 11.



Figura 5. 11 - Estrutura S500 [45].

Como a nova estrutura é mais pequena que a anterior, teve que se usar hélices com um diâmetro menor. Optou-se por utilizar hélices de 10x4,5 polegadas já equilibradas. Com esta nova estrutura conseguiu-se colocar o *quadcopter* a voar. A maior rigidez da nova estrutura trouxe maior estabilidade ao voo.

5.6 Verificação do controlo de voo

Com todos os subsistemas a funcionarem, procedeu-se à montagem da aeronave. Na Figura 5. 12 tem-se o diagrama do sentido de rotação de cada um dos motores e o sentido dos eixos x e y do *quadcopter*.

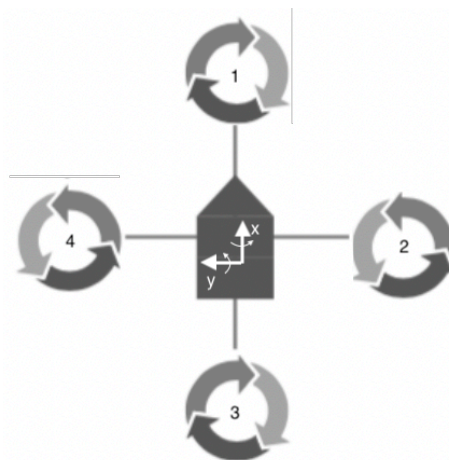


Figura 5. 12 - Diagrama dos sentidos de rotação dos motores e do referencial do *quadcopter*.

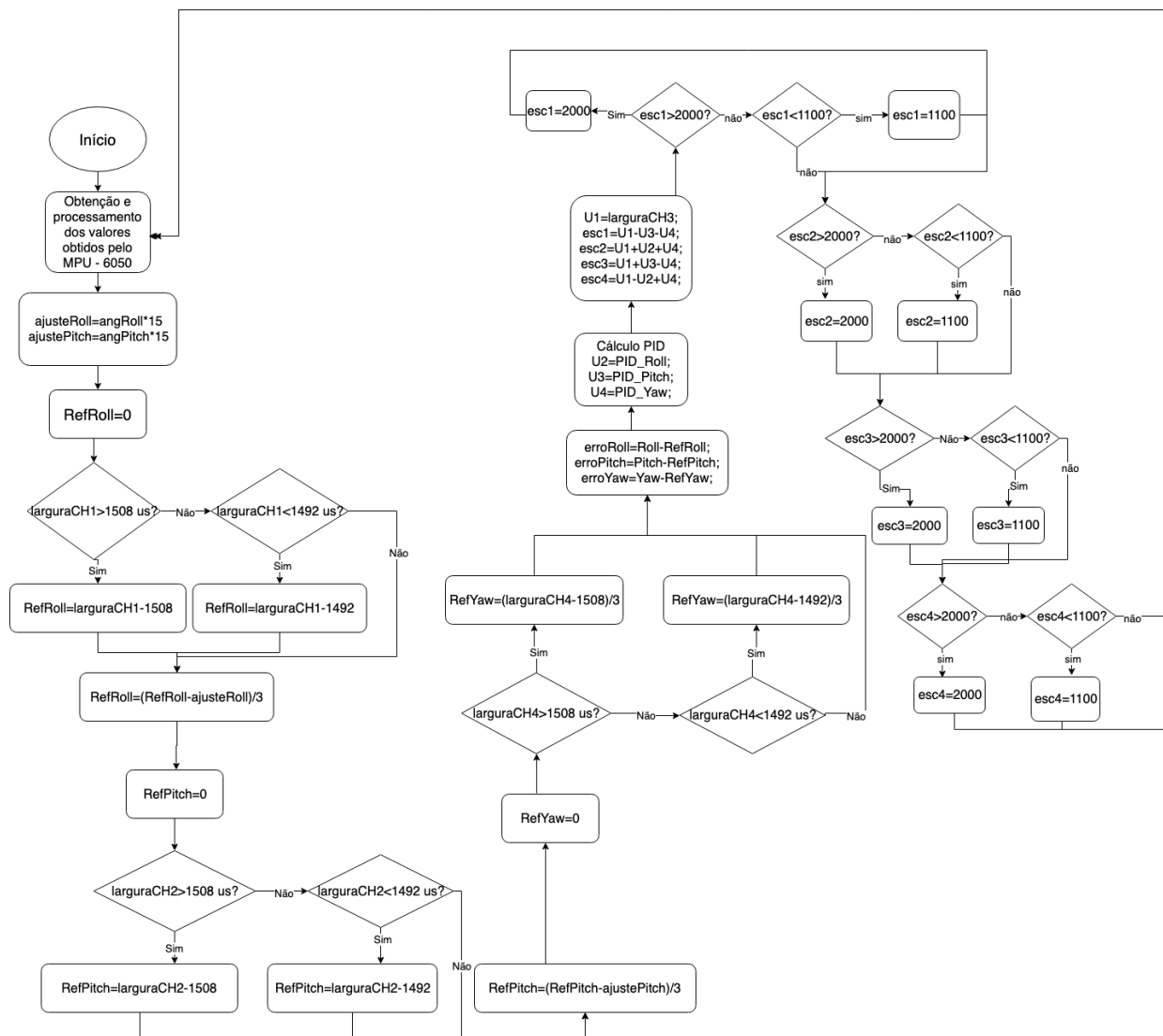


Figura 5. 14 - Fluxograma do controlador de orientação implementado experimentalmente.

Os dados obtidos pelo sensor são processados de forma a terem-se os valores do giroscópio em °/s e de se conseguirem estimar os ângulos a partir dos valores do acelerômetro. No controlador implementado para os ângulos ϕ e θ necessita-se do valor destes e da sua derivada, enquanto que para o ângulo ψ apenas se utiliza a sua derivada. No fluxograma da Figura 5. 14, as variáveis “angRoll” e “angPitch” dizem respeito ao valor atual de ϕ e θ , e são obtidas através de um filtro complementar que engloba os valores do giroscópio com os do acelerômetro. Como para o ângulo ψ não é possível utilizar o acelerômetro para determinar o seu valor, não se estimou no controlador o seu valor.

As variáveis “ajusteRoll” e “ajustePitch” são obtidas através dos valores atuais dos ângulos ϕ e θ . Estas variáveis servem para fazer um ajuste ao valor de referência imposto pelos sinais do telecomando.

A largura dos canais rádio determinam o sinal de referência para o controlo. A largura dos impulsos dos canais varia entre 1000 μs e 2000 μs . Para os canais 1,2 e 4 que correspondem respetivamente aos movimentos de *roll*, *pitch* e *yaw*, quando a largura está entre 1492 e 1508 μs , na posição central dos manípulos, isto significa que o *quadcopter* não se deve movimentar, ou seja, a referência é 0. Quando a largura dos impulsos é superior a 1508 μs , ou inferior a 1492 μs , significa que se pretende alterar a orientação da aeronave. Consoante o canal que se alterar, irá ocorrer movimento de *roll*, *pitch* ou *yaw*. Na parte do fluxograma em que consoante a largura dos impulsos, se subtrai por 1508 ou por 1492 e depois se divide por 3, está-se a mapear o sinal de comando para que varie entre -164 %/s a 164 %/s.

A correção aos valores de referência “RefRoll” e “RefPitch” que se faz através da subtração do “ajusteRoll” e “ajustePitch” deve-se ao facto de se estar a comandar a aeronave através da derivada dos ângulos. Quando se move o manípulo relativo ao *roll* ou *pitch*, o *quadcopter* irá começar a inclinar-se. Quando o ângulo de inclinação atinge um valor que faz anular o sinal do comando, ou seja:

$$\text{larguraCH1} - 1508 = \text{ajusteRoll} \text{ ou } \text{larguraCH1} - 1492 = \text{ajusteRoll}; \quad (5.1)$$

$$\text{larguraCH2} - 1508 = \text{ajustePitch} \text{ ou } \text{larguraCH2} - 1492 = \text{ajustePitch}, \quad (5.2)$$

o *quadcopter* deixa de se inclinar e fica com um ângulo de inclinação constante.

Depois de definidos os valores de referência, calculam-se os erros e com os compensadores PID de cada ângulo obtêm-se os sinais de controlo U_2 , U_3 e U_4 . O sinal U_1 corresponde ao canal rádio do acelerador. Na prática quem realiza o controlo de altura é o próprio utilizador no comando com o manípulo do acelerador. Os valores dos compensadores PID utilizados na prática estão na Tabela 5. 2 e foram baseados em [46].

Tabela 5. 2 - Valores dos compensadores PID utilizados na prática

Compensador	Controlo		
	Roll	Pitch	Yaw
Proporcional (K_p)	1,3	1,3	4
Integral (K_i)	0,04	0,04	0,02
Derivativo (K_d)	18	18	0

Após a determinação de todos os sinais de controlo, calculavam-se as velocidades de cada motor. Como se optou por uma estrutura em forma de + para o *quadcopter*, em que o eixo do x está alinhado com os motores 1 e 3, e o eixo y alinhado com o motor 2 e 4, o sinal de controlo U_2 aparecia nas variáveis “esc2” e “esc4” e o sinal U_3 aparecia nas variáveis “esc1” e “esc3”. Por fim, colocaram-se limites à velocidade dos motores. O limite superior de 2000 deve-se ao facto de os reguladores de velocidade receberem

no máximo um sinal PWM com uma largura de 2 ms. O limite inferior de 1100 foi estabelecido para evitar que os motores chegassem a parar durante o voo.

5.6.2 Programa completo

Na Figura 5. 15 apresenta-se o fluxograma que explica o funcionamento do programa completo implementado no *quadcopter*.

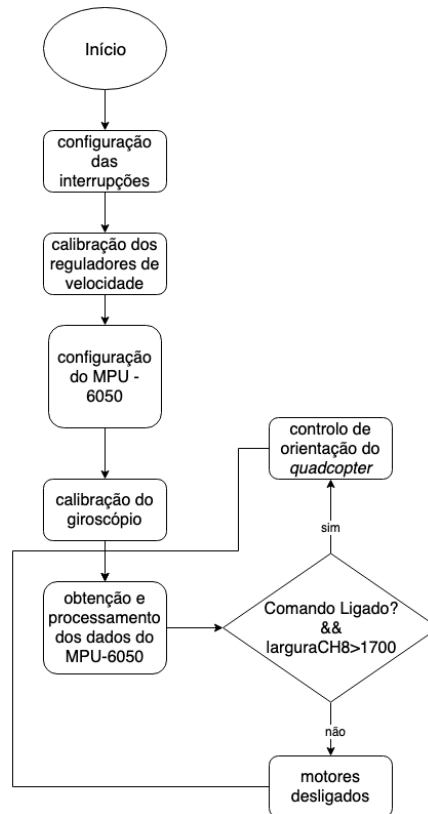


Figura 5. 15 - Fluxograma do programa completo para o controle do *quadcopter*.

O programa inicia-se assim que a alimentação do *quadcopter* é ligada e só termina quando a alimentação é desligada. Para que o controlador de voo comece a funcionar é necessário que o comando esteja ligado e que o botão de ligar e desligar os motores também esteja ligado. Para verificar o estado desse botão no programa, verifica-se a largura dos impulsos do canal 8 do recetor rádio. Se a largura for superior a 1700 μ s o botão está ligado, caso contrário está desligado e os motores não rodam.

Com o programa final carregado para o microcontrolador do quadcopter, testou-se o voo da aeronave. Com os testes realizados conseguiu-se descolar, realizar os movimentos de *roll*, *pitch* e *yaw* e aterrar. Na Figura 5. 16 apresenta-se uma imagem capturada durante um teste de voo realizado com o *quadcopter* criado neste projeto.



Figura 5. 16 - Teste experimental do voo do *quadcopter*.

Com o controlador de voo desenvolvido conseguiu-se ter voos estáveis. Apenas nas situações em que o giroscópio não foi calibrado corretamente, quando o *quadcopter* era iniciado sobre uma superfície com inclinação, o *drone* não se conseguia manter imóvel no ar durante o voo, porque com o giroscópio descalibrado não era possível controlar corretamente a sua orientação.

Na verificação experimental, devido a erros de medição do giroscópio e do acelerómetro, e à capacidade de processamento disponível, apenas se aplicou a técnica de controlo PID. Tal como na simulação, a nível experimental conseguiu-se controlar a orientação do *quadcopter*. Ao contrário da simulação, na verificação experimental, como nenhum sensor permitia monitorizar a altura do *quadcopter*, o controlo de altura foi feito pelo piloto do *drone*, através do ajuste manual do acelerador.

6 Conclusões

Neste capítulo apresentam-se as conclusões retiradas deste projeto e apresentam-se sugestões para trabalhos futuros.

6.1 Conclusões gerais

Neste trabalho começou-se por apresentar o conceito de *quadcopter* e verificou-se em que áreas este tipo de aeronaves são mais utilizadas. As áreas da fotografia, militar, vigilância, de busca e salvamento e de transportes recorrem ao uso de *quadcopters* devido à capacidade de descolagem e aterragem na vertical, como também à capacidade de se conseguirem manter imóveis no ar.

Apresentaram-se os dois tipos de formas mais comuns que os *quadcopter* podem tomar, em $+$ e em x . A forma em $+$ tem a vantagem do controlo de voo ser menos complexo, mas ao mesmo tempo tem a desvantagem, em certas áreas como a da fotografia, porque o braço, o motor e a hélice do *quadcopter* são capturados no plano da câmara. A forma em x resolve essa desvantagem, mas torna o controlo mais complexo.

Apresentou-se e explicaram-se os movimentos de rotação que o *quadcopter* consegue efetuar e de que forma este se movimenta no espaço. São três os movimentos de rotação: *roll*, *pitch* e *yaw*. No movimento *roll*, o *quadcopter* roda em torno do seu eixo x e é este movimento que faz com que o *drone* se mova para a direita ou esquerda. No movimento *pitch*, o *quadcopter* roda em torno do eixo y e é este movimento que permite o *quadcopter* se deslocar para a frente ou para trás. O movimento *yaw*, é o movimento em que o *quadcopter* gira em torno do eixo do z , ou seja, roda em torno de si próprio.

Explicaram-se os ângulos de Euler, que são responsáveis por descrever a orientação de um corpo rígido num espaço tridimensional e apresentou-se a matriz de rotação responsável por transferir grandezas entre o referencial fixo da terra e o referencial da aeronave. Esta matriz permite trabalhar com grandezas que estão representadas em referenciais distintos, como a força gravítica, representada no referencial fixo, e a força de propulsão dos motores, representada no referencial do *quadcopter*.

Apresentou-se um resumo de vários trabalhos de investigação desenvolvidos que abordam diferentes técnicas de controlo de orientação e de posição do *quadcopter*. Também se apresentaram vários *quadcopters* comerciais com diferentes finalidades, desde o uso recreativo até ao transporte de mercadorias e passageiros.

Apresentou-se a arquitetura típica de um *quadcopter* e explicou-se a finalidade de cada um dos subsistemas constituintes. Estudou-se o funcionamento dos sensores utilizados para a estimação da orientação, principalmente do acelerómetro e do

giroscópio e verificou-se as respectivas características. O giroscópio é um sensor que mede a variação dos ângulos, logo a orientação é obtida através da integração das medições deste. O erro da estimativa da orientação através do giroscópio tem origem na acumulação de erro na integração das medições. O acelerómetro mede acelerações, mas é possível estimar a orientação dos ângulos ϕ e θ , quando o acelerómetro apenas está sujeito à aceleração gravítica. O uso de um filtro complementar com os ângulos do giroscópio e do acelerómetro permite ter uma melhor estimativa dos ângulos de Euler.

Os dois tipos de motores utilizados em *quadcopters*, BLDC e DC, foram estudados e indicaram-se as diferenças entre ambos. As principais diferenças são: o motor BLDC é um motor trifásico de corrente alternada, enquanto que o motor DC é um motor de corrente contínua, e o motor BLDC necessita de menos manutenção que o motor DC. Também se apresentaram técnicas de controlo de velocidade dos motores BLDC, sendo as mais utilizadas, as técnicas que não recorrem ao sensor de posição do rotor.

Estudaram-se as equações que descrevem a dinâmica do *quadcopter* e as técnicas de controlo de orientação e posição. Verificou-se a diferença na matriz dos momentos do *quadcopter* para uma estrutura em forma de $+$ e em forma de \times , mostrando assim a razão pela qual o controlo de orientação depende da forma da estrutura. Enquanto que na forma em $+$, o movimento de *roll* é controlado pelos motores 2 e 4 e o movimento de *pitch* pelos motores 1 e 3, na estrutura em \times esses movimentos são controlados pelos quatro motores em simultâneo.

Os modelos da dinâmica do *quadcopter* e as técnicas de controlo apresentadas no foram simulados, e conseguiu-se com todas as técnicas controlar a orientação e a altura do *drone*.

No controlo de altura verificou-se que com o controlador PID se obtinha uma resposta com algum tremor na subida e com *overshoot*. Com a técnica de *backstepping*, o tremor desaparecia e teve-se uma resposta mais rápida, mas com uma percentagem de *overshoot* maior. No controlo de orientação, nos ângulos ϕ e θ , verificou-se que as únicas respostas com 0 % de *overshoot* foram as obtidas com a técnica Fuzzy-PID porque nessa técnica à medida que o erro vai diminuindo, são atribuídos níveis linguísticos ao erro e à sua derivada para que o compensador PID tome valores que tornem a resposta mais suave. Das três técnicas testadas no controlo de orientação, o *backstepping* é o método com o melhor desempenho de uma forma geral, tendo em conta à percentagem de *overshoot* e ao tempo de resposta.

Na parte experimental verificou-se o que ao se substituir a estrutura do *drone* por outra mais rígida, conseguiu-se trazer mais estabilidade ao voo. Também se verificou que

quando as hélices não estão equilibradas, ao rodarem produzem oscilações que adicionam ruído às medições do giroscópio.

Experimentalmente verificou-se também que para aumentar a velocidade de reação dos motores, teve-se que aumentar a frequência do sinal de PWM enviado para os reguladores. No controlo de voo, a frequência deste sinal tem de ser igual ou inferior à frequência de amostragem do controlador.

Devido a erros de medição e à reduzida capacidade do processador, a nível experimental optou-se por controlar através da técnica PID e das derivadas dos ângulos, a orientação do *quadcopter*. Tal como na simulação, conseguiu-se efetuar o controlo da orientação.

6.2 Trabalhos futuros

Como trabalho futuro propõe-se utilizar um microcontrolador com mais capacidade de cálculo para implementar as técnicas Fuzzy-PID e *backstepping*.

Além disso, propõe-se explorar outras técnicas de controlo como LQR e H infinito para o controlo de orientação e altura.

Estudar variantes do *quadcopter*, como um *hexacopter* (6 motores) ou um *octocopter* (8 motores) para aumentar a capacidade de carga do *drone* e para implementar mecanismos de segurança em caso de falha de um motor.

Introduzir um sensor que permita monitorizar o nível da bateria, para que o *quadcopter* possa aterrar em segurança antes que a energia da bateria deixe de ser suficiente para manter o *drone* a voar.

Utilizar outro sistema rádio que permita ao *quadcopter* enviar informações importantes para o utilizador como imagens em tempo real, altura e tempo de voo restante.

Por último, introduzir GPS e sensores de obstáculos no *quadcopter* para poder implementar, na prática, o controlo de posição e realizar voos autónomos.

7 Referências

- [1] H. C. T. E. Fernando, A. T. A. De Silva, M. D. C. De Zoysa, K. A. D. C. Dilshan, e S. R. Munasinghe, «Modelling, simulation and implementation of a quadrotor UAV», *2013 IEEE 8th Int. Conf. Ind. Inf. Syst. ICIS 2013 - Conf. Proc.*, pp. 207–212, 2013.
- [2] M. Mardan, «Attitude and Position Control of Quadrotors : Design , Implementation and Experimental Evaluation», University of Manitoba, 2016.
- [3] N. Zimmerman, «Flight control and hardware design of multi- rotor systems», Marquette University, 2016.
- [4] F. Sabatino, «Quadrotor control : modeling , nonlinear control design , and simulation», KTH Electrical Engineering, 2015.
- [5] K. M. Street e P. Alegre, «Mathematical Modelling and Simulation of Quadrotor», *J. Electr. Electron. Eng.*, vol. 12, n. 6, pp. 11–18, 2017.
- [6] V. Mistler, A. Benallegue, e N. K. M. Sirdi, «Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback», *IEEE Int. Work. Robot Hum. Interact. Commun.*, pp. 586–593, 2001.
- [7] T. Madani e A. Benallegue, «Control of a Quadrotor Mini-Helicopter via Full State Backstepping Technique», *Proc. 45th IEEE Conf. Decis. Control*, pp. 1515–1520, 2006.
- [8] W. Dong, G. Gu, X. Zhu, e H. Ding, «Modeling and Control of a Quadrotor UAV with Aerodynamic Concepts», *World Acad. Sci. Eng. Technol.*, vol. 7, n. 5, pp. 377–382, 2013.
- [9] D. C. Tosun, Y. Işık, e H. Korul, «LQR control of a quadrotor helicopter», *New Dev. Pure Appl. Math. LQR*, pp. 247–252, 2015.
- [10] E. Kuantama, T. Vesselenyi, S. Dzitac, e R. Tarca, «PID and Fuzzy-PID control model for quadcopter attitude with disturbance parameter», *Int. J. Comput. Commun. Control*, vol. 12, n. 4, pp. 519–532, 2017.
- [11] F. Justin, «The History Of Drones (Drone History Timeline From 1849 To 2018)», 2018. [Em linha]. Disponível em: <https://www.dronethusiast.com/history-of-drones/>. [Acedido: 27-Set-2018].
- [12] «Amazon.com: Parrot AR.Drone Quadcopter Controlled by iPod touch, iPhone, iPad, and Android Devices (Orange/Blue) (Discontinued by Manufacturer): Camera & Photo». [Em linha]. Disponível em: https://www.amazon.com/Parrot-Quadcopter-Controlled-Discontinued-Manufacturer/dp/B003ZVSHB0/ref=cm_cr_arp_d_product_top?ie=UTF8. [Acedido: 27-Set-2018].
- [13] IFIXIT, «Parrot AR.Drone Teardown - iFixit», 2010. [Em linha]. Disponível em: <https://www.ifixit.com/Teardown/Parrot+AR.Drone+Teardown/3984>. [Acedido: 27-Set-2018].
- [14] Amazon, «Amazon.com: Prime Air», 2018. [Em linha]. Disponível em: <https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011>. [Acedido: 27-Set-2018].
- [15] Z. Dukowitz, «What’s the Most Popular Drone in the U.S.? New FAA Data Answers This Question and More - UAV Coach», 2017. [Em linha]. Disponível em: <https://uavcoach.com/faa-drone-data/>. [Acedido: 27-Set-2018].

- [16] DJI, «DJI Phantom 4 – Specs, FAQ, Tutorials and Downloads», 2018. [Em linha]. Disponível em: <https://www.dji.com/phantom-4/info>. [Acedido: 27-Set-2018].
- [17] DJI, «Phantom 4 - DJI's smartest flying camera ever.» [Em linha]. Disponível em: <https://www.dji.com/phantom-4>. [Acedido: 27-Set-2018].
- [18] M. Margaritoff, «Watch the Ehang 184 Passenger Drone Successful Taxi Someone Around - The Drive», 2018. [Em linha]. Disponível em: <http://www.thedrive.com/aerial/18261/watch-the-ehang-184-passenger-drone-successfully-taxi-someone-around>. [Acedido: 27-Set-2018].
- [19] M. P., «EHang 184 is a manned UAV you will never get to fly», 2016. [Em linha]. Disponível em: <https://www.dronethusiast.com/ehang-184-is-a-manned-uav-you-will-never-get-to-fly/>. [Acedido: 27-Set-2018].
- [20] Arduino, «Arduino Playground - MPU-6050». [Em linha]. Disponível em: <https://playground.arduino.cc/Main/MPU-6050>. [Acedido: 01-Out-2018].
- [21] D. Patton, «Simple Explanation of How Gyroscopes Work | Sciencing», 2017. [Em linha]. Disponível em: <https://sciencing.com/simple-explanation-gyroscopes-work-6133912.html>. [Acedido: 02-Out-2018].
- [22] P. Jan, «Reading a IMU Without Kalman: The Complementary Filter | Pieter-Jan.com», 2013. [Em linha]. Disponível em: <http://www.pieter-jan.com/node/11>. [Acedido: 02-Out-2018].
- [23] Motion Control Online, «Brushed DC Motors Vs. Brushless DC Motors | Motion Control Blog», 2017. [Em linha]. Disponível em: <https://www.motioncontrolonline.org/blog-article.cfm/Brushed-DC-Motors-Vs-Brushless-DC-Motors/24>. [Acedido: 03-Out-2018].
- [24] M. Dirjish, «What's The Difference Between Brush DC And Brushless DC Motors?», 2012. [Em linha]. Disponível em: <https://www.electronicdesign.com/electromechanical/what-s-difference-between-brush-dc-and-brushless-dc-motors>. [Acedido: 03-Out-2018].
- [25] I. Janpan, R. Chaisricharoen, e P. Boonyanant, «Control of the brushless DC motor in combine mode», *Procedia Eng.*, vol. 32, n. December 2012, pp. 279–285, 2012.
- [26] S. Baldursson, «BLDC Motor Modelling and Control – A Matlab / Simulink Implementation», Institutionen för Energi och Miljö, 2005.
- [27] J. Shao, «Direct Back EMF Detection Method for Sensorless Brushless DC (BLDC) Motor Drives», Virginia Polytechnic Institute, 2003.
- [28] I. Viveiros, «Controlo de motores de um drone», Universidade da Madeira, 2018.
- [29] O. Liang, «How To Choose Radio Transmitter & Receiver for Racing Drones and Quadcopter - Oscar Liang», 2013. [Em linha]. Disponível em: <https://oscarliang.com/choose-rc-transmitter-quadcopter/>. [Acedido: 10-Out-2018].
- [30] «Regulamento 1093/2016, 2016-12-14 - DRE». [Em linha]. Disponível em: https://dre.pt/home/-/dre/105367104/details/maximized?serie=II&print_preview=print-preview&parte_filter=33&day=2016-12-14&date=2016-12-01&dreid=105362832. [Acedido: 27-Nov-2018].
- [31] Diário da República, «Regulamento n.º 1093/2016», pp. 20–23, 2016.
- [32] H. talla M. N. ElKholly, «Dynamic Modeling and Control of a Quadrotor Using Linear and Nonlinear Approaches», *Master Thesis, Am. Univ. Cairo*, pp. 1–143, 2014.
- [33] «Newton-Euler Equations From Dynamics of Systems of Particles to Rigid Body Dynamics», *University of Pennsylvania*. [Em linha]. Disponível em:

- https://alliance.seas.upenn.edu/~meam535/cgi-bin/pmwiki/uploads/Main/Newton_Euler_11.pdf. [Acedido: 15-Out-2018].
- [34] R. C. Dorf e N. K. Sinha, *Modern Control Systems*, 12.^a ed., vol. 11, n. 8. New Jersey: Pearson Education, 2011.
- [35] D. Mcgriffy, *Make: Drones*, 1.^a ed. San Francisco: Maker Media, 2017.
- [36] Mathworks, «Simulink[®] Design Optimization[™] Reference», Natick, 2015.
- [37] A. A. Khan e N. Rapal, «Fuzzy PID Controller: Design, Tuning and Comparison with Conventional PID Controller», *2006 IEEE Int. Conf. Eng. Intell. Syst.*, n. 1, pp. 1–6, 2006.
- [38] S. F. Ahmed, K. Kushsairy, M. I. A. Bakar, D. Hazry, e M. K. Joyo, «Attitude stabilization of Quad-rotor (UAV) system using Fuzzy PID controller (an experimental test)», *2015 2nd Int. Conf. Comput. Technol. Inf. Manag. ICCTIM 2015*, pp. 99–104, 2015.
- [39] K. Passino e S. Yurkovich, *Fuzzy Control*, 1.^a ed. Ohio: Addison Wesley Longman, 1998.
- [40] «Basic Lyapunov theory», 2008. [Em linha]. Disponível em: <https://stanford.edu/class/ee363/lectures/lyap.pdf>. [Acedido: 22-Nov-2018].
- [41] V. K. Tripathi, L. Behera, e N. Verma, «Design of sliding mode and backstepping controllers for a quadcopter», *Proc. 2015 39th Natl. Syst. Conf. NSC 2015*, 2016.
- [42] Arduino, «Arduino Uno Rev3», 2019. [Em linha]. Disponível em: <https://store.arduino.cc/arduino-uno-rev3>. [Acedido: 26-Jan-2019].
- [43] HobbyKing, «Turnigy 9X 9Ch Transmitter w/ Module & iA8 Receiver (Mode 2) (AFHDS 2A system) | HobbyKing». [Em linha]. Disponível em: https://hobbyking.com/pt_pt/turnigy-9x-9ch-transmitter-w-module-ia8-receiver-mode-2-afdhs-2a-system.html?__store=pt_pt. [Acedido: 26-Jan-2019].
- [44] InvenSense Inc., «InvenSense - MPU-6000 and MPU-6050 Product Specification», *Inven. Inc.*, vol. 1, n. 408, pp. 1–57, 2016.
- [45] HobbyKing, «S500 Glass Fiber Quadrotor Quadro 480 milímetros - Integrated PCB Versão». [Em linha]. Disponível em: https://hobbyking.com/pt_pt/s500-glass-fiber-quadcopter-frame-480mm-integrated-pcb-version.html?__store=pt_pt. [Acedido: 28-Jan-2019].
- [46] J. Brokking, «Brokking.net - Project YMFC-AL - The Arduino auto-level quadcopter - Home.», 2016. [Em linha]. Disponível em: http://www.brokking.net/ymfc-al_main.html. [Acedido: 26-Jan-2019].
- [47] R. Nave, «Moment of Inertia». [Em linha]. Disponível em: <http://hyperphysics.phy-astr.gsu.edu/hbase/mi.html>. [Acedido: 22-Out-2018].
- [48] R. Nave, «Parallel Axis Theorem». [Em linha]. Disponível em: <http://hyperphysics.phy-astr.gsu.edu/hbase/parax.html>. [Acedido: 22-Out-2018].
- [49] D. Hartman, K. Landis, M. Mehrer, S. Moreno, e J. Kim, «Quadcopter Dynamic Modelling and Simulation», 2014. [Em linha]. Disponível em: <https://github.com/dch33/Quad-Sim>.
- [50] M. Pedley, «Tilt sensing using three-axis a accelerometer», Austin, 2013.

Anexo A – Momento de inércia do *quadcopter*

O momento de inércia é o equivalente para o movimento de rotação à massa no movimento de translação. Enquanto que no movimento linear se tem um corpo com uma determinada velocidade, aceleração e massa, no movimento rotacional o corpo tem uma determinada velocidade angular, aceleração angular e momento de inércia.

O momento de inércia é definido em relação a um eixo de rotação. O momento de inércia (I) de um ponto de massa é igual à massa (m) a multiplicar pelo quadrado da distância entre esse ponto e o eixo de rotação (r) [47]:

$$I = mr^2. \quad (\text{A. 1})$$

A expressão anterior é válida apenas para pontos de massa. Quando se trata de distribuições contínuas de massa, o momento de inércia passa a ser dado pelo integral dos momentos de inércia de todos os elementos infinitesimais da distribuição contínua de massa [47]:

$$I = \int dI = \int_0^m r^2 dm. \quad (\text{A. 2})$$

Para calcular o momento de inércia do *quadcopter*, é calculado o momento de inércia de cada um dos componentes do *drone*. Cada um dos componentes é simplificado a uma forma geométrica básica com uma densidade constante. Através do teorema de Steiner, Figura A. 1, é determinada a contribuição do momento de inércia dos diferentes componentes em cada um dos eixos do *quadcopter*.

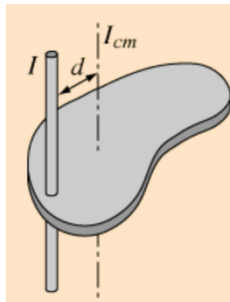


Figura A. 1 - Momento de inércia de dois eixos paralelos [48].

O Teorema de Steiner explica que o momento de inércia de um objeto em relação a um eixo que atravessa o seu centro de massa, é o valor mínimo de momento de inércia que um eixo na direção do centro de massa pode ter. O momento de inércia sobre um eixo paralelo ao eixo que atravessa o centro de massa é dado por [48]:

$$I_{\text{eixo-paralelo}} = I_{CM} + mr^2, \quad (\text{A. 3})$$

em que I_{CM} é o momento de inércia de um objeto sobre o seu próprio eixo, m é a massa do objeto e r a distância entre os dois eixos paralelos.

A.1 Motores

Para calcular o momento de inércia dos motores, estes são considerados cilindros sólidos. Na Figura A. 2 estão representadas as medidas necessárias para calcular o momento de inércia dos motores.

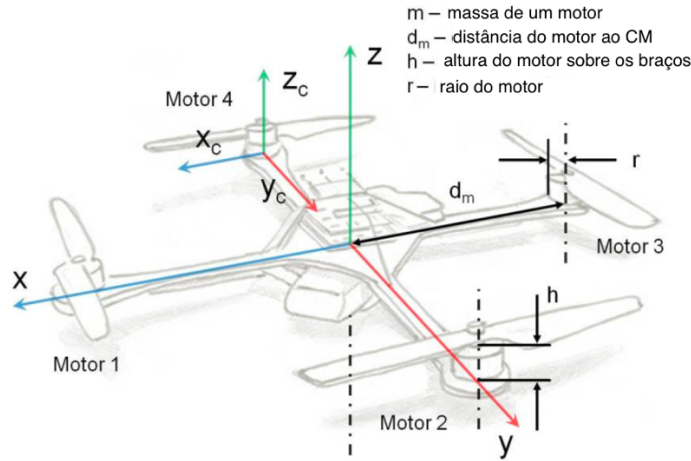


Figura A. 2 – Representação de todas as medidas para o cálculo do momento de inércia dos motores [49].

Além de considerar os motores como cilindros, também é necessário considerar duas rotações sobre dois eixos diferentes, Figura A. 3.



Figura A. 3 - Diferentes rotações dos motores em relação: (a) aos eixos x e y; (b) ao eixo z [49].

Para o cilindro (a) da Figura A. 3, o momento de inércia é dado por [49]:

$$I_{CM} = \frac{1}{4}mr^2 + \frac{1}{3}mh^2. \quad (\text{A. 4})$$

No caso do cilindro (b) da Figura A. 3, o momento de inércia é dado por [49]:

$$I_{CM} = \frac{1}{2}mr^2. \quad (\text{A. 5})$$

Como o *quadcopter* tem uma estrutura simétrica e todos os motores são iguais, o momento de inércia tanto para o eixo x como y é igual [49]:

$$I_{x,M} = I_{y,M} = 2 \left[\frac{1}{4}mr^2 + \frac{1}{3}mh^2 \right] + 2 \left[\frac{1}{4}mr^2 + \frac{1}{3}mh^2 + md_m^2 \right]. \quad (\text{A. 6})$$

Para explicar a expressão (A. 6) tem-se em consideração o momento de inércia sobre o eixo do x, $I_{x,M}$. O primeiro termo da expressão é relativo ao momento de inércia dos

motores 1 e 3. Como estes dois motores coincidem com o eixo do x , o termo mr^2 do Teorema de Steiner não aparece. O segundo termo da expressão (A. 6) diz respeito ao momento de inércia dos motores 2 e 4. Como estes dois motores estão sobre o eixo do y , no momento de inércia destes motores já aparece o termo md_m^2 , devido ao Teorema de Steiner.

Como o momento de inércia em z é igual para todos os motores, o momento de inércia total $I_{z,M}$ é dado pelo somatório dos momentos de cada motor [49]:

$$I_{z,M} = 4 \left[\frac{1}{2} mr^2 + md_m^2 \right]. \quad (\text{A. 7})$$

A expressão do momento $I_{z,M}$ é obtida através da expressão do momento do cilindro que roda em torno de si próprio. Apesar dos motores estarem a rodar sobre si próprios, estes são paralelos ao eixo z do *quadcopter*, logo pelo Teorema de Steiner, surge na expressão (A. 7) o termo md_m^2 .

A.2 ESC

Para calcular o momento de inércia dos controladores de velocidade, estes são considerados como placas finas planas, Figura A. 4.

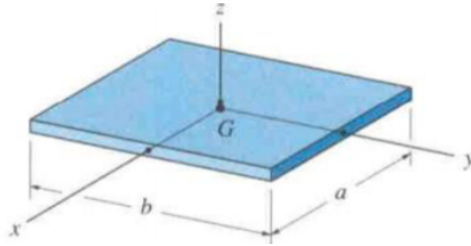


Figura A. 4 - Forma geométrica simplificada do ESC a rodar em torno dos eixos x, y e z [49].

Para uma placa fina, os momentos de inércia para cada um dos eixos são dados por [49]:

$$I_{CM,x} = \frac{1}{12} mb^2; \quad (\text{A. 8})$$

$$I_{CM,y} = \frac{1}{12} ma^2; \quad (\text{A. 9})$$

$$I_{CM,z} = \frac{1}{12} m(a^2 + b^2); \quad (\text{A. 10})$$

em que a é a largura do ESC, b o seu comprimento e m a sua massa.

Devido à simetria do *quadcopter*, os momentos de inércia sobre x e y são iguais. A partir das expressões (A. 8) e (A. 9), os momentos de inércia sobre os eixos x e y devidos aos ESC são dados por [49]:

$$I_{x,S} = I_{y,S} = 2 \left[\frac{1}{12} ma^2 \right] + 2 \left[\frac{1}{12} mb^2 + md_s^2 \right], \quad (\text{A. 11})$$

sendo d_s a distância dos ESC ao centro do *quadcopter*. Considerando o momento $I_{x,S}$, o primeiro termo da expressão anterior é relativo à inércia dos controladores de velocidade dos motores 1 e 3. Como estes motores estão sobre o eixo x , a inércia dos controladores 1 e 3 é obtida através da expressão (A. 9). O segundo termo da expressão (A. 11) é relativo à inércia dos controladores 2 e 4. Como estes controladores são perpendiculares ao eixo x e rodam sobre um eixo paralelo ao eixo x , no termo relativo à inercias destes dois controladores, pelo teorema de Steiner, surge md_s^2 .

O momento de inércia em z , $I_{z,S}$ é obtido através da expressão (A. 10) [49]:

$$I_{z,S} = 4 \left[\frac{1}{12} m(a^2 + b^2) + md_s^2 \right]. \quad (\text{A. 12})$$

Na expressão anterior, a multiplicação por 4 deve-se ao facto da inércia ser a mesma para todos os ESC e o termo md_s^2 deve-se ao eixo z dos controladores ser paralelo ao eixo z *quadcopter*.

A.3 Núcleo central

Para calcular o momento de inércia do núcleo central do *quadcopter*, este é aproximado a um cilindro sólido, Figura A. 5.

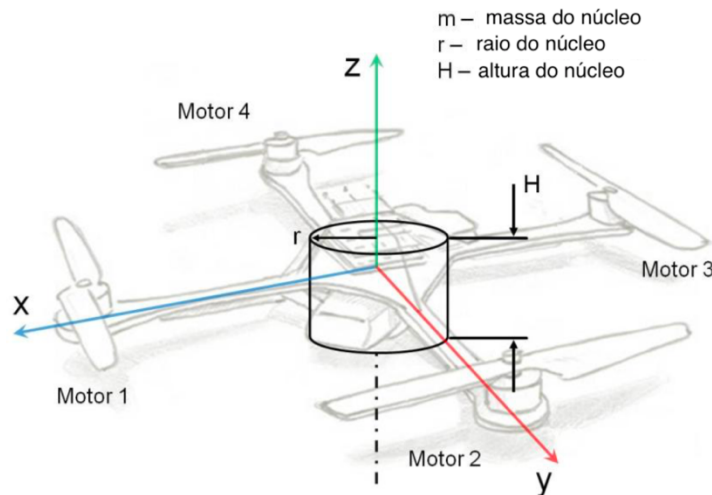


Figura A. 5 - Representação da forma geométrica simplificada do núcleo central do *quadcopter* [49].

Para poder determinar a inércia em cada eixo, é necessário ter em conta os dois tipos de rotações da Figura A. 6.



Figura A. 6 - Dois tipos de rotações possíveis para o núcleo central [49].

A rotação (a) da Figura A. 6 diz respeito aos eixos x e y enquanto que a (b) é em relação ao eixo z . O momento de inércia da rotação em (a) é dado por [49]:

$$I_{CM} = \frac{1}{4}mr^2 + \frac{1}{12}mh^2, \quad (\text{A. 13})$$

em que m é a massa do cilindro, r o raio e h a altura. Para a rotação (b), o momento de inércia é [49]:

$$I_{CM} = \frac{1}{2}mr^2. \quad (\text{A. 14})$$

Com isto os momentos de inércia causados pelo núcleo central em x e em y são obtidos através da expressão (A. 13) [49]:

$$I_{x,N} = I_{y,N} = \frac{1}{4}mr^2 + \frac{1}{12}mH^2, \quad (\text{A. 15})$$

onde r é o raio, H a altura e m a massa.

Como o eixo z do núcleo é o mesmo que o do *quadcopter*, o momento de inércia em z é obtido através da expressão (A. 14) [49]:

$$I_{z,N} = \frac{1}{2}mr^2. \quad (\text{A. 16})$$

A.4 Braços

Para determinar o momento de inércia dos braços do *drone*, estes são considerados como hastes cilíndricas. Para calcular os momentos de inércia em cada eixo, para cada braço é necessário conhecer a massa (m), o comprimento (L), o raio (r) e a distância entre o centro de massa e o braço (d_a). Estas medidas estão ilustradas na Figura A. 7.

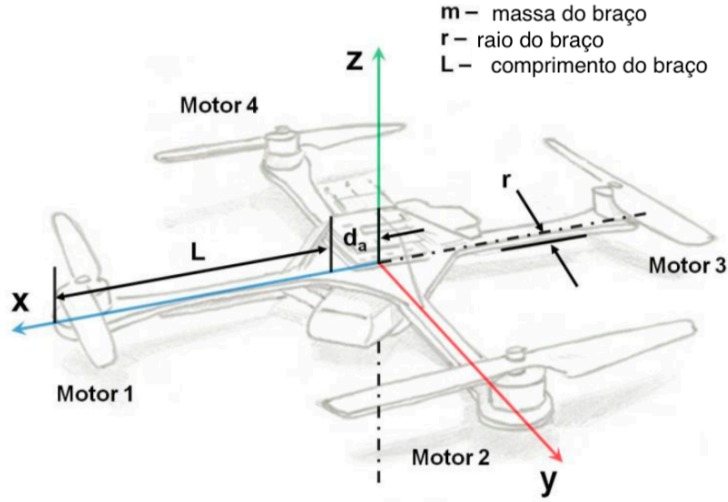


Figura A. 7 - Ilustração das medidas necessárias para calcular o momento de inércia dos braços do *quadcopter* [49].

O momento de inércia dos braços baseia-se nas mesmas rotações e momentos que foram utilizados para calcular a inércia dos motores (Figura A. 3 e expressões (A. 4) e (A. 5)).

O momento de inércia em x e em y , devido à simetria, são iguais e são expressos da seguinte forma [49]:

$$I_{x,A} = I_{y,A} = 2 \left[\frac{1}{2} mr^2 \right] + 2 \left[\frac{1}{4} mr^2 + \frac{1}{3} mL^2 + md_a^2 \right]. \quad (\text{A. 17})$$

O primeiro termo da expressão diz respeito aos braços que estão sobre o eixo de rotação (braços 1 e 3 no caso de $I_{x,A}$, 2 e 4 no caso de $I_{y,A}$) e é obtido através da expressão (A. 5). O segundo termo diz respeito aos braços perpendiculares ao eixo de rotação (braços 2 e 4 no caso de $I_{x,A}$, 1 e 3 no caso de $I_{y,A}$) e é obtido através da expressão (A. 4), em conjunto com o teorema de Steiner.

O momento de inércia em z é semelhante ao segundo termo da expressão (A. 17), porque o eixo z de cada braço é paralelo ao do *quadcopter* e atravessa o seu diâmetro (rotação (a) da Figura A. 3) [49]:

$$I_{z,A} = 4 \left[\frac{1}{4} mr^2 + \frac{1}{3} mL^2 + md_a^2 \right]. \quad (\text{A. 18})$$

A.5 Matriz de inércia

Com os momentos todos calculados para cada componente, cada elemento da matriz de inércia do *quadcopter* é dado pelo somatório de todos os momentos em cada um dos eixos [49]:

$$I_{xx} = I_{x,M} + I_{x,S} + I_{x,N} + I_{x,A}; \quad (\text{A. 19})$$

$$I_{yy} = I_{y,M} + I_{y,S} + I_{y,N} + I_{y,A}; \quad (\text{A. 20})$$

$$I_{zz} = I_{z,M} + I_{z,S} + I_{z,N} + I_{z,A}. \quad (\text{A. 21})$$

Anexo B – Detecção da inclinação através do acelerómetro de três eixos

Os acelerómetros são muito utilizados em dispositivos móveis para detetar a sua inclinação. Nos *quadcopters*, os acelerómetros presentes nos IMU além de medirem a aceleração linear da aeronave também permitem estimar a orientação. Através de um filtro complementar, a orientação do *quadcopter* é obtida através do giroscópio e do acelerómetro.

O acelerómetro é sensível à aceleração linear e também ao campo gravítico. Este sensor mede a diferença entre a aceleração linear do seu referencial e o vetor do campo gravitacional da Terra. Na ausência de aceleração linear, o acelerómetro consegue sentir a rotação do vetor do campo gravítico, permitindo assim estimar os ângulos *roll* e *pitch* do acelerómetro.

A saída de um acelerómetro é dada pelo vetor G_p e depende da aceleração linear do acelerómetro (a_r) e da aceleração gravítica da Terra (g) [50]:

$$G_p = \begin{bmatrix} G_{px} \\ G_{py} \\ G_{pz} \end{bmatrix} = R(g - a_r), \quad (\text{B. 1})$$

em que R é a matriz de rotação que descreve a orientação do acelerómetro em relação ao referencial da Terra.

A estimativa da orientação parte da expressão (B. 1), mas para tal, a aceleração linear tem de ser nula. Qualquer aceleração que o acelerómetro tenha para além da aceleração gravítica, introduz erro na estimativa da orientação. Considerando que a aceleração linear é 0 e que a aceleração gravítica é 1 g, a saída do acelerómetro fica:

$$G_p = \begin{bmatrix} G_{px} \\ G_{py} \\ G_{pz} \end{bmatrix} = Rg = R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (\text{B. 2})$$

A matriz de rotação R é obtida através de um conjunto de três rotações elementares (expressão (2. 1) a (2. 3)). A matriz de rotação depende da ordem de como as rotações de *roll*, *pitch* e *yaw* são aplicadas. No total existem seis matrizes de rotação diferentes. As expressões seguintes mostram a saída do acelerómetro para cada matriz de rotação diferente [50]:

$$R_{xyz} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = R_x(\phi)R_y(\theta)R_z(\psi) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{aligned}
&= \begin{bmatrix} c(\theta)c(\psi) & c(\theta)s(\psi) & -s(\theta) \\ c(\psi)s(\theta)s(\phi) - c(\phi)s(\psi) & c(\phi)c(\psi) + s(\theta)s(\phi)s(\psi) & c(\theta)s(\phi) \\ c(\phi)c(\psi)s(\theta) + s(\phi)s(\psi) & c(\phi)s(\theta)s(\psi) - c(\psi)s(\phi) & c(\theta)c(\phi) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} -s(\theta) \\ c(\theta)s(\phi) \\ c(\theta)c(\phi) \end{bmatrix}; \tag{B. 3}
\end{aligned}$$

$$\begin{aligned}
&R_{yxx} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = R_y(\theta)R_x(\phi)R_z(\psi) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} c(\psi)c(\theta) - s(\theta)s(\phi)s(\psi) & s(\psi)c(\theta) + s(\theta)s(\phi)c(\psi) & -s(\theta)c(\phi) \\ -c(\phi)s(\psi) & c(\phi)c(\psi) & s(\phi) \\ c(\theta)s(\phi)s(\psi) + s(\theta)c(\psi) & -c(\psi)c(\theta)s(\phi) + s(\psi)s(\theta) & c(\theta)c(\phi) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} -s(\theta)c(\phi) \\ s(\phi) \\ c(\theta)c(\phi) \end{bmatrix}; \tag{B. 4}
\end{aligned}$$

$$\begin{aligned}
&R_{xzy} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = R_x(\phi)R_z(\psi)R_y(\theta) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} c(\theta)c(\psi) & s(\psi) & -c(\psi)s(\theta) \\ -c(\phi)c(\theta)s(\psi) + s(\phi)s(\theta) & c(\phi)c(\psi) & c(\theta)s(\phi) + c(\phi)s(\theta)s(\psi) \\ c(\theta)s(\psi)s(\phi) + c(\phi)s(\theta) & -c(\psi)s(\phi) & c(\theta)c(\phi) - s(\theta)s(\phi)s(\psi) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} -c(\psi)s(\theta) \\ c(\theta)s(\phi) + c(\phi)s(\psi)s(\theta) \\ c(\phi)c(\theta) - s(\theta)s(\phi)s(\psi) \end{bmatrix}; \tag{B. 5}
\end{aligned}$$

$$\begin{aligned}
&R_{yzz} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = R_y(\theta)R_z(\psi)R_x(\phi) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} c(\psi)c(\theta) & c(\phi)c(\theta)s(\psi) + s(\theta)s(\phi) & c(\theta)s(\phi)s(\psi) - s(\theta)c(\phi) \\ -s(\psi) & c(\phi)c(\psi) & c(\psi)s(\theta) \\ c(\psi)s(\theta) & -c(\theta)s(\phi) + c(\phi)s(\psi)s(\theta) & c(\theta)c(\phi) + s(\theta)s(\phi)s(\psi) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} c(\theta)s(\phi)s(\psi) - c(\phi)s(\theta) \\ c(\psi)s(\phi) \\ c(\theta)c(\phi) + s(\theta)s(\phi)s(\psi) \end{bmatrix}; \tag{B. 6}
\end{aligned}$$

$$\begin{aligned}
R_{zxy} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} &= R_z(\psi)R_x(\phi)R_y(\theta) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} c(\psi)c(\theta) + s(\theta)s(\phi)s(\psi) & c(\phi)s(\psi) & c(\theta)s(\phi)s(\psi) - s(\theta)c(\psi) \\ -c(\theta)s(\psi) + c(\psi)s(\phi)s(\theta) & c(\phi)c(\psi) & c(\psi)c(\theta)s(\phi) + s(\theta)s(\psi) \\ c(\phi)s(\theta) & -s(\phi) & c(\theta)c(\phi) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} c(\theta)s(\phi)s(\psi) - c(\psi)s(\theta) \\ c(\psi)c(\theta)s(\phi) + s(\theta)s(\psi) \\ c(\theta)c(\phi) \end{bmatrix}; \tag{B. 7}
\end{aligned}$$

$$\begin{aligned}
R_{zyx} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} &= R_z(\psi)R_y(\theta)R_x(\phi) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} c(\psi)c(\theta) & c(\phi)s(\psi) + c(\psi)s(\phi)s(\theta) & s(\phi)s(\psi) - c(\phi)c(\psi)s(\theta) \\ -c(\theta)s(\psi) & c(\psi)c(\phi) - s(\theta)s(\phi)s(\psi) & c(\psi)s(\phi) + c(\phi)s(\psi)s(\theta) \\ s(\theta) & -c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} s(\phi)s(\psi) - c(\phi)c(\psi)s(\theta) \\ c(\psi)s(\phi) + c(\phi)s(\psi)s(\theta) \\ c(\theta)c(\phi) \end{bmatrix}. \tag{B. 8}
\end{aligned}$$

Consegue-se visualizar pelas expressões (B. 3) a (B. 8) que o vetor de gravidade medido é diferente para cada uma das matrizes de rotação. Isto significa que não é possível estimar a orientação sem antes definir a ordem com que são aplicadas as rotações.

Quatro destas expressões podem ser imediatamente eliminadas por serem inadequadas para determinar a orientação do *quadcopter*. A saída do acelerómetro tem três componentes, mas como a amplitude do vetor tem de ser 1 g, na ausência de aceleração linear, então só existem dois graus de liberdade. O vetor do acelerómetro fica situado numa esfera com raio 1 g. Não é possível resolver as expressões que dependem dos três ângulos *roll* (ϕ), *pitch* (θ) e *yaw* (ψ) [50].

As equações (B. 3) e (B. 4) como apenas dependem dos ângulos *roll* (ϕ) e *pitch* (θ), podem ser resolvidas. Normalmente é considerado que o eixo *z* do acelerómetro está alinhado com o campo gravítico. Assim, o acelerómetro não consegue sentir uma rotação sobre esse eixo, daí não ser possível estimar o ângulo de *yaw* a partir das medições deste tipo de sensor. Por isso, é utilizada a matriz R_{xyz} ou R_{yxz} na estimação dos ângulos de *roll* e *pitch*, porque são essas as matrizes que produzem um vetor de saída que apenas depende destes dois ângulos.

A equação (B. 3) pode ser reescrita fazendo relacionar os ângulos θ e ϕ em função do valor normalizado do acelerômetro [50]:

$$\frac{G_p}{\|G_p\|} = \begin{bmatrix} -\text{sen}(\theta) \\ \cos(\theta)\text{sen}(\phi) \\ \cos(\theta)\cos(\phi) \end{bmatrix} \Leftrightarrow \frac{1}{\sqrt{G_{px}^2 + G_{py}^2 + G_{pz}^2}} \begin{bmatrix} G_{px} \\ G_{py} \\ G_{pz} \end{bmatrix} = \begin{bmatrix} -\text{sen}(\theta) \\ \cos(\theta)\text{sen}(\phi) \\ \cos(\theta)\cos(\phi) \end{bmatrix}. \quad (\text{B. 9})$$

Resolvendo a expressão anterior em função dos ângulos de *roll* e *pitch* e colocando o subscrito *xyz* para mostrar que os ângulos foram obtidos com a matriz de rotação R_{xyz} [50]:

$$\tan(\phi_{xyz}) = \frac{G_{py}}{G_{pz}}; \quad (\text{B. 10})$$

$$\tan(\theta_{xyz}) = \frac{-G_{px}}{G_{py}\text{sen}(\phi) + G_{pz}\cos(\phi)} = \frac{-G_{px}}{\sqrt{G_{py}^2 + G_{pz}^2}}. \quad (\text{B. 11})$$

A sequência de rotação R_{xyz} é amplamente utilizada na indústria aeroespacial e é conhecida pela “sequência de rotação aeroespacial” [50].

À semelhança do procedimento feito para a expressão (B. 3), o mesmo pode ser feito utilizando a expressão (B. 4):

$$\frac{G_p}{\|G_p\|} = \begin{bmatrix} -\text{sen}(\theta)\cos(\phi) \\ \text{sen}(\phi) \\ \cos(\theta)\cos(\phi) \end{bmatrix} \Leftrightarrow \frac{1}{\sqrt{G_{px}^2 + G_{py}^2 + G_{pz}^2}} \begin{bmatrix} G_{px} \\ G_{py} \\ G_{pz} \end{bmatrix} = \begin{bmatrix} -\text{sen}(\theta)\cos(\phi) \\ \text{sen}(\phi) \\ \cos(\theta)\cos(\phi) \end{bmatrix}, \quad (\text{B. 12})$$

e resolvendo em função dos ângulos com um subscrito *yxz* da matriz de rotação R_{yxz} obtém-se o seguinte:

$$\tan(\phi_{yxz}) = \frac{G_{py}}{\sqrt{G_{px}^2 + G_{pz}^2}}; \quad (\text{B. 13})$$

$$\tan(\theta_{yxz}) = \frac{-G_{px}}{G_{pz}}. \quad (\text{B. 14})$$

Comparando as expressões (B. 10) e (B. 11) com (B. 13) e (B. 14) verifica-se que os valores dos ângulos estimados são diferentes para a mesma medição do acelerômetro. Esta é uma consequência das matrizes de rotação dependerem da ordem de como são aplicadas as rotações elementares. Desta forma é necessário referir a ordem das rotações quando se estima a orientação de um corpo utilizando o acelerômetro [50].

Anexo C – Modelos de simulação do *quadcopter*

Neste anexo estão presentes imagens dos modelos e controladores implementados em Simulink, e também o código desenvolvido.

C.1 Modelo em linguagem MATLAB

Na Figura C. 1 está representado o modelo, em Simulink, em que implementa a dinâmica do *quadcopter*, bem como o controlo de altura e de orientação.

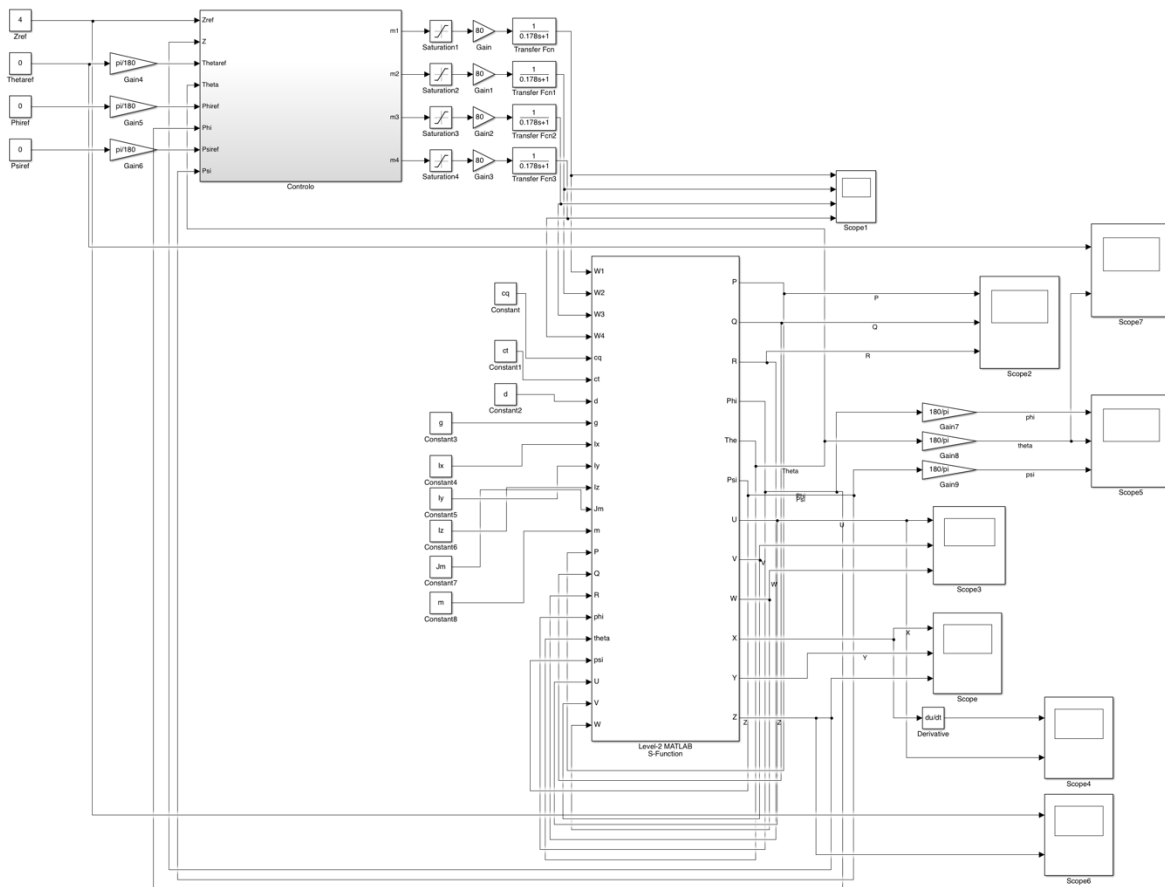


Figura C. 1 - Modelo da dinâmica e controlo do *quadcopter* implementado em Simulink.

C.1.1 Código da dinâmica do *quadcopter*

```
function modelo_Drone(block)
setup(block);
function setup(block)
%número de entradas
block.NumInputPorts=22;
%número de saídas
block.NumOutputPorts=12;
%caracterização das entradas e saídas
for i = 1:22;
block.InputPort(i).Dimensions      = 1;
block.InputPort(i).DirectFeedthrough = false;
```

```

block.InputPort(i).SamplingMode      = 'Sample';
end
for i = 1:12;
block.OutputPort(i).Dimensions      = 1;
block.OutputPort(i).SamplingMode    = 'Sample';
end

block.SampleTimes=[0 0];%tempo contínuo de simulação
block.SimStateCompliance = 'DefaultSimState';
block.NumContStates = 12;
block.RegBlockMethod('InitializeConditions', @InitializeConditions);
block.RegBlockMethod('Outputs', @Outputs);
block.RegBlockMethod('Derivatives', @Derivatives);%atualiza as derivadas
dos estados contínuos

function InitializeConditions(block)
P=0;
Q=0;
R=0;
Phi=0;
The=0;
Psi=0;
U=0;
V=0;
W=0;
X=0;
Y=0;
Z=0;
init = [P,Q,R,Phi,The,Psi,U,V,W,X,Y,Z];
for i=1:12
block.OutputPort(i).Data = init(i);
block.ContStates.Data(i) = init(i);
end
function Outputs(block)
for i = 1:12;
    block.OutputPort(i).Data = block.ContStates.Data(i);
end
function Derivatives(block)
P = block.ContStates.Data(1);
Q = block.ContStates.Data(2);
R = block.ContStates.Data(3);
Phi = block.ContStates.Data(4);
The = block.ContStates.Data(5);
Psi = block.ContStates.Data(6);
U = block.ContStates.Data(7);
V = block.ContStates.Data(8);
W = block.ContStates.Data(9);
X = block.ContStates.Data(10);
Y = block.ContStates.Data(11);
Z = block.ContStates.Data(12);
W1 = block.InputPort(1).Data;
W2 = block.InputPort(2).Data;
W3 = block.InputPort(3).Data;
W4 = block.InputPort(4).Data;
cq = block.InputPort(5).Data;
ct = block.InputPort(6).Data;
d = block.InputPort(7).Data;
g = block.InputPort(8).Data;
Ix = block.InputPort(9).Data;
Iy = block.InputPort(10).Data;
Iz = block.InputPort(11).Data;
Jm = block.InputPort(12).Data;
m = block.InputPort(13).Data;

```

```

P = block.InputPort(14).Data;
Q = block.InputPort(15).Data;
R = block.InputPort(16).Data;
phi = block.InputPort(17).Data;
theta = block.InputPort(18).Data;
psi = block.InputPort(19).Data;
U = block.InputPort(20).Data;
V = block.InputPort(21).Data;
W = block.InputPort(22).Data;
Jb=[Ix,0,0;0,Iy,0;0,0,Iz];
%binários giroscopicos
    %Jm-inercia do motor
    %W1,W2,W3 e W4 velocidades em RPM dos motores
Tau_phi=Jm*Q*(pi/30)*(W1-W2+W3-W4);
Tau_theta=Jm*P*(pi/30)*(-W1+W2-W3+W4);
%Matriz dos momentos aerodinamicos, giroscopicos de de propulsão
    %Configurações "+"
    %d distancia do centro de massa ao motor
    %ct "coeficiente" de propulsão
    %cq coeficiente de torque do sistema motor/hélice
M=[d*ct*(W2^2)-d*ct*(W4^2)+Tau_phi;
-d*ct*(W1^2)+d*ct*(W3^2)+Tau_theta;
-cq*(W1^2)+cq*(W2^2)-cq*(W3^2)+cq*(W4^2)];
%Forças devido às propulsões
Fb=[0;0;ct*((W1^2)+(W2^2)+(W3^2)+(W4^2))];
%Matriz das velocidades angulares P,Q e R
omega_bi=[P;Q;R];
    %Omega_bi-matriz do produto vetorial da velocidade rotacional
    Omega_bi=[0,-R,Q;
R,0,-P;
-Q,P,0];
%Acelerações angular P',Q'e R'
Diff_Omega=(inv(Jb))*(M-Omega_bi*Jb*[P;Q;R]);
%Euler Kinematic equation-determina a taxa de variação dos ângulos de
%Euler no referencial inercial Phi',Theta' e Psi'
H_Phi=[1,tan(theta)*sin(phi),tan(theta)*cos(phi);
0,cos(phi),-sin(phi);
0,sin(phi)/cos(theta),cos(phi)/cos(theta)];
Diff_Phi=H_Phi*omega_bi;%=[P';Q';R']
%Matriz de rotações
C_bi=[cos(theta)*cos(psi),cos(theta)*sin(psi),-sin(theta);
-
cos(phi)*sin(psi)+sin(phi)*sin(theta)*cos(psi),cos(theta)*cos(psi)+sin(phi)
*sin(theta)*sin(psi),sin(phi)*cos(theta);
sin(phi)*sin(psi)+cos(phi)*sin(theta)*cos(psi),-
sin(phi)*cos(psi)+cos(phi)*sin(theta)*sin(psi),cos(phi)*cos(theta)];
%acelerações no referencial da terra
ge=[0;0;-g];
%acelerações no referencial do quadcopter
gb=C_bi*ge;
%acelerações lineares no quadcopter U,V e W
vb=[U;V;W];
Diff_v=(1/m)*Fb+gb-Omega_bi*vb;
%equações de estado da posição X',Y' e Z'
diff_P=C_bi'*vb;
dX=diff_P(1);
dY=diff_P(2);
dZ=diff_P(3);
dP=Diff_Omega(1);
dQ=Diff_Omega(2);
dR=Diff_Omega(3);
dPhi=Diff_Phi(1);
dTheta=Diff_Phi(2);

```

```

dPsi=Diff_Phi(3);
dU=Diff_v(1);
dV=Diff_v(2);
dW=Diff_v(3);
%limite do ch,,o
if ((Z<=0) && (dZ<=0))
    dZ = 0;
    block.ContStates.Data(12) = 0;
end
f = [dP dQ dR dPhi dTheta dPsi dU dV dW dX dY dZ].';
%This is the state derivative vector
block.Derivatives.Data = f;

```

C.2 Modelo em diagrama de blocos

Na Figura C. 2 está representado o modelo em diagrama de blocos da dinâmica do *quadcopter* com os blocos de controlo.

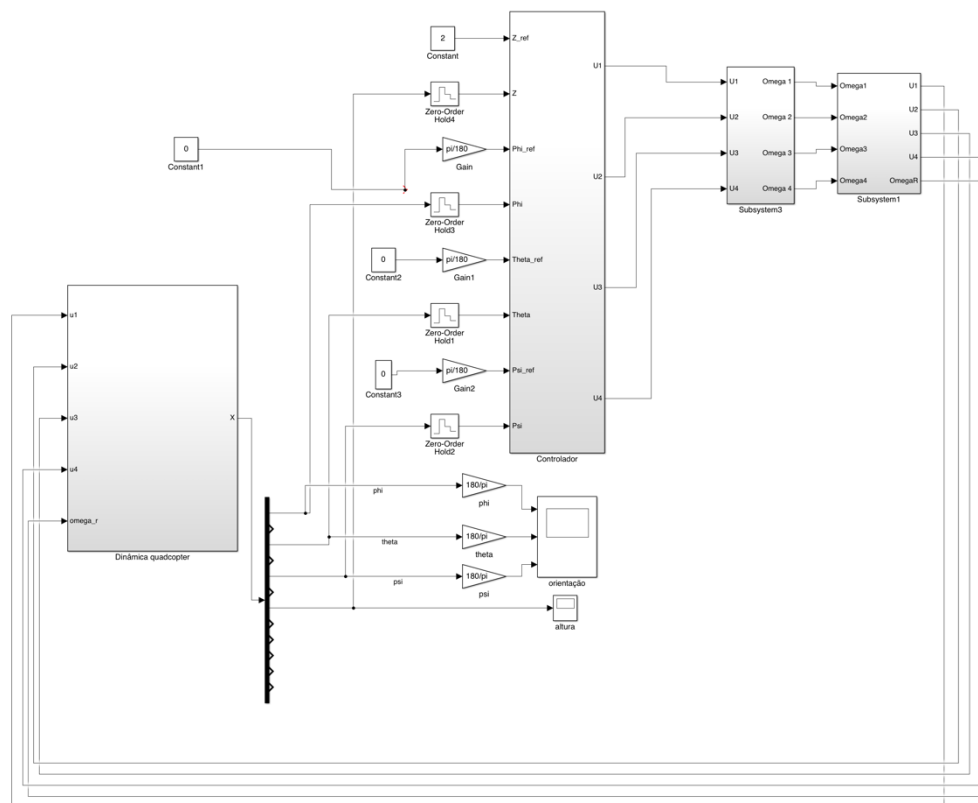


Figura C. 2 - Modelo em diagrama de blocos completo da dinâmica e do controlo do *quadcopter*.

C.3 Controladores

C.3.1 Controlador PID

Na Figura C. 3 está representado o controlador PID da altura do *quadcopter*. Os controladores da orientação são semelhantes a este, apenas mudam as variáveis de controlo e os valores dos compensadores. Os valores dos compensadores proporcional, integral e derivativo foram obtidos através das ferramentas de otimização do Simulink.

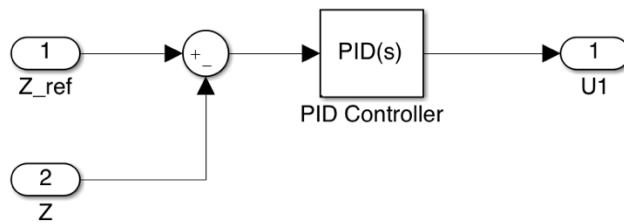


Figura C. 3 - Controlador PID da altura do *quadcopter* implementado no Simulink.

C.3.2 Controlador Fuzzy-PID

A Figura C. 4 apresenta o controlador Fuzzy-PID implementado em Simulink para o controlo da orientação do *quadcopter*.

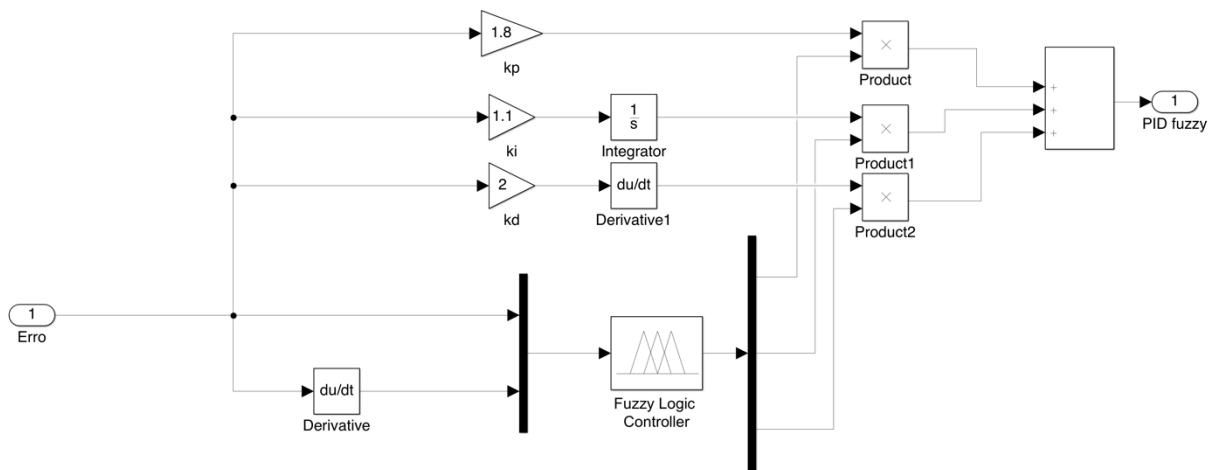


Figura C. 4 - Controlador Fuzzy-PID.

O controlador de lógica fuzzy foi implementado através da ferramenta “Fuzzy Logic Designer” do MATLAB, Figura C. 5. Com este editor, definiram-se as entradas e saídas do controlador fuzzy e inseriram-se as regras de inferência da Tabela 3. 3.

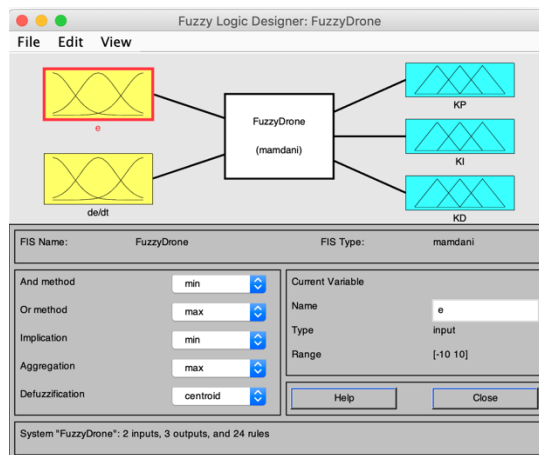


Figura C. 5 - Editor de lógica fuzzy do MATLAB.

C.3.3 Controlador *Backstepping*

Para implementar o controlador com a técnica de *Backstepping* no Simulink, utilizaram-se as expressões (3. 81), (3. 93), (3. 94) e (3. 97). Da Figura C. 6 à Figura C. 9 estão representados os controladores *backstepping* implementados em Simulink para o controlo da orientação e altura do *quadcopter*.

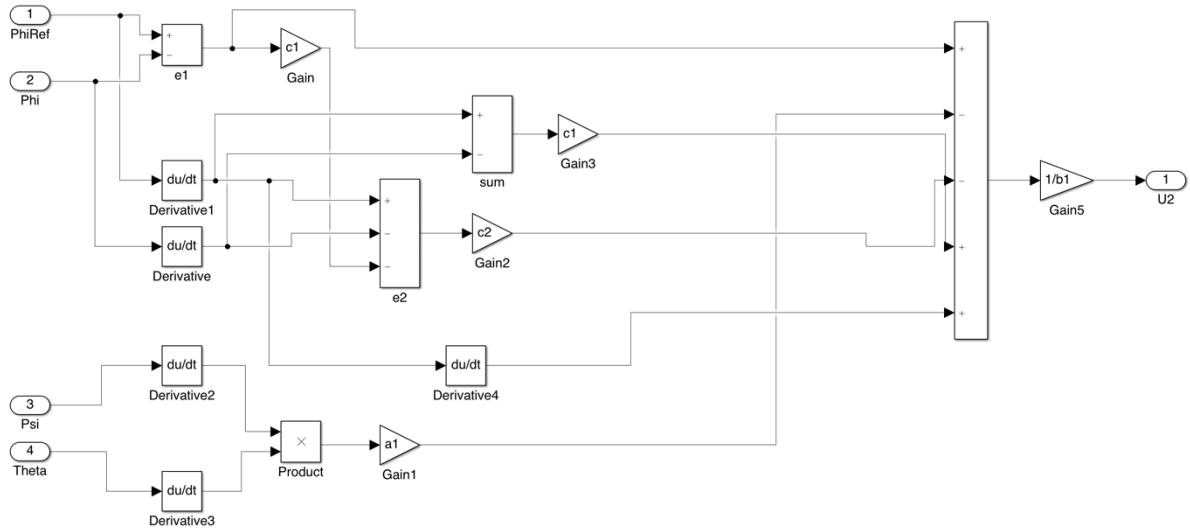


Figura C. 6 - Controlador *backstepping* do ângulo ϕ em simulink.

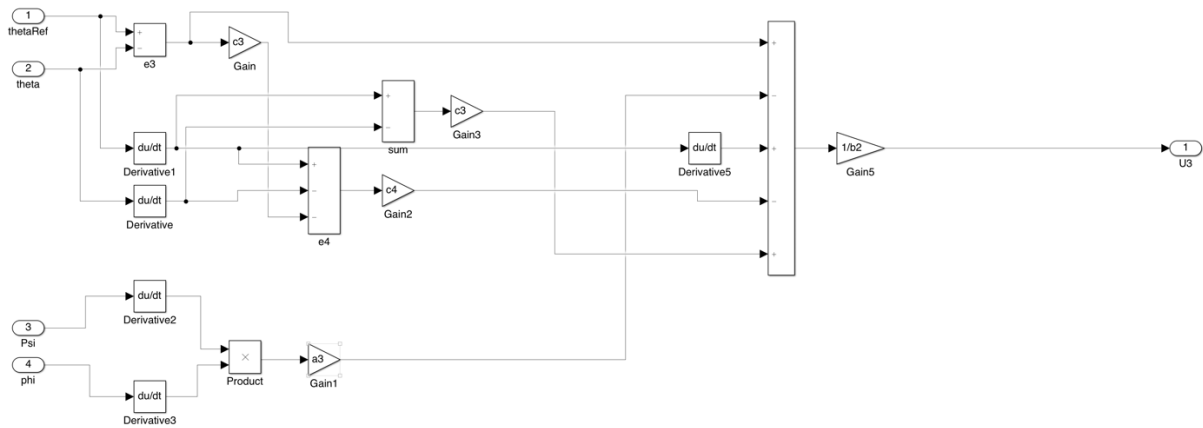


Figura C. 7 - Controlador *backstepping* do ângulo θ .

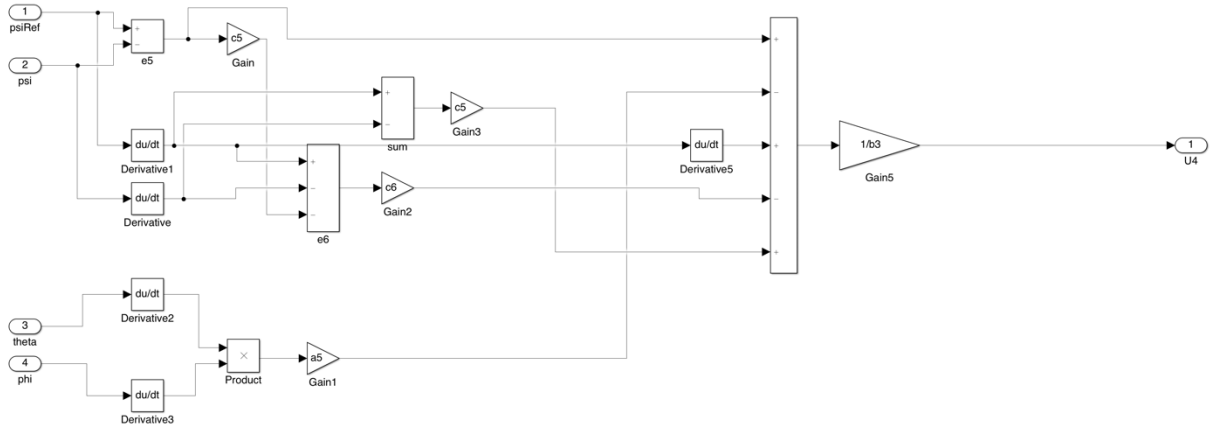


Figura C. 8 - Controlador *backstepping* do ângulo ψ .

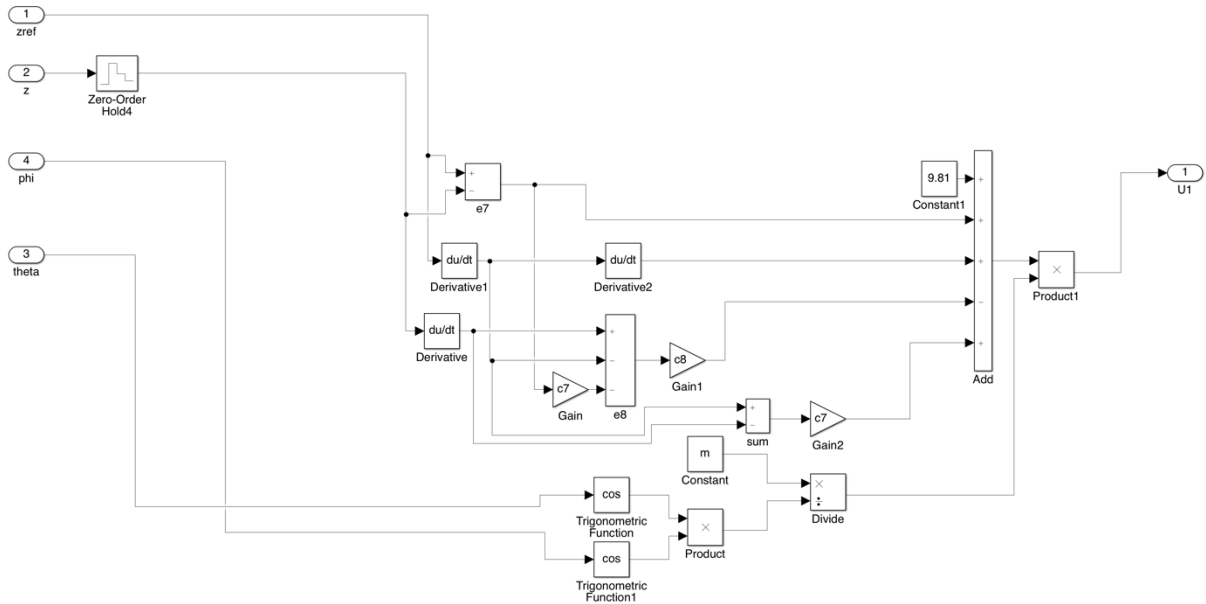


Figura C. 9 - Controlador *backstepping* da altura.