

## Controlo e Simulação de um Quadrirotor convencional

Sérgio Eduardo Aurélio Pereira da Costa

Dissertação para obtenção do Grau de Mestre em

**Engenharia Aeroespacial**

**Júri**

Presidente: Prof. Fernando José Parracho Lau (DEM)

Orientação: Prof. José Raúl Carreira Azinheira (DEM)

Vogais: Prof<sup>a</sup>. Alexandra Bento Moutinho (DEM)

Setembro de 2008

## RESUMO

A presente tese teve como objectivo propor um modelo capaz de controlar e simular um UAV de asas rotativas. Esse UAV é caracterizado por quatro rotores, responsáveis pela geração do movimento, integrados numa estrutura cruzada que embarca, no seu centro, o conjunto do material necessário à sua automatização: este tipo de UAV denomina-se Quadrirotor.

Partindo de um Quadrirotor existente, o ALIV - Autonomous Locomotion Individual Vehicle, procurou-se construir em ambiente MATLAB/SIMULINK um modelo adaptado, MALIV - My Autonomous Locomotion Individual Vehicle, capaz de pôr à prova as reais capacidades de um UAV deste género. Para isso, foram utilizados, não só os sensores a bordo do ALIV, como foram implementados dois módulos diferentes para melhorar o seu controlo integrado. Um deles faz uso de um joystick, e o segundo de uma câmara embarcada para o controlo da posição do Quadrirotor.

Através das saídas provenientes dos sensores modelados, foi possível realizar um filtro de Kalman para a observação dos estados globais. A essa estimativa foi aplicado um controlador óptimo, o LQR – Regulador Linear Quadrático, para consequentemente se fechar o anel através da realimentação para cada um dos motores. A este tipo de controlador, que usufrui do princípio da separação dá-se o nome de: LQG – Regulador Linear Gaussiano.

Por fim, com base nos módulos anteriores, foi construído um módulo final em tempo real para simular a reacção do MALIV em missões destinadas a UAVs deste género: este módulo integra a estabilização a uma dada altitude controlado pelo joystick, como o seguimento de um objecto animado com velocidades moderadas.

**Palavras-chave:** Quadrirotor; Filtro de Kalman; Controlo Óptimo; Regulador Linear Gaussiano; Simulação em Tempo Real.

## ABSTRACT

The present thesis proposes a model to simulate and control a rotary wing UAV. This UAV is characterized by four rotors that are responsible for the whole movement and are placed in a cross structure that includes, in its center, all the necessary equipment for its automatization. Such a UAV is usually called Quadrotor.

Based on an existing Quadrotor, named ALIV - Autonomous Locomotion Individual vehicle, a model was built in MATLAB/SIMULINK, named MALIV - My ALIV, able to test the real capacities of a UAV like this one.

Through the modelling of its real sensors, it was possible to design a Kalman Filter to estimate the Quadrotor state. Using this estimated state, a Linear Quadratic Regulator (LQR) was applied, to obtain the feedback for each one of the rotors. This type of control design, based on the separation principle, is often called LQG - Linear Quadratic Gaussian.

Finally, based on the above modules, a final real-time SIMULINK model was built, which simulates the ALIV reaction for missions targetted for this type of UAVs: this integrated model, either performs the stabilization at a certain altitude, under joystick control, or visually tracks a slowly moving object.

**Keywords:** Quadrotor; Kalman Filter; Optimal Control; LQG - Linear Quadratic Gaussian; Real-Time Simulation.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>9</b>
1.1	UAVs - Introdução . . . . .	9
1.2	Quadrirotos convencionais . . . . .	9
1.3	História dos Quadrirotos . . . . .	10
1.3.1	O princípio da História dos Quadrirotos . . . . .	10
1.3.2	Os Quadrirotos presentemente . . . . .	11
1.4	Objectivos . . . . .	13
1.5	Organização/Estrutura da tese . . . . .	14
<b>2</b>	<b>Plataforma</b>	<b>15</b>
2.1	Plataforma Quadrirotor convencional . . . . .	15
2.2	Plataforma ALIV . . . . .	16
2.2.1	Funcionamento geral do ALIV . . . . .	17
2.2.2	Equipamento a bordo do ALIV . . . . .	18
2.3	Plataforma considerada no estudo . . . . .	19
<b>3</b>	<b>Modelação matemática do problema</b>	<b>20</b>
3.1	Modelação da dinâmica . . . . .	20
3.2	Modelação da cinemática . . . . .	21
3.2.1	Ângulos de Euler . . . . .	21
3.2.2	Quaterniões . . . . .	22
3.3	Modelação do conjunto motor mais hélice . . . . .	24
3.4	Peso e momentos de inércia . . . . .	26
3.5	Sensores . . . . .	28
3.5.1	Acelerómetros . . . . .	28
3.5.2	Magnetómetro . . . . .	29
3.5.3	Câmara embarcada . . . . .	30
<b>4</b>	<b>Controlador</b>	<b>32</b>
4.1	Linearização das equações . . . . .	32
4.2	LQR (Regulador Quadrático Linear) . . . . .	34
4.3	Estimador . . . . .	34
<b>5</b>	<b>Implementação em SIMULINK</b>	<b>37</b>
5.1	Módulos finais . . . . .	38
5.1.1	Módulo com Joystick . . . . .	38
5.1.2	Módulo de visualização . . . . .	40
5.1.3	Módulo integrado . . . . .	41
5.2	Tempo real . . . . .	42
5.2.1	Criação de um executável através do <i>Real-Time Workshop</i> . . . . .	42
5.2.2	Real-Time Windows Target . . . . .	43
5.3	Visualização 3D dos resultados . . . . .	44

<b>6</b>	<b>Apresentação e discussão dos resultados</b>	<b>46</b>
6.1	Sistema em anel fechado ideal . . . . .	46
6.1.1	Cálculo da matriz de ganho $K$ . . . . .	47
6.1.2	Apresentação dos resultados para o anel fechado ideal . . . . .	48
6.2	Sistema com sensor de posição ideal . . . . .	50
6.2.1	Construção do estimador . . . . .	50
6.2.2	Apresentação dos resultados para o sensor de posição ideal . . . . .	52
6.3	Módulos finais . . . . .	57
6.3.1	Módulo com joystick . . . . .	58
6.3.2	Módulo de visualização . . . . .	63
6.3.3	Módulo integrado . . . . .	69
<b>7</b>	<b>Conclusões</b>	<b>79</b>
7.1	Trabalhos futuros . . . . .	79

## Lista de Figuras

1	Quadrirotor convencional [4]	10
2	Primeiro Quadrirotor, Bréguet-Richet Quad-Rotor Helicopter, 1907 [5]	10
3	Quadrirotor de Etienne Oemichen, 1922 [3]	10
4	Draganflyer da RC Toys [9]	11
5	X-4 Flyer II [11]	12
6	Quadrirotor, projecto STARMAC II[12]	12
7	ALIV - Nomenclatura	15
8	Relação de rotação para os movimentos do Quadrirotor	15
9	ALIV	16
10	ALIV - Translação lateral	17
11	ALIV - Translação frontal	18
12	ALIV - Movimento de guinada	18
13	Emissor	19
14	Receptor	19
15	Referenciais: NED e ABC	20
16	Funcionamento genérico de um motor eléctrico de corrente contínua	24
17	Actuador	25
18	Hélice 9x4,5 APCE	26
19	Filosofia dos módulos durante a fase de simulação	37
20	Ilustração do módulo com joystick durante a fase de simulação	40
21	Ilustração do módulo de visualização durante a simulação	41
22	Ilustração do módulo integrado durante a simulação	42
23	Implementação do joystick em tempo real	44
24	O ALIV real e virtual	45
25	Simulação do ALIV no FLIGHTGEAR	45
26	Sistema em anel fechado ideal	46
27	Posição para deslocação a $5m$ de altitude e $3m$ segundo $x$ e $y$ (anel fechado ideal)	48
28	Atitude para deslocação a $5m$ de altitude e $3m$ segundo $x$ e $y$ (anel fechado ideal)	48
29	Rotação nos motores na deslocação a $5m$ de altitude e $3m$ segundo $x$ e $y$ (anel fechado ideal)	49
30	Sistema com sensor de posição ideal	50
31	Estimador em diagrama de blocos para o caso do sistema com sensor de posição ideal	51
32	Erro entre os estados reais e estimados, sensor de posição ideal	53
33	Posição para deslocação a $5m$ de altitude e $3m$ segundo $x$ e $y$ , sensor de posição ideal	54
34	Atitude para deslocação a $5m$ de altitude e $3m$ segundo $x$ e $y$ , sensor de posição ideal	54
35	Erro entre a posição estimada e real, sensor de posição ideal	55
36	Erro entre a atitude estimada e real, sensor de posição ideal	56
37	Posição e atitude para uma situação limite, sensor de posição ideal	56
38	Diagrama de blocos do módulo integrado	57
39	Controlador em diagrama de blocos adoptado para o módulo com joystick	58
40	Posição $xyz$ ao longo da segunda trajectória	60

41	Atitude ao longo da segunda trajectória . . . . .	60
42	Voltagem aplicada aos motores ao longo da segunda trajectória . . . . .	61
43	Velocidades lineares ao longo da segunda trajectória . . . . .	62
44	Pontos de referência no referencial NED, $R_{NED}$ . . . . .	63
45	Trajectória do objecto ao longo da simulação no referencial NED, $R_{NED}$ . . . . .	66
46	Posição de vigilância $xyz$ ao longo da trajectória . . . . .	66
47	Atitude ao longo da trajectória de vigilância . . . . .	67
48	Posição de vigilância $xyz$ ao longo da trajectória, segunda trajectória . . . . .	67
49	Atitude ao longo da segunda trajectória de vigilância . . . . .	68
50	Velocidades lineares ao longo da segunda trajectória . . . . .	68
51	Posição do MALIV ao longo do tempo, Missão nº1 . . . . .	70
52	Selector dos módulos ao longo da primeira missão . . . . .	70
53	Posição do MALIV ao longo do tempo, Missão nº2 . . . . .	71
54	Selector dos módulos ao longo da segunda missão . . . . .	71
55	Erro associado entre as saídas estimas e as saídas reais para a coordenada $y$ dos pontos de referência . . . . .	72
56	Velocidades lineares do MALIV ao longo do tempo, Missão nº2 . . . . .	72
57	Atitude do MALIV ao longo do tempo, Missão nº2 . . . . .	73
58	Alteração dos pontos de referência em diagrama de blocos . . . . .	74
59	Trajectória do objecto a ser vigiado segundo $xy$ . . . . .	75
60	Trajectória do objecto a ser vigiado segundo $xy$ ao longo do tempo . . . . .	75
61	Selector dos módulos ao longo da terceira missão . . . . .	76
62	Posição do MALIV ao longo do tempo, Missão nº3 . . . . .	76
63	Atitude do MALIV ao longo do tempo, Missão nº3 . . . . .	77
64	Velocidades lineares do MALIV ao longo do tempo, Missão nº3 . . . . .	77
65	Equações em diagrama de blocos que definem a dinâmica do ALIV . . . . .	98
66	Equações em diagrama de blocos que definem os sensores do ALIV . . . . .	98
67	Parâmetros adoptados para a resolução da simulação . . . . .	99
68	Configuração adoptada para a geração do executável, “ <i>Generic Real-Time Target</i> ” . . . . .	99
69	“ <i>Hardware Implementation</i> ” do “ <i>Configuration Parameters</i> ” . . . . .	100
70	Configuração adoptada para a geração do executável, “ <i>Real-Time Windows Target</i> ” . . . . .	100
71	“ <i>offsets</i> ” do ficheiro <i>ufo.xml</i> . . . . .	101
72	Animação da primeira hélice no FLIGHTGEAR através do ficheiro <i>ufo.xml</i> . . . . .	102
73	Blocos para interface entre o Simulink e o FLIGHTGEAR . . . . .	102
74	<i>Flat Earth to LLA</i> utilizado . . . . .	103
75	Modelo completo utilizado na interface entre o Simulink e o FLIGHTGEAR . . . . .	103

## Lista de Tabelas

1	Diferencial de rotação aplicado aos rotores para obtenção dos movimentos desejados	16
2	Características do motor AXI 2212-12 . . . . .	25
3	Características da hélice $9 \times 4,5$ APCE . . . . .	26
4	Peso dos componentes a bordo do ALIV . . . . .	27
5	Limitações dos Acelerómetros utilizados . . . . .	29



# 1 Introdução

## 1.1 UAVs - Introdução

Os UAVs (Unmanned Aerial Vehicle) são definidos de acordo com o Departamento de Defesa dos Estados Unidos, como veículos aéreos com propulsão própria e que não possuem uma cabine de pilotagem. Fazem em geral uso de forças aerodinâmicas para provocarem sustentação, e podem ser voados autonomamente ou por controlo remoto [1].

Os UAVs tornaram-se num objecto de estudo e desenvolvimento depois de terem sido provadas as suas capacidades para enfrentarem situações em que por alguma razão o objectivo da missão não fosse a mais aconselhada para uma aeronave de pilotagem convencional. Os UAVs têm actualmente variadíssimas configurações e capacidades, de acordo com as missões a que se propõem. Entre essas missões, destacam-se situações de guerra, onde os UAVs podem desempenhar um papel importante quer em termos de reconhecimento e vigilância, quer em termos de bombardeamento. Mas não foram só para situações de guerra que os UAVs se tornaram importantes, actualmente existem inúmeros UAVs que desempenham funções preponderantes na observação de fenómenos meteorológicos, onde não é possível a presença humana, em zonas de vigilância prolongadas em que não se é afectado pelo cansaço inerente dos pilotos e onde situações de voo estabilizado são conseguidas, em laboratórios e universidades onde são usados para fins de investigação científica, entre muitas outras situações [2].

Ao longo do tempo surgiram inúmeros UAVs, construídos e desenvolvidos de acordo com o fim a que se propunham. De entre as várias configurações, destacam-se os UAVs de asa rotativa, por serem os pioneiros no tipo de UAV utilizado para o desenvolvimento desta tese, a qual se cingirá ao controlo e simulação de um Quadrirotor.

## 1.2 Quadrirotores convencionais

O UAV de asa rotativa utilizado é o chamado Quadrirotor, ou Quadrotor. O Quadrirotor convencional é um UAV de dimensões reduzidas constituído por quatro rotores fixos. Os rotores são instalados em cada um dos quatro cantos de uma estrutura cruzada, que embarca no seu centro, o conjunto de equipamento necessário para a sua automatização, Figura 1. De modo a anular a resistência associada à rotação das pás das hélices, cada um dos pares de rotores opostos, gira num sentido diferente. É por isso considerado de fácil construção. Outra vantagem é o facto de, por possuir quatro rotores, os diâmetros das hélices poderem ser menores do que as de um helicóptero convencional. Sendo assim, cada um deles necessita de menor energia cinética, e ainda possibilita ao UAV alcançar locais de menor acessibilidade. No entanto, os Quadrirotores apresentam uma dinâmica algo difícil de se conseguir estabilizar, que se poderá tornar numa dificuldade acrescida, principalmente se o mesmo for constituído por equipamentos de auxílio de “baixo custo”. Actualmente, os Quadrirotores possuem diversos sensores e sistemas de controlo para obter a sua estabilização, e a grande vantagem é a simplicidade de construção, e o facto de se poder controlá-lo integralmente, variando apenas a velocidade de rotação dos rotores [3].



Figura 1: Quadrirotor convencional [4]

### 1.3 História dos Quadrirotos

#### 1.3.1 O princípio da História dos Quadrirotos

Os Quadrirotos surgiram no princípio dos anos 1900 e vários exemplares foram criados nos vinte anos seguintes. Inicialmente, a ideia seria transportar um piloto que controlasse os quatro rotores. O primeiro Quadrirotor da história foi criado em 1907 por Charles Richet e pelos irmãos Breguet com o nome Bréguet-Richet Quad-Rotor Helicopter (Figura 2). O Quadrirotor pesava cerca de  $578\text{ kg}$ , e possuía um mecanismo básico de ligação dos controlos do piloto aos rotores. Consta-se que, não conseguiu levantar mais que  $1.5\text{ m}$ , não conseguindo obter a estabilidade necessária à sua pilotagem [5]. Apesar dos resultados insatisfatórios, Charles Richet e os irmãos Breguet introduziram a teoria dos pares de rotores a rodar em sentidos opostos, ainda hoje utilizada.

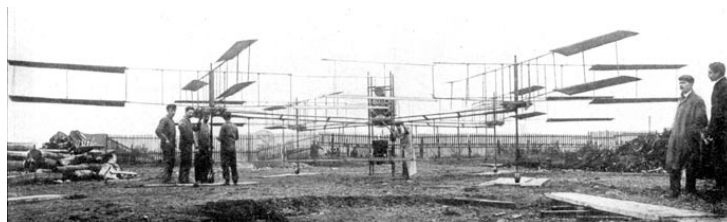


Figura 2: Primeiro Quadrirotor, Bréguet-Richet Quad-Rotor Helicopter, 1907 [5]

Outro Quadrirotor na história dos UAVs de asa rotativa, foi criado em 1922 por Etienne Oemichen. Este Quadrirotor possuía para além dos quatro rotores, oito propulsores, estando tudo ligado ao mesmo motor (Figura 3). Para a altura em que se concretizou, este Quadrirotor revelou um considerável grau de liberdade e de controlabilidade, realizando mais de mil ensaios durante a década de 1920 .

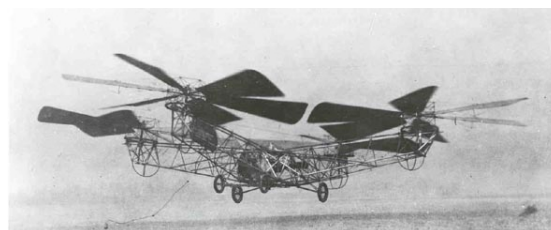


Figura 3: Quadrirotor de Etienne Oemichen, 1922 [3]

Devido à dificuldade para se atingir a estabilidade de uma aeronave deste tipo, bem como da carga física exercida por o piloto durante a fase de pilotagem, os Quadrirotos acabaram por perder

o interesse, ressurgindo sobre a forma de UAVs de asas rotativas entre as décadas de 80 e 90. A sua simplicidade de construção, o facto de todos os rotores contribuírem para a força propulsiva, e ainda a carga útil à descolagem serviram como principais atributos[3].

### 1.3.2 Os Quadrirotores presentemente

Hoje em dia, existem não só diversos Quadrirotores comercializados, como vários laboratórios de investigação utilizam-nos para realizarem experiências para diferentes fins. Os objectivos destas investigações passam primordialmente por testar micro elementos para controlo de aeronaves não tripuladas, bem como a obtenção de novos processos de estabilização e controlo dos Quadrirotores [6, 7, 8]. Neste capítulo, serão apresentadas algumas dessas investigações, tendo como base os Quadrirotores.

O Draganflyer (Figura 4) é um Quadrirotor comercializado pela RC Toys [9]. O Draganflyer é controlado através de um transmissor de radio controlo (R/C) e a sua estrutura é composta por tubos de fibra de carbono que lhe conferem um peso reduzido. Da electrónica existente a bordo do Draganflyer, destaca-se o receptor dos comandos do piloto, três giroscópios piezoeléctricos, e um micro controlador para efectuar cálculos de controlo e enviar a tensão necessária para os motores. Actualmente, o Draganflyer já possui quatro sensores infravermelhos, que possibilitam que este se estabilize sozinho quando opera no exterior. Muitas universidades e investigadores utilizam este modelo como base das suas investigações.



Figura 4: Draganflyer da RC Toys [9]

Na Austrália foi desenvolvido e testado outro Quadrirotor, o X-4 Flyer, Figura 5 [10, 11]. Um dos testes referiu-se ao estudo da influência do peso das unidades de medidas inerciais. Para isso, foi construída uma nova unidade muito mais leve que as comerciais existentes, e verificou-se que melhores condições de estabilidade eram conseguidas. Foi introduzida uma linearização do modelo da dinâmica em torno de um ponto de equilíbrio, da qual surgiram as funções de transferência que relacionam as entradas do piloto com os ângulos de Euler. O X-4 Flyer construído pesa cerca de  $2\text{ kg}$  e tem um comprimento de  $70\text{ cm}$  tendo as hélices um diâmetro de  $28\text{ cm}$ . Os testes a este modelo realizaram-se com uma bateria ligada por um fio ao Quadrirotor, mas revelaram ser insuficientes para conferir propulsão necessária ao Quadrirotor para este se tornar controlável. Com base nisto, os trabalhos passaram por construir uma nova estrutura, capaz de realizar maior sustentação, ser mais leve, embarcar as fontes de energia, um sistema *wireless* e uma câmara. Este novo design é caracterizado por possuir os rotores invertidos (Figura 5), e simulações realizadas, indicaram que podia ser fácil

conduzi-lo através de um controlo remoto ou de um piloto automático.

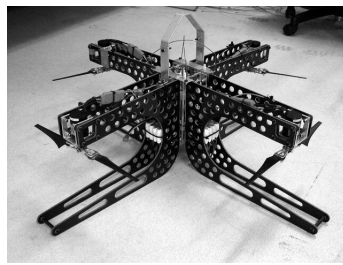


Figura 5: X-4 Flyer II [11]

Um grupo de investigadores da universidade de Stanford, Stanford Testbed of Autonomous Rotorcraft for Multi-Agent (STARMAC) [12], está a realizar um projecto que visa demonstrar capacidades de vigilância e controlo em ambiente real por parte de um conjunto de Quadrirotores (Figura 6). O projecto propõe-se a demonstrar o controlo de vários Quadrirotores em simultâneo, que voarão com total autonomia seguindo uma trajectória fornecida. Inicialmente, e numa primeira fase, os objectivos passaram pelo domínio dos conceitos de voo. Nesta fase do projecto foi utilizado o modelo comercial DraganFlyer IV, no entanto este revelou ter uma dinâmica muito instável e muito rápida que o tornava extremamente difícil de pilotar. Assim foi criado com sucesso um controlador a bordo, que através do amortecimento tornasse a pilotagem mais facilitada. Quanto ao seu estado era definido por três giroscópios, três acelerómetros e três magnetómetros mais um GPS para o controlo da posição. De modo a alcançarem os objectivos propostos, realizaram ainda novas configurações tanto a nível da electrónica de bordo como da estrutura utilizada. No final desta primeira fase, apesar dos primeiros objectivos terem sido obtidos, verificaram-se algumas limitações do Quadrirotor que levaram para a segunda fase do projecto, a alteração aos elementos do Quadrirotor original.

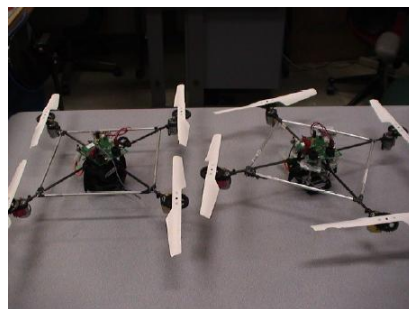


Figura 6: Quadrirotor, projecto STARMAC II[12]

De entre as modificações ocorridas, destacam-se: a instalação de novos rotores para se obter maior força de sustentação e com isso possibilitarem a instalação de baterias mais poderosas e de outros sistemas electrónicos. O aumento da resistência de elementos estruturais, e a troca de elementos de comunicação via *bluetooth* por *Wifi* que se traduziu numa melhoria da estimação da posição via GPS. Para além do STARMAC II (Figura 6) experimental, estão em construção outros cinco Quadrirotores, que com base nas investigações anteriores, pretendem começar a realizar missões em ambiente real.

Um grupo de investigadores da Pensilvânia [13] desenvolveu um Quadrirotor usando um modelo comercial, o HMX-4, semelhante ao Draganflyer. Neste trabalho são usados dois computadores, um

a bordo e outro na estação de trabalho, para combinarem informação dos sensores inerciais a bordo e da câmara no solo que controla o Quadrirotor através de marcas específicas colocadas na base da estrutura do mesmo. Devido às limitações de peso do HMX-4, nem GPS nem acelerómetros adicionais puderam ser colocados. O computador da estação de trabalho processa as imagens da câmara posicionada no solo e calcula as entradas dos motores. O computador de bordo estabiliza o Quadrirotor usando a informação dos giroscópios e recebendo as entradas para os motores provenientes do computador da estação de trabalho. Depois deste trabalho, desenvolveram desta vez um método com duas câmaras, uma a bordo e outra no solo, para estimar a altura e a atitude do Quadrirotor [14]. Este processo demonstrou ter erros menores quer na posição quer na atitude.

Uma equipa do Instituto Federal de Tecnologia Suíça (EPFL) [15] trabalhou no controlo da razão angular e na altitude de um Quadrirotor. Antes da implementação real, foram realizadas inúmeras simulações em ambiente MATLAB. Usaram a dinâmica do modelo do Quadrirotor para testar dois controladores, o PID (proportional, integral and derivative) e o óptimo (LQ: Linear Quadratic). A função do controlador era estabilizar a altitude enquanto compensava os erros iniciais nos ângulos de rolamento, picada e guinada. O LQ obteve um resultado pior do que o PID, devido ao facto da dinâmica do actuador não estar incluída na análise usada para determinar a matriz de ganho [15].

## 1.4 Objectivos

A presente tese tem como objectivo principal obter o controlo e a simulação de um Quadrirotor convencional através do MATLAB/SIMULINK, para uma futura implementação num UAV deste tipo, de modo a melhorar a estabilidade e a operacionalidade do mesmo.

Para esta tese foi utilizado como modelo de referência um novo conceito de Quadrirotor, o ALIV - Autonomous Locomotion Individual Vehicle<sup>1</sup>. O ALIV é um Quadrirotor inovador que para além das características gerais de um UAV deste tipo, possui ainda graus de liberdade extra como consequência da rotação de dois dos seus rotores. Este tipo de capacidade, leva a que seja possível não só deslocar o ALIV horizontalmente, como por outro lado, seja possível estabilizá-lo em posições diferentes das habituais. Actualmente o ALIV é pilotado através de dois joysticks, um que basicamente efectua o controlo de potência e as deslocações horizontais devido à rotação própria dos dois rotores, e outro que controla a atitude do ALIV e a consequente deslocação horizontal através do diferencial de rotação entre os pares de rotores, situação convencional. Desta forma é extremamente complicado conseguir estabilizar e controlar o ALIV de acordo com as missões pretendidas. Por isso, foi proposta a simulação e o controlo do ALIV em termos convencionais, isto é, desprezando a liberdade adicional de rotação dos dois rotores. Esta situação traz o inconveniente de não fazer uso das propriedades adicionais deste Quadrirotor e de todos os benefícios que daí possam surgir, por outro lado, vem simplificar o problema, tornando-se posteriormente num óptimo ponto de partida para o projecto final em que tais características sejam consideradas. Com base nisto, foi desenvolvido um modelo matemático representativo da situação convencional, e criado um sistema de controlo adequado às suas exigências operacionais, tornando-o mais versátil e de melhor pilotagem. Para isso foram tidas

---

<sup>1</sup>O ALIV insere-se num projecto liderado pelo Eng. Severino Raposo e tem uma patente pendente relativa ao seu sistema e processo: WO 2008/054234 (PCT/PT2006/000026) System and Process of vector propulsion with independent control of three translation and three rotation axis. Ao Eng. Severino Raposo presto os meus mais sinceros agradecimentos pela disponibilidade prestada no esclarecimento de qualquer dúvida referente ao assunto, e à facultação de todos os dados presentes nesta tese a que a ele se referem.

em consideração todas as características do ALIV de modo a aproximar o mais possível a simulação da realidade, e tornar o modelo mais fiável para uma futura implementação.

Analisando o equipamento do ALIV, verifica-se que este não possui qualquer sensor de posição, fazendo com que o mesmo não possa funcionar de uma forma autónoma. É objectivo desta tese apresentar um modelo em que apenas um joystick seja o responsável pelo controlo da posição, e posteriormente, um sensor de posição baseado numa câmara incorporada que permita ao ALIV tornar-se completamente autónomo. Em qualquer destas situações, fazendo uso também dos outros sensores a bordo do ALIV, os acelerómetros e o magnetómetro, foi possível controlar e estabilizar o Quadrirotor com maior facilidade, conseguindo com isto, demonstrar as reais capacidades que um UAV deste tipo pode apresentar em diversas missões.

## 1.5 Organização/Estrutura da tese

A presente tese, após a respectiva introdução em que é dado a conhecer o estado da arte e um pouco da história dos Quadrirotores convencionais ao longo do capítulo 1, prossegue no capítulo 2 com a explicação da plataforma do Quadrirotor convencional e a descrição do ALIV, o Quadrirotor utilizado como referência para todo o trabalho. O capítulo contém uma breve introdução às características adicionais que o mesmo oferece em comparação com um Quadrirotor convencional, e as principais diferenças que daí advêm na obtenção dos movimentos transversais. Ainda no mesmo capítulo, é explicado ao pormenor o conceito da plataforma convencional de um Quadrirotor. Utilizando o ALIV como exemplo, é demonstrado como os seis graus de liberdade podem ser obtidos através única e exclusivamente do diferencial de rotação dos pares de rotores. O capítulo termina com a descrição das características principais do equipamento a bordo do ALIV que serão utilizadas futuramente na modelação matemática do problema.

O capítulo 3 apresenta o modelo matemático referente à dinâmica e à cinemática com base na descrição da plataforma de um Quadrirotor convencional ilustrada no capítulo 2, bem como de algum equipamento a bordo do ALIV. O mesmo capítulo inclui nomeadamente, o cálculo dos momentos de inércia, a estimativa do peso, e a modelação matemática dos actuadores com base no conjunto motor mais hélice, atendendo às características dos componentes utilizados e anteriormente apresentados no capítulo 2. O mesmo capítulo termina com a modelação dos sensores utilizados para o controlo da posição e da atitude do ALIV. Todas estas variáveis revelam ser de grande importância não só em termos das equações da dinâmica como da aproximação entre o modelo real e o simulado.

O capítulo 4 apresenta o controlador abordado para a resolução de um problema deste tipo, justificando as aproximações tomadas que simplificam e justificam o uso de tal controlador, o controlador óptimo LQR - Regulador Linear Quadrático.

O capítulo 5 apresenta o trabalho realizado em ambiente MATLAB/SIMULINK. Neste capítulo, é apresentada a filosofia dos módulos de simulação implementados para o cumprimento dos objectivos a que se propõe a presente tese, bem como a adaptação dos mesmos para simulações em tempo real e na demonstração virtual em 3D dos resultados, utilizando o simulador de voo, FLIGHTGEAR.

Por fim, no capítulo 6, são apresentados e discutidos os resultados obtidos ao longo do trabalho desenvolvido. Estes resultados englobam não só a fase inicial de validação da dinâmica e do estimador implementado, bem como da funcionalidade dos módulos finais.

No capítulo 7, são apresentadas as conclusões e as possibilidades de trabalho futuro.

## 2 Plataforma

### 2.1 Plataforma Quadrirotor convencional

Como foi visto na introdução, os movimentos associados a um Quadrirotor convencional estão puramente associados à variação de rotação de cada um dos rotores. A cada rotor, está associada uma força de sustentação, um binário e uma força de resistência em torno do seu eixo de rotação. Em consequência, é regra geral colocar os rotores opostos, um e dois, a rodar num determinado sentido, e os rotores três e quatro a rodar no sentido contrário (Figura 7). Desta forma, é possível compensar os binários resistentes envolvidos, forçando a aceleração angular do Quadrirotor (perpendicular ao plano da Figura 7) a zero. Assim, é comprovado que não é necessário a existência de um rotor de cauda como nos helicópteros convencionais para obter a estabilização da guinada, contribuindo todos os rotores neste caso para a força de sustentação.

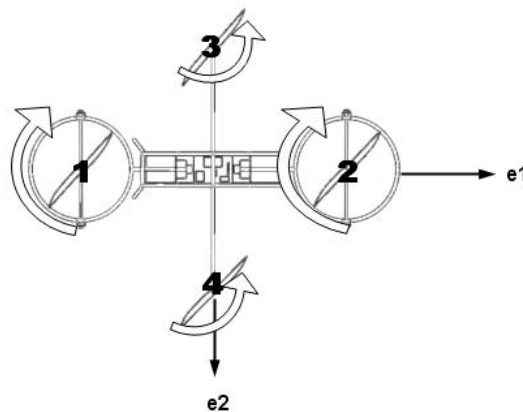


Figura 7: ALIV - Nomenclatura

No entanto, se a ideia for introduzir alguma guinada no UAV, é necessário aplicar um diferencial de rotação entre os pares de rotores, mantendo-se a força de sustentação total constante (situação (a) da Figura 8). Todos os outros movimentos do Quadrirotor processam-se de uma forma semelhante.

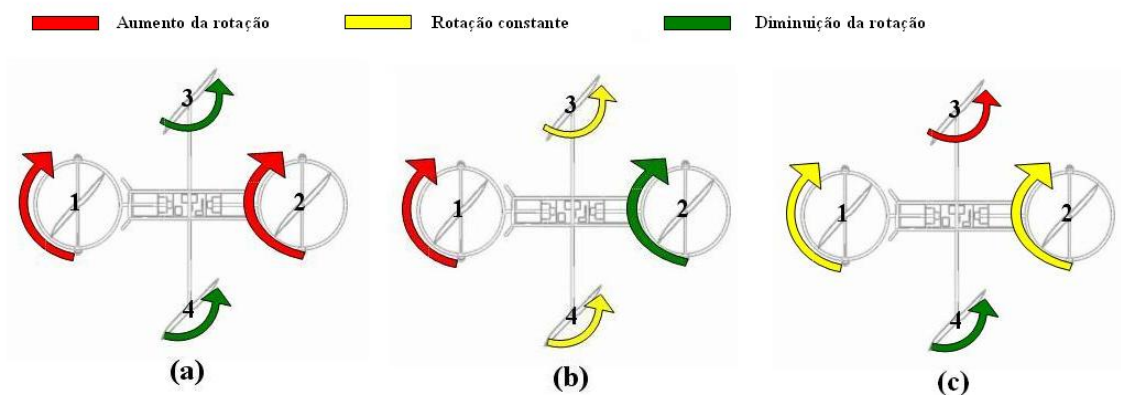


Figura 8: Relação de rotação para os movimentos do Quadrirotor

Assim, se se quiser introduzir um movimento de picada, e a consequente deslocação do Quadrirotor

segundo a direcção  $e_1$ , é necessário aplicar um diferencial na rotação do par de rotores um e dois, mantendo a força de sustentação do par constante (situação (b) da Figura 8), enquanto se se quiser aplicar um movimento de rolamento, e a consequente deslocação do Quadrirotor segundo a direcção  $e_2$ , o diferencial deve ser aplicado ao par de rotores três e quatro, mantendo mais uma vez constante a força de sustentação criada por esse par (situação (c) da Figura 8) [3].

A Tabela 1 resume a teoria atrás avançada.

Motores	1	2	3	4	
$\phi$	=	=	+	-	$\Rightarrow$ Deslocamento positivo segundo $e_2$
$\theta$	+	-	=	=	$\Rightarrow$ Deslocamento positivo segundo $e_1$
$\psi$	+	+	-	-	$\Rightarrow$ Rotação anti-horária
h	+	+	+	+	$\Rightarrow$ Deslocamento negativo segundo $e_3$

Tabela 1: Diferencial de rotação aplicado aos rotores para obtenção dos movimentos desejados

O controlo e o movimento do Quadrirotor está simplesmente associado à velocidade de rotação de cada um dos rotores, devido ao binário resistivo que actua nas pás do rotor na direcção oposta à sua rotação. É este o princípio básico que rege a dinâmica deste sistema. Construir o modelo matemático para o tipo de movimento característico de um Quadrirotor, é assim o primeiro passo a ser realizado na análise e obtenção do sistema de controlo necessário.

## 2.2 Plataforma ALIV

O ALIV (Figura 9) é um Quadrirotor inovador em relação aos modelos de Quadrirotores existentes. No entanto, de uma forma geral, o ALIV é parecido aos chamados Quadrirotores convencionais até aqui referidos. Este é constituído por quatro rotores nas extremidades de uma estrutura cruzada, e embarca todo o equipamento a bordo no centro da sua estrutura, o mais centrado e alinhado possível com o seu centro de gravidade. Sendo assim, o ALIV continua a usufruir das qualidades e características inerentes a um UAV deste tipo. Entre estas podemos destacar a facilidade de construção comparativamente a UAVs genéricos de asa fixa, e do método típico de funcionamento operacional em que todos os seus rotores contribuem para a força propulsiva, ao contrário do sucedido em UAVs de asa rotativa típicos como o são os helicópteros convencionais.

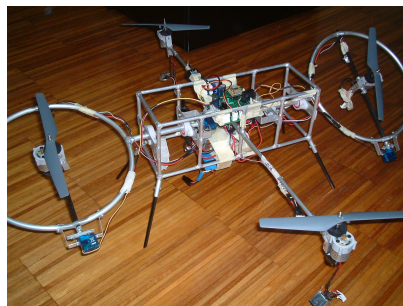


Figura 9: ALIV

Um Quadrirotor convencional é caracterizado por ter os quatro rotores fixos, isto é, todos os movimentos de translação são criados através do diferencial de rotação dos pares de rotores existentes (Secção 2.1). Quanto ao ALIV, para além destas capacidades, a possibilidade de rotação do eixo de



apoio de um dos seus pares de rotores possibilita também algumas destas deslocações transversais. Este tipo de conceito torna de facto o ALIV inovador em relação aos outros modelos existentes, fazendo-o conseguir realizar manobras de maior complexidade ou atingir posições estacionárias diferentes das convencionais.

Nesta secção será dado a conhecer com algum pormenor o ALIV, o Quadrirotor adoptado como referência para todo o trabalho desenvolvido. É por isso importante fazer uma descrição do seu modo de funcionamento, tanto em termos de graus de liberdade (Secção 2.2.1) como na descrição de uma forma sucinta e adequada do seu equipamento a bordo (Secção 2.2.2) que serão relevantes na fase de construção da dinâmica do problema e do próprio controlador.

### 2.2.1 Funcionamento geral do ALIV

Actualmente, o ALIV é pilotado através de dois joysticks que tentam realizar e controlar todo o tipo de movimentos de translação ou rotação. O presente capítulo serve para ilustrar e demonstrar de uma forma idealizada esse tipo de movimentos.

De uma forma geral, um dos joysticks é responsável pelos movimentos de translação, guinada e controlo de potência, e o outro pelos movimentos de rotação. Em relação ao primeiro joystick, ao botão “*thrust*” está associado o controlo de potência integral, isto é, um controlo idêntico de potência a cada um dos rotores, controlo da altitude. É portanto necessário um constante ajuste de potência transmitida aos motores para que estes mantenham o Quadrirotor a uma dada altitude e igualem a força gravítica causada pelo peso do corpo quando este efectua uma deslocação transversal.

Em termos da translação lateral segundo  $e_2$  para a esquerda ou para a direita, esta pode ser obtida através da rotação do conjunto motor mais hélice segundo o seu eixo transversal  $e_1$  (Figura 10).



Figura 10: ALIV - Translação lateral

Este tipo de movimento é obtido com a deslocação do primeiro joystick para a esquerda ou para a direita e constitui a parte inovadora em termos de movimentos associados aos Quadrirotores convencionais.

Para uma translação frontal segundo  $e_1$ , para a frente ou para trás, o método é em tudo semelhante. O controlo é efectuado deslocando mais uma vez o primeiro joystick mas desta vez segundo o outro eixo. O resultado consiste na rotação do conjunto motor mais hélice segundo o seu eixo transversal  $e_2$  (Figura 11).



Figura 11: ALIV - Translação frontal

Através do primeiro joystick ainda é possível obter um controlo direccionado através da rotação do ALIV em torno do seu eixo vertical, movimento de guinada. Para isso, roda-se o joystick para a esquerda ou para a direita consoante a guinada pretendida, o que equivale a uma rotação oposta segundo o eixo transversal  $e_1$  do conjunto motor mais hélice entre o mesmo par de rotores, Figura 12.



Figura 12: ALIV - Movimento de guinada

Ao segundo joystick estão associados os movimentos de rotação. Estes movimentos são causados pelo diferencial de rotação entre o mesmo par de rotores como explicado na Secção 2.1 que caracteriza o tipo de funcionamento de um Quadrirotor convencional. A este tipo de movimentos, de uma forma idealizada, não é necessário aplicar um ajuste na potência uma vez que a força vertical gerada pelo conjunto dos quatro rotores, força propulsiva, se mantém constante.

### 2.2.2 Equipamento a bordo do ALIV

De modo a cumprir os requisitos de pilotagem e futuramente de capacidade de voo autónomo, o ALIV, à semelhança de outros UAVs, possui a bordo diferentes tipos de equipamento mecânico e electrónico. Nesta secção pretende-se identificar algum desse equipamento mais relevante bem como a sua interligação.

Analisando o ALIV numa óptica mais estrutural e mecânica, o mesmo é construído sobre uma estrutura de tubos cilíndricos em alumínio com aproximadamente  $6\text{ mm}$  de diâmetro que servem de apoio a todo o material embarcado. Tal como tem vindo a ser descrito, o mesmo possui quatro conjuntos motor mais hélice responsáveis pela geração da sua dinâmica e controlados através de quatro servos GWS Pico. Actualmente o motor utilizado é o GWS 350 e as hélices são idênticas ao modelo X-UFO com aproximadamente  $19\text{ cm}$  de diâmetro. Estes conjuntos são alimentados através de uma bateria de lítio de  $11.1\text{ V}$  e  $1.7\text{ Ah}$ . O ALIV possui ainda uma bateria auxiliar para alimentação do microcontrolador e dos servos, de NiMh  $12\text{ V}$ ,  $0.8\text{ Ah}$ . Do ponto de vista electrónico, o ALIV é composto por um emissor e placa de comando actual (Figura 13, cedida pelo Eng. Severino Raposo)

responsável pela emissão do sinal de controlo via rádio. Através do controlo por rádio FM com transceiver é possível controlar e ler os sensores internos do ALIV a uma amostragem de cerca de 8 amostras por segundo com possibilidade de extensão para 16 amostras por segundo.

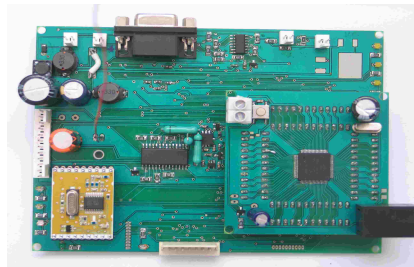


Figura 13: Emissor

Em termos de sensores, existem três acelerómetros para cada um dos eixos ortogonais,  $(e_1, e_2, e_3)$ , que são os LIS3L06AL com capacidade leitura de  $\pm 2g$ , extensível até  $\pm 6g$  da aceleração global num dado ponto, e os sensores de rotação, os Axis Magneto-Inductive Sensor Driver que funcionam como giroscópios para leitura da atitude. As leituras efectuadas por ambos os sensores são processadas num microprocessador. Existem ainda sensores da corrente total e da tensão às baterias. Para a recepção dos dados emitidos pelo emissor, é utilizado um receptor, ilustrado na Figura 14, cedida pelo Eng. Severino Raposo. Este receptor é constituído por quatro saídas em PWM de 10bits para os quatro motores DC, cada uma com  $10A$  a  $20A$  de corrente máxima e  $15V$  máximo na saída.

O sistema tem 3 LEDs de medição de potência de sinal de rádio e 3 LEDs para a potência da bateria, um canal livre ADC, um porto digital I/O de 8 bits e sete linhas I/O adicionais controláveis através dos dois joysticks.

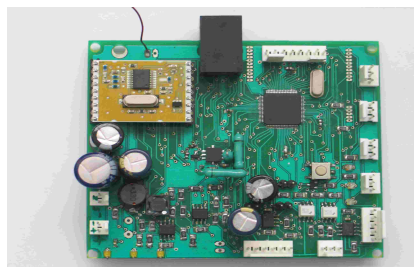


Figura 14: Receptor

### 2.3 Plataforma considerada no estudo

O Quadrirotor é um UAV caracterizado pela sua simplicidade de funcionamento mas em contrapartida por instabilidades que o levam a ser de difícil pilotagem. O ALIV, partindo do mesmo princípio, apresenta dificuldades acrescidas pela adição de graus de liberdade suplementares. Foi principalmente por esta razão, que se optou por considerar inicialmente as características do ALIV actuando como se de um Quadrirotor convencional se tratasse, acabando por desprezar os graus de liberdade suplementares. Assim, como referido nos objectivos, adoptando esta simplificação o trabalho tornou-se mais acessível, não deixando de ser um bom ponto de partida para a futuros trabalhos.

### 3 Modelação matemática do problema

#### 3.1 Modelação da dinâmica

Para a elaboração da dinâmica foram assumidos dois referenciais. O primeiro referencial é um referencial considerado inercial e fixo, ligado à superfície da Terra com os dois primeiros eixos tangentes ao meridiano e ao paralelo, referencial NED (North-East-Down). O segundo referencial é um referencial móvel, ligado ao corpo rígido do Quadrirotor com origem no centro de gravidade, será denominado ABC (Aircraft Body Centered), Figura 15. Ambos os referenciais possuem as direcções  $e_z$  e  $e_3$  quando estabilizados a apontar para o centro da Terra [17].

- Referencial NED (North-East-Down),  $R_{NED} = [e_x \ e_y \ e_z]$
- Referencial ABC (Aircraft Body Centered),  $R_{ABC} = [e_1 \ e_2 \ e_3]$

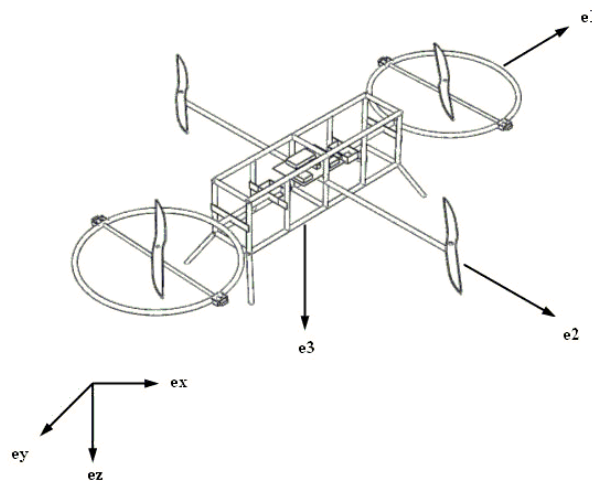


Figura 15: Referenciais: NED e ABC

O Quadrirotor possui seis graus de liberdade definidos por doze estados. Seis dos doze estados controlam a atitude do sistema, isto é, os ângulos de Euler ( $\phi \ \theta \ \psi$ ) e as velocidades angulares no referencial ABC,  $R_{ABC}$ , ( $P \ Q \ R$ ). Os outros seis estados controlam tanto a posição do centro de gravidade do Quadrirotor ( $X \ Y \ Z$ ), como a sua velocidade linear ( $U \ V \ W$ ) em termos do referencial inercial NED,  $R_{NED}$ .

Para se proceder à derivação dos estados entre estes referenciais, torna-se assim necessário obter a matriz de rotação entre os dois sistemas de coordenadas. A matriz de rotação global de NED para ABC resulta da sequência de três rotações, isto é:

$$S_E = S_\phi S_\theta S_\psi \quad (1)$$

o que leva a<sup>2</sup>:

---

<sup>2</sup>A dedução pormenorizada das equações e matrizes apresentadas nesta secção, podem ser consultadas ao longo do Apêndice A.

$$S_E = \begin{bmatrix} \cos \psi \cos \theta & \sin \psi \cos \theta & -\sin \theta \\ \cos \psi \sin \theta \sin \phi - \cos \phi \cdot \sin \psi & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \cos \theta \sin \phi \\ \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi & \sin \psi \sin \theta \cos \phi - \sin \phi \cos \psi & \cos \theta \cos \phi \end{bmatrix} \quad (2)$$

O modelo matemático é baseado nas equações do movimento newtoniano e o referencial ABC,  $R_{ABC}$ , é considerado o referencial de trabalho porque as unidades sensoriais identificam as componentes neste mesmo referencial. Para o movimento do corpo rígido foi considerada a segunda lei de Newton que permite obter as equações da dinâmica do Quadrirotor expressas no referencial local ABC,  $R_{ABC}$ .

$$F_T = \frac{d}{dt} [mV_T]_B + \omega \times [mV_T]_B \quad (3)$$

$$M_T = \frac{d}{dt} [I\omega]_B + \omega \times [I\omega]_B \quad (4)$$

em que  $V_T$  é o vector com as velocidades lineares ( $U \ V \ W$ ) e  $\omega$  o vector com as velocidades angulares ( $P \ Q \ R$ ). Considerando que a força por acção da gravidade é uma constante no referencial inercial e pode ser expressa no referencial local,  $R_{ABC}$ , através de uma transformação de Euler:

$$F_g = mS_E \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T = m \begin{bmatrix} -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix}_B^T g \quad (5)$$

e que a força total exercida no Quadrirotor,  $F_T$ , é constituída pela soma da força gravítica,  $F_g$ , e da força de propulsão,  $F_p$ , gerada pela rotação de cada um dos rotores, a expansão da Equação 3 conduz a:

$$\begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} F_{px} \\ F_{py} \\ F_{pz} \end{bmatrix} + g \begin{bmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{bmatrix} - \begin{bmatrix} QW - RV \\ RU - PW \\ PV - QU \end{bmatrix} \quad (6)$$

Analisando agora a Equação 4 e considerando a massa e inércia do Quadrirotor constantes, com  $I_{yz} = I_{xy} = I_{xz} = 0$  a matriz de inércia torna-se numa matriz principal e a expansão da Equação 4 conduz a:

$$M_T = \begin{bmatrix} I_{11}\dot{P} \\ I_{22}\dot{Q} \\ I_{33}\dot{R} \end{bmatrix} + \begin{bmatrix} (I_{33} - I_{22})RQ \\ (I_{11} - I_{33})QP \\ (I_{22} - I_{11})PQ \end{bmatrix} \quad (7)$$

onde  $M_T$  corresponde ao vector dos momentos aplicados ao Quadrirotor,  $M_T = (M_x \ M_y \ M_z)$

## 3.2 Modelação da cinemática

### 3.2.1 Ângulos de Euler

Com base na cinemática do problema, e assumindo que a distância do centro do referencial NED ao centro de gravidade do Quadrirotor (origem do referencial,  $R_{ABC}$ ) é dada por  $r$ , então o vector posição  $\vec{r}$  do Quadrirotor referente ao referencial inercial,  $R_{NED}$ , pode ser escrito como:

$$\vec{r} = X \vec{i} + Y \vec{j} + Z \vec{k} \quad (8)$$

o que leva a que a velocidade absoluta seja:

$$\dot{\vec{r}} = \dot{X} \vec{i} + \dot{Y} \vec{j} + \dot{Z} \vec{k} \quad (9)$$

Sendo assim as componentes da velocidade absoluta em termos do referencial inercial ( $\dot{X}$   $\dot{Y}$   $\dot{Z}$ ) podem ser obtidas através das velocidades no referencial local ( $U$   $V$   $W$ ) através da forma:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = S_E^T \begin{bmatrix} U \\ V \\ W \end{bmatrix} \quad (10)$$

É possível obter a velocidade angular em termos dos ângulos de Euler, no referencial inercial  $R_{NED}$ , com base nas velocidades de rotação ( $P$   $Q$   $R$ ) do referencial local  $R_{ABC}$ . Para isso basta realizar a seguinte transformação matemática:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \quad (11)$$

Em que  $T$  é igual a:

$$T = \begin{bmatrix} 1 & \tan \theta \sin \phi & \tan \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sec \theta \sin \phi & \sec \theta \cos \phi \end{bmatrix} \quad (12)$$

### 3.2.2 Quaterniões

Para além do método tradicional de representação da orientação correspondente à rotação de um sistema de coordenadas de uma aeronave com base nos ângulos de Euler, existem outros métodos que usam quatro ou mais variáveis com o mesmo propósito. O principal objectivo destes sistemas é eliminar as singularidades existentes na representação da rotação em termos de ângulos de Euler associadas às funções trigonométricas, bem como incrementar o tempo de processamento computacional durante a fase dos cálculos de navegação. De entre os modelos alternativos existentes e desenvolvidos até ao momento, o método dos Quaterniões, usando quatro variáveis, é aquele que actualmente é mais utilizado [16].

Para a utilização deste método são definidas quatro variáveis, os quaterniões sendo:

$$\begin{cases} q_0 = \cos \frac{\delta}{2} \\ q_1 = \cos \alpha \sin \frac{\delta}{2} \\ q_2 = \cos \beta \sin \frac{\delta}{2} \\ q_3 = \cos \gamma \sin \frac{\delta}{2} \end{cases} \quad (13)$$

e a matriz de rotação global  $S_q$ :

$$S_q = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (14)$$

No entanto os parâmetros  $q_i$  podem ser representados em termos de ângulos de Euler:

$$\begin{cases} q_0 = \pm (\cos(\phi/2) \cos(\theta/2) \cos(\psi/2) + \sin(\phi/2) \sin(\theta/2) \sin(\psi/2)) \\ q_1 = \pm (\sin(\phi/2) \cos(\theta/2) \cos(\psi/2) - \cos(\phi/2) \sin(\theta/2) \sin(\psi/2)) \\ q_2 = \pm (\cos(\phi/2) \sin(\theta/2) \cos(\psi/2) + \sin(\phi/2) \cos(\theta/2) \sin(\psi/2)) \\ q_3 = \pm (\cos(\phi/2) \cos(\theta/2) \sin(\psi/2) - \sin(\phi/2) \sin(\theta/2) \cos(\psi/2)) \end{cases} \quad (15)$$

em que a escolha do sinal é arbitrária mas deverá ser consistente.

É ainda possível obter as equações da cinemática equivalentes em termos de Quaterniões. De entre um dos vários métodos, destaca-se o de Whittaker [16] em que partindo das equações das velocidades angulares em termos de quaterniões:

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \quad (16)$$

e verificando que a matriz que lhe é característica é ortogonal e quando invertida as equações são bi-lineares em termos de  $q_i$  e velocidades angulares, podemos obter a derivada dos quaterniões como sendo [16]:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = -\frac{1}{2} \begin{bmatrix} 0 & P & Q & R \\ -P & 0 & -R & Q \\ -Q & R & 0 & -P \\ -R & -Q & P & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = -\frac{1}{2} \Omega_q \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (17)$$

É com base nesta última equação e na nova matriz de rotação global  $S_q$ , que partindo dos valores iniciais dos ângulos de Euler, os quaterniões são obtidos (Equação 15) e os cálculos durante a fase de navegação são iniciados. As Equações 18, 19, 20 e 21 constituem assim o modelo completo da dinâmica a ser usado no cálculo da trajectória e da atitude de uma aeronave com base no sistema de coordenadas NED.

$$\begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} F_{px} \\ F_{py} \\ F_{pz} \end{bmatrix} + g \begin{bmatrix} 2(q_1q_3 - q_0q_2) \\ 2(q_2q_3 + q_0q_1) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} - \begin{bmatrix} QW - RV \\ RU - PW \\ PV - QU \end{bmatrix} \quad (18)$$

$$\begin{bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix} = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} \frac{1}{I} - \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \times I \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \quad (19)$$

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = -\frac{1}{2}\Omega_q \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (20)$$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = S_q^T \begin{bmatrix} U \\ V \\ W \end{bmatrix} \quad (21)$$

Dado que estas equações não são lineares, estas terão de ser primeiramente linearizadas e usadas em espaço de estados para se proceder à construção do sistema de controlo moderno (Secção 4.1) para as futuras simulações.

### 3.3 Modelação do conjunto motor mais hélice

A única força propulsiva existente no Quadr rotor é gerada pela rotação das hélices. É por isso de esperar que a escolha do conjunto motor mais hélice seja de extrema importância. De uma forma geral, a eficiência das pás está associada ao motor que a propulsiona, e a eficiência desse conjunto está associada à estrutura que se pretende operar [18]. Isto revela a dificuldade em projectar um conjunto constituído por um motor mais uma hélice ideal para este tipo de UAV, dada a sua importância acrescida relativamente à atribuição dos graus de liberdade do mesmo. Partindo deste pressuposto, para a modelação deste conjunto foram assumidos modelos de referência mais ou menos comuns para um UAV com estas características. Numa situação deste tipo, estamos interessados em calcular a força e os momentos gerados por um dado conjunto motor mais hélice através da tensão aplicada a cada um dos motores e poder relacioná-la se possível com a rotação envolvida no conjunto. Como tal foi assumido o andamento característico de um motor em termos de rotação sofrida por tensão aplicada. Este andamento engloba uma zona morta em que a velocidade de rotação é nula, e uma rotação limite (saturação) que não permite que a velocidade exceda a rotação máxima, Figura 16.

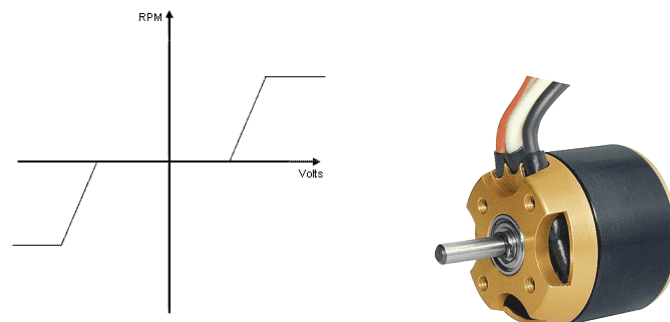


Figura 16: Funcionamento genérico de um motor eléctrico de corrente contínua



Um problema que ocorre com os motores eléctricos é que apesar de serem idênticos, não se comportam exactamente da mesma forma. Isto faz com que se tivesse de realizar ensaios experimentais a cada um dos quatro conjuntos motor mais hélice, e através por exemplo de um sistema massa mola obter uma equação que permitisse relacionar a tensão aplicada com a força gerada. Esta situação acabou por não se realizar em relação ao conjunto utilizado no ALIV, e como tal, teve de se criar um modelo matemático, que apesar das suas limitações, pudesse de certa forma modelar com algum realismo o conjunto.

Para isso foi assumido o motor, AXI 2212-12 (Figura 16). Este motor é comum para utilizações semelhantes às pretendidas e os valores padrão são apresentados na Tabela 2:

RPM/V	2000
Gama entrada [V]	9-12
RPM máximo	6000
Thrust [g]	350
$P_{out}$ [W]	28
$P_{in}$ [W]	56

Tabela 2: Características do motor AXI 2212-12

Com base nos dados da Tabela 2 e na curva avançada na Figura 16, foi possível caracterizar a gama de funcionamento do motor e a relação existente entre a tensão aplicada e a rotação gerada, com a constante do motor:

$$2000rpm = \frac{2000 \times 2\pi}{60} rad/s = K_v$$

$$\begin{cases} \omega_m = 0 & |V| \leq V_0 \\ \omega_m = K_v (V - V_0) = 2000 (V - 9) = \frac{2000 \times 2\pi}{60} (V - 9) & |V| > V_0 \end{cases} \quad (22)$$

O funcionamento do motor foi simulado usando o SIMULINK e os blocos existentes para o efeito são apresentados na Figura 17.

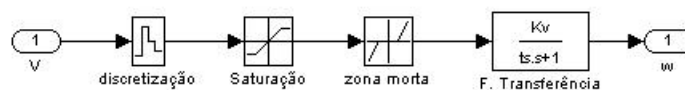


Figura 17: Actuador

O actuador foi construído com base na tensão aplicada ao motor tendo o mesmo uma constante de tempo de sistema de primeira ordem de  $t_s = 20 ms$  e uma sensibilidade  $K_v$ .

Para a modelação da hélice foi utilizada, segundo o mesmo princípio do motor, a hélice 9x4,5 APCE, Figura 18. É possível modelar uma hélice através de três dos seus parâmetros característicos,  $C_t$ ,  $C_p$ , e  $r$  [18]. Os primeiros dois são o coeficiente de impulso (*thrust*) e de potência (*power*) e são obtidos experimentalmente podendo ser consultados com base em tabelas características da hélice consoante o valor das velocidades envolvidas, e o último o comprimento da pá da hélice. Os valores característicos desta hélice podem ser consultados na Tabela 3, onde  $C_t$  e  $C_p$  equivalem aos

valores experimentais correspondentes a gamas muito baixas, dadas as velocidades associadas aos movimentos do ALIV.

Raio da hélice [m]	0.1229
Coefficiente de impulso ( <i>thrust</i> ) $C_t$	0.1047
Coefficiente de potência ( <i>power</i> ) $C_p$	0.0374

Tabela 3: Características da hélice  $9 \times 4,5$  APCE

Com base nesta informação, é possível construir um modelo matemático característico deste conjunto através das seguintes expressões [18]:

$$F_p = \frac{4\rho r^4 C_t}{\pi^2} \omega_m^2 \quad (23)$$

$$M_T = -\frac{P}{\omega_m} = -\frac{4\rho r^5 C_p}{\pi^3} \omega_m^2 \quad (24)$$



Figura 18: Hélice  $9 \times 4,5$  APCE

Daqui se retira que tanto a força como os momentos gerados variam apenas com a rotação aplicada ao conjunto, podendo as equações anteriores serem escritas como:

$$F_p = K_f \omega_m^2, \quad K_f = \frac{4\rho r^4 C_t}{\pi^2} = \frac{4 \times 1.2 \times (0.2458/2)^4 \times 0.1047}{\pi^2} = 1.1617 \times 10^{-5} \text{ kg m/rad}^2 \quad (25)$$

$$M_T = K_m \omega_m^2, \quad K_m = \frac{4\rho r^5 C_p}{\pi^3} = \frac{4 \times 1.2 \times (0.2458/2)^5 \times 0.0374}{\pi^3} = 1.6234 \times 10^{-7} \text{ kg m}^2/\text{rad}^2 \quad (26)$$

$$K_K = \frac{r \times C_p}{\pi \times C_t} = 0.0140 \text{ m}, \quad M_p = K_K F_p \quad (27)$$

onde  $K_f$ ,  $K_m$  e  $K_K$  são constantes características da modelação para a hélice e o motor adoptados.

### 3.4 Peso e momentos de inércia

Um aspecto fulcral na performance de um Quadrirotor, bem como de qualquer UAV, reside no peso da estrutura e do conjunto de equipamento embarcado. Um Quadrirotor caracteriza-se por transportar a maior parte do seu equipamento numa secção central o mais perto possível do centro de gravidade desejado. É necessário por isso, que o equipamento seja não só leve como de pequenas dimensões. Isto faz com que o preço desse equipamento aumente. O espaço também não é ilimitado,

e por isso a relação entre equipamento mais relevante e custos envolvidos é algo inevitável. Se por um lado queremos mais sensores para ter maior instrumentação, por outro, devido ao espaço, os mesmos têm de ser mais pequenos e por isso mais caros. Este é o compromisso assumido num projecto deste tipo. Para a elaboração deste trabalho foi assumido o peso total do ALIV. Este peso engloba todo o material envolvido na descrição do equipamento a bordo do ALIV (Secção 2.2.2), resultando num peso total de 1.2 kg como é demonstrado com base nos pesos apresentados na Tabela 4.

Bateria de Lítio	160
Circuito Receptor de Comunicações	50
4 Servos GWS Pico	40
4 Motores DC GWS 350	260
Bateria Auxiliar	90
Cabos Eléctricos	50
4 Rolamentos	40
Interruptores ON/OFF	20
Estrutura de Alumínio	490
Total [g]	1200

Tabela 4: Peso dos componentes a bordo do ALIV

Em relação aos momentos de inércia, foram desprezados os efeitos causados não só pelo equipamento como pela própria estrutura em si, e a matriz inercial foi considerada diagonal principal. Este tipo de aproximações é uma técnica comum em qualquer tipo de abordagem referenciada neste trabalho, dada a estrutura particular em forma de cruz de um Quadrirotor. Como tal foram assumidos para o cálculo dos momentos de inércia, os motores e a sua localização dentro da estrutura. Foi definido:

- $m_m$ - Massa do motor
- $x_m$ - Comprimento do motor ao longo do eixo  $x$
- $y_m$ - Comprimento do motor ao longo do eixo  $y$
- $z_m$ - Comprimento do motor ao longo do eixo  $z$
- $l_l$ - Distância do centro de gravidade ao centro dos motores 1 e 2
- $l_c$ - Distância do centro de gravidade ao centro dos motores 3 e 4

Para o momento de inércia  $I_{xx}$  obtém-se:

$$\begin{cases} I_{x_1} = I_{x_2} = \frac{1}{12}m_m (y_m^2 + z_m^2) \\ I_{x_1} = \frac{1}{12} \times 0.060((15 \times 10^{-3})^2 + (32 \times 10^{-3})^2) = 6.245 \times 10^{-6} \text{ kg m}^2 \\ I_{x_3} = I_{x_4} = \frac{1}{12}m_m (y_m^2 + z_m^2) + m_m l_c^2 \\ I_{x_3} = \frac{1}{12} \times 0.060((15 \times 10^{-3})^2 + (32 \times 10^{-3})^2) + 0.060 \times 0.64^2 = 0.0246 \text{ kg m}^2 \end{cases} \quad (28)$$

$$I_{xx} = 2 \times I_{x_1} + 2 \times I_{x_3} = 2 \times (6.245 \times 10^{-6}) + 2 \times 0.0246 = 0.0492 \text{ kg m}^2 \quad (29)$$

Para o momento de inércia  $I_{yy}$  obtém-se:

$$\begin{cases} I_{y_1} = I_{y_2} = \frac{1}{12}m_m(x_m^2 + z_m^2) + m_m l_l^2 \\ I_{y_1} = \frac{1}{12} \times 0.060((15 \times 10^{-3})^2 + (32 \times 10^{-3})^2) + 0.060 \times 0.60^2 = 0.0216 \text{ kg m}^2 \\ I_{y_3} = I_{y_4} = \frac{1}{12}m_m(x_m^2 + z_m^2) \\ I_{y_3} = \frac{1}{12} \times 0.060((15 \times 10^{-3})^2 + (32 \times 10^{-3})^2) = 6.245 \times 10^{-6} \text{ kg m}^2 \end{cases} \quad (30)$$

$$I_{yy} = 2 \times I_{y_1} + 2 \times I_{y_3} = 2 \times 0.0216 + 2 \times (6.245 \times 10^{-6}) = 0.0432 \text{ kg m}^2 \quad (31)$$

Para o momento de inércia  $I_{zz}$  obtém-se:

$$\begin{cases} I_{z_1} = I_{z_2} = \frac{1}{12}m_m(x_m^2 + z_m^2) + m_m l_l^2 \\ I_{z_1} = \frac{1}{12} \times 0.060((15 \times 10^{-3})^2 + (32 \times 10^{-3})^2) + 0.060 \times 0.64^2 = 0.0216 \text{ kg m}^2 \\ I_{z_3} = I_{z_4} = \frac{1}{12}m_m(x_m^2 + z_m^2) + m_m l_c^2 \\ I_{z_3} = \frac{1}{12} \times 0.060((15 \times 10^{-3})^2 + (32 \times 10^{-3})^2) + 0.060 \times 0.60^2 = 0.0246 \text{ kg m}^2 \end{cases} \quad (32)$$

$$I_{zz} = 2 \times I_{z_1} + 2 \times I_{z_3} = 2 \times 0.0216 + 2 \times 0.0246 = 0.0924 \text{ kg m}^2 \quad (33)$$

O que resulta na matriz inercial:

$$I = \begin{bmatrix} 0.0492 & 0 & 0 \\ 0 & 0.0432 & 0 \\ 0 & 0 & 0.0924 \end{bmatrix} \text{ kg m}^2 \quad (34)$$

### 3.5 Sensores

Com base no que tem vindo a ser descrito, torna-se evidente que para o aumento do realismo da simulação e uma futura implementação para testes em ambientes reais, é de extrema importância fazer uso do equipamento a bordo do ALIV. O ALIV tem a bordo um conjunto de três acelerómetros para medição da aceleração global num dado ponto e um magnetómetro (Secção 2.2.2) para o cálculo da atitude, onde através dos quais, é possível calcular a estimativa dos estados envolvidos no controlo da atitude. Para que estes possam ser simulados com realismo, é necessário realizar uma modelação matemática dos mesmos, e com isso observar quantos estados estamos em condições de recuperar.

#### 3.5.1 Acelerómetros

Um acelerómetro é um sensor que mede a aceleração absoluta no seu referencial. Essa aceleração é medida no ponto em que o acelerómetro se encontra, por isso de modo a aumentar a sua eficácia é de extrema importância aproximá-lo o mais possível do centro de gravidade do corpo. No caso do ALIV, o acelerómetro encontra-se 5 mm para a frente e 30 mm para cima do seu centro de gravidade. Um acelerómetro num determinado ponto  $P$  mede a componente de aceleração  $a'_p$  dada por [16]:

$$a'_p = a_p - g \quad (35)$$

onde  $a_p$  e  $g$  são respectivamente a aceleração absoluta e a força gravítica no ponto  $P$ . Assumindo que  $r$ , é o vector que liga o ponto de medição do acelerómetro ao centro de gravidade e que o Quadrirotor tem velocidade angular,  $\omega$ , em relação ao referencial inercial  $R_{NED}$ , então a aceleração absoluta para um dado ponto  $P$  pode ser escrita como<sup>3</sup>[16]:

$$a_p = \frac{F_{ABC} + S_E mg}{m} + \dot{\omega}_{ABC} \times r + \omega_{ABC} \times (\omega_{ABC} \times r) \quad (36)$$

Esta equação é aplicada na modelação dos três acelerómetros, um para cada eixo sendo  $a_p = [a_{px} \ a_{py} \ a_{pz}]$  faltando adicionar o ruído e os parâmetros de ordem eléctrica que limitam a gama de trabalho do acelerómetro. Neste caso foram usados os parâmetros definidos no *datasheet* do acelerómetro do ALIV, cedido pelo Eng. Severino Raposo (Tabela 5).

Gama de leitura	+/-2g
Número de bits	10
Resolução	0.0383g

Tabela 5: Limitações dos Acelerómetros utilizados

A resolução foi calculada com base na seguinte definição:

$$\Delta y = \frac{y_{max} - y_{min}}{2^N} \quad (37)$$

em que  $y_{max}$  corresponde à gama de leitura máxima e  $N$  ao número de bits.

### 3.5.2 Magnetómetro

Os magnetómetros são aparelhos que em termos aeronáuticos são úteis no controlo da atitude das aeronaves. Geralmente são dispositivos que medem a magnitude e direcção do campo magnético, estático ou alternado, baseando-se nas propriedades de saturação de ligas metálicas. Devido à sua complexidade, optou-se por fazer a implementação de um modelo matemático mais simples de modo a puder dar continuidade ao problema.

Assim, foi admitido que o magnetómetro funciona com base na direcção e sentido indicados por uma agulha inicialmente calibrada a apontar para o norte magnético, magnetómetro tri-axial. O magnetómetro tem assim como saídas a orientação instantânea da agulha em termos do referencial inercial consoante a atitude do Quadrirotor. Este princípio pode ser modelado através da seguinte equação matemática:

$$\begin{bmatrix} N_{mag} \\ E_{mag} \\ D_{mag} \end{bmatrix} = S_E \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (38)$$

em que o vector  $(N_{mag}, E_{mag}, D_{mag})$  corresponde à projecção do vector unitário inicialmente a apontar para norte, e  $S$  à matriz de rotação global definida na Equação 2.

<sup>3</sup>Por motivo de interesse e de espaço, a dedução da Equação 36 foi deixada para consulta no Apêndice B

### 3.5.3 Câmara embarcada

De todos os sensores modelados, este é o único que não está actualmente presente no ALIV. Foi concebido de forma a permitir um controlo da posição com base na visualização de determinados pontos de referência através de uma câmara embarcada. Isto permite, construir simulações mais abrangentes, tornando o ALIV mais autónomo e versátil.

Para a modelação da câmara embarcada foram considerados três referenciais:

- Referencial NED (North-East-Down),  $R_{NED} = [e_x \ e_y \ e_z]$
- Referencial ABC (Aircraft Body Centered),  $R_{ABC} = [e_1 \ e_2 \ e_3]$
- Referencial da imagem xyz,  $R_{xyz} = [x \ y \ z]$

Com base na posição dos pontos de referência no referencial NED, e da posição do Quadrirotor em relação ao mesmo, através das características da câmara modelada é possível projectar os pontos de referência no referencial da imagem,  $R_{xyz}$ . Para isso, para além da utilização da matriz  $S$  que define a rotação entre o referencial inercial,  $R_{NED}$ , e o móvel,  $R_{ABC}$ , é necessário uma nova matriz  $S_{cam}$  que define a passagem entre este último e o referencial de imagem,  $R_{xyz}$ . A matriz  $S$  neste caso é igual à matriz  $S_E$  apresentada na Secção 3.1 e demonstrada no Apêndice A. Quanto à matriz  $S_{cam}$  é baseada no mesmo princípio, e como tal também é igual à matriz  $S$ , so que desta vez,  $(\phi \ \theta \ \psi)$  correspondem à orientação da câmara em relação ao referencial móvel, isto é,  $(\phi_{cam} \ \theta_{cam} \ \psi_{cam})$ . Assim, a matriz total de passagem do referencial inercial,  $R_{NED}$ , para o da imagem,  $R_{xyz}$ , pode ser dada por:

$$S_T = S_{cam}S \quad (39)$$

em que a nova atitude pode ser retirada da matriz  $S_T$  através das seguintes equações:

$$\begin{cases} \phi_T = -\text{atan2}(S_T(2,3), S_T(3,3)) \\ \theta_T = -\arcsin\left(\frac{S_T(1,3)}{\det(S_T)}\right) = -\arcsin(S_T(1,3)) \\ \psi_T = -\text{atan2}(S_T(1,2), S_T(1,1)) \end{cases} \quad (40)$$

e a posição da câmara em relação ao referencial inercial,  $R_{NED}$ , através da matriz  $S$  como sendo:

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} X_{NED} \\ Y_{NED} \\ Z_{NED} \end{bmatrix} + S^T \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{bmatrix} \quad (41)$$

em que  $(X_{NED} \ Y_{NED} \ Z_{NED})$  é igual ao vector com a posição actual do Quadrirotor no referencial inercial,  $R_{NED}$ , e  $(X_{cam} \ Y_{cam} \ Z_{cam})$  a posição da câmara em relação ao referencial móvel,  $R_{ABC}$ . Partindo do conjunto da nova posição, Equação 41 e da atitude, Equação 40, juntamente com as coordenadas de um qualquer ponto de referência no referencial fixo e do comprimento focal  $f$  que caracteriza a câmara, é possível obter a sua projecção no referencial da imagem,  $R_{xyz}$  através da matriz  $M$  definida pela multiplicação das matrizes que caracterizam a rotação,  $S$ , e translação,  $T$ , dos pontos de referência.<sup>4</sup>

<sup>4</sup>Por motivos de interesse e de espaço, a dedução completa das equações referentes à modelação da câmara foi deixada para consulta ao longo do Apêndice B

$$\begin{bmatrix} Q_P^T & 1 \end{bmatrix} = \begin{bmatrix} Q^T & 1 \end{bmatrix} \begin{bmatrix} S_T^T & 0 \\ -T^T & 1 \end{bmatrix} = \begin{bmatrix} Q & 1 \end{bmatrix} M \quad (42)$$

em que  $Q$  corresponde à matriz com as coordenadas dos pontos de referência no referencial inercial  $R_{NED}$ , e  $Q_P$  à matriz com a projecção desses mesmos pontos no referencial da imagem,  $R_{xyz}$ .

$$Q_P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = SQ - T \quad (43)$$

em que juntando a projecção fica:

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} x/z \\ y/z \\ 1 \end{bmatrix} \quad (44)$$

Como foi dito anteriormente, a projecção dos pontos vem no referencial da imagem,  $R_{xyz}$ , em termos de  $(x, y)$ , pelo que, de modo a simular mais correctamente a utilização de uma câmara verídica, procedeu-se à pixelização para uma resolução tipicamente VGA de  $h=640$  por  $w=380$  pixels. Aos valores de saída em termos de  $(x, y)$  correspondentes à matriz  $Q_P$  calculada, foram aplicadas as seguintes equações:

$$\begin{cases} x_p = \frac{1}{w} \text{round}(x_i \times w) \\ y_p = \frac{1}{h} \text{round}(y_i \times h) \end{cases} \quad (45)$$

## 4 Controlador

O objectivo deste trabalho é conseguir estabilizar o Quadrirotor numa dada posição de referência. Essa posição de referência consiste numa primeira fase, numa situação de voo pairado a uma dada altitude. Para isso é necessário considerar um sistema de controlo de modo a permitir que as instabilidades do movimento sejam compensadas. O controlador escolhido, foi o controlador moderno LQR (*Linear Quadratic Regulator*). Este controlador é dos mais eficientes e mais usados em técnicas de controlo, particularmente devido à sua fácil implementação e à robustez nas técnicas de controlo MIMO (Multi Input Multi Output) como é o caso deste sistema. Para aplicação do LQR é necessário que as equações do movimento do Quadrirotor se apresentem sob a forma de espaço de estados linear, isto é:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \quad (46)$$

No entanto para que tal seja possível, é necessário que as equações sejam lineares ou linearizadas.

### 4.1 Linearização das equações

É possível observar que as equações obtidas na Secção 3.2.2 referentes à modelação da dinâmica do sistema, deram origem a equações não lineares. Como foi referido anteriormente, para aplicação do sistema de controlo adoptado, as equações têm de ser lineares. Isto quer dizer que se tem de proceder à linearização das equações anteriormente obtidas em torno de uma posição de referência.

O estado considerado para a linearização do Quadrirotor inclui as velocidades linear e angular no referencial ABC,  $R_{ABC}$ ,  $[U \ V \ W]^T$  e  $[P \ Q \ R]^T$  respectivamente, a posição no referencial fixo NED,  $R_{NED}$ ,  $[e_x \ e_y \ e_z]^T$  e os ângulos de Euler  $[\phi \ \theta \ \psi]^T$ , resultando num estado de 12 variáveis. A entrada é a tensão fornecida aos quatro motores,  $U = [U_1 \ U_2 \ U_3 \ U_4]^T$ .

Para uma situação de voo pairado a uma altitude  $h$ , considerando os ângulos de Euler para controlo, temos como referência:

$$X_0 = [U \ V \ W \ P \ Q \ R \ e_x \ e_y \ e_z \ \phi \ \theta \ \psi] = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ -h \ 0 \ 0 \ 0] \quad (47)$$

E usando a expansão em termos de séries de Taylor de primeira ordem, temos aproximadamente que a derivada em ordem ao tempo das perturbações dos estados é igual a:

$$\dot{x} = f(X_0, U_0) + \frac{\partial f}{\partial X} \Big|_{X=X_0} x + \frac{\partial f}{\partial U} \Big|_{U=U_0} u \quad (48)$$

Considerando que o estado  $X$  é definido pela soma de uma perturbação  $x$  com a sua posição de referência  $X_0$ ,  $X = x + X_0$  e da mesma forma  $U = u + U_0$ , a expressão anterior fica:

$$\dot{x} = f(X_0, U_0) + \frac{\partial f}{\partial X}(X - X_0) + \frac{\partial f}{\partial U}(U - U_0) \quad (49)$$

Resolvendo agora  $f(X_0, U_0)$  para cada uma das equações do movimento, em que  $U_0$  é tal que o somatório exercido em cada um dos rotores leva a  $\dot{W} = 0$  (posição de referência a uma dada altitude com aceleração vertical nula), obtém-se uma situação de equilíbrio:



$$f(X_0, U_0) = 0 \quad (50)$$

isto quer dizer que da Equação 48 inicial resulta apenas:

$$\dot{x} = \frac{\partial f}{\partial X}(X - X_0) + \frac{\partial f}{\partial U}(U - U_0) \quad (51)$$

Por comparação com a Equação 46 deduz-se a matriz da dinâmica:

$$A = \frac{\partial f}{\partial X} = \begin{bmatrix} \frac{\partial f_1}{\partial X_1} & \frac{\partial f_1}{\partial X_2} & \cdots & \frac{\partial f_1}{\partial X_{12}} \\ \frac{\partial f_2}{\partial X_1} & \frac{\partial f_2}{\partial X_2} & \cdots & \cdots \\ \vdots & \vdots & \ddots & \\ \frac{\partial f_{12}}{\partial X_1} & \vdots & & \frac{\partial f_{12}}{\partial X_{12}} \end{bmatrix}_{X=X_0; U=U_0} \quad (52)$$

isto quer dizer que a matriz  $A$  corresponde à matriz Jacobiana em que cada uma das funções  $f$  corresponde a uma equação do movimento (equações de estado) que podem ser consultadas no Apêndice A e  $X$  é o vector dos estados do Quadrirotor, assumindo os ângulos de Euler para controlo ao invés dos quatérnios. Analogamente, a matriz de entrada  $B$  corresponde à matriz Jacobiana mas em termos dos parâmetros de entrada  $U$  correspondentes a cada um dos motores,  $U = [U_1 \ U_2 \ U_3 \ U_4]$ :

$$B = \frac{\partial f}{\partial U} = \begin{bmatrix} \frac{\partial f_1}{\partial U_1} & \frac{\partial f_1}{\partial U_2} & \cdots & \frac{\partial f_1}{\partial U_4} \\ \frac{\partial f_2}{\partial U_1} & \frac{\partial f_2}{\partial U_2} & \cdots & \cdots \\ \vdots & \vdots & \ddots & \\ \frac{\partial f_{12}}{\partial U_1} & \vdots & & \frac{\partial f_{12}}{\partial U_4} \end{bmatrix}_{X=X_0; U=U_0} \quad (53)$$

Para o cálculo da matriz  $B$ , é assumido que as forças segundo  $x$  e  $y$  são nulas,  $F_x \simeq F_y \simeq 0$  quando comparadas com a força segundo  $z$  criada pela totalidade dos rotores, e é considerado que todos eles se comportam de igual maneira e que por isso todos contribuem na mesma proporção para a força propulsiva:

$$F_z = \sum_{i=1}^4 F_i = F_1 + F_2 + F_3 + F_4 \quad (54)$$

Em relação aos momentos temos que:

$$\begin{cases} M_x = (F_3 - F_4) L \\ M_y = (F_2 - F_1) L \\ M_z = (F_1 + F_2 - F_3 - F_4) K_K \end{cases} \quad (55)$$

em que  $L$  corresponde à distância entre o CG (assumido como sendo exactamente igual ao ponto central da estrutura) e o eixo de rotação de cada um dos rotores,  $K_K$  é a constante característica da modelação definida na Secção 3.3, que relaciona o momento segundo  $e_3$  com as forças aplicadas.

## 4.2 LQR (Regulador Quadrático Linear)

Nesta secção será dado a conhecer de uma forma breve os princípios teóricos do controlador adoptado, LQR. O LQR consiste na minimização de uma função de custo  $J$  que pode assumir variadíssimas formas, correspondendo a situações e objectivos diferentes, cuja optimização em conjunção com a restrição do sistema (expressa pelas equações da dinâmica), conduz a uma solução  $u^0$  que minimiza o valor de  $J$ . No caso usual de uma referência nula e de um intervalo de tempo semi-infinito, correspondente ao problema do regulador quadrático linear, onde no caso estacionário a função de custo tem matrizes de ponderação (matrizes  $Q$  e  $R$ ) constantes:

$$J = \frac{1}{2} \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (56)$$

e conduz a uma solução óptima linear da forma:

$$u^0 = -Kx \quad (57)$$

A minimização da função de custo com a restrição da dinâmica do sistema resulta na resolução de uma equação algébrica de Riccati (no caso estacionário):

$$A^T P + P A - P B R^{-1} B^T P + Q = 0 \quad (58)$$

em ordem a uma matriz  $P$  constante, simétrica, semidefinida e positiva, a partir da qual é obtida a matriz de realimentação [17]:

$$K = R^{-1} B^T P \quad (59)$$

A resolução do problema do regulador quadrático linear pode ser obtida através do comando *lqr* do MATLAB dadas as quatro matrizes:  $A$ ,  $B$ ,  $Q$ ,  $R$  e o período de amostragem  $t_s$ . A matriz de realimentação  $K$  obtida é então colocada no anel de realimentação de forma a fechar o anel e a estabilizar o sistema. A matriz  $Q$  e a matriz  $R$  são denominadas matrizes de ponderação dos estados e entradas respectivamente, sendo os seus valores “arbitrários” ( $Q \geq 0$ ;  $R > 0$ ), influenciando não só a estabilidade como a rapidez do sistema. O método de Bryson sugere uma definição para  $Q$  e  $R$ , colocando-as como matrizes diagonais onde cada um dos termos é o quadrado do inverso do máximo esperado para cada variável durante a manobra [17]:

$$Q_{ii} = \frac{1}{x_{i,max}^2} \quad R_{ii} = \frac{1}{u_{i,max}^2} \quad (60)$$

## 4.3 Estimador

Num controlo realístico nem sempre os estados não estão todos disponíveis para ser realizada a realimentação. Em vez disso, são as saídas provenientes dos sensores que são acessíveis. Usando controlo moderno, é possível consoante o número e o tipo de medições retiradas da dinâmica do sistema, estimar os estados parciais ou totais do problema. É função do estimador saber calcular o valor dos estados através de informação parcial recebida dos sensores. Após a estimação dos estados,

são esses mesmos estados estimados que são usados na realimentação do sistema.

Para o projecto de um Quadrirotor estamos interessados em controlar doze estados. Três velocidades lineares, três velocidades angulares, a posição e a atitude do UAV em termos de ângulos de Euler, com base na rotação de cada um dos quatro rotores envolvidos. Só assim conseguiremos uma capacidade de voo autónomo por parte de um Quadrirotor. O ALIV só tem disponíveis os três acelerómetros e o magnetómetro, e não é possível obter uma estimativa dos doze estados pretendidos, em vez disso, o estimador apenas consegue observar seis dos mesmos estados, as três velocidades angulares,  $(P \ Q \ R)$ , e a atitude do ALIV,  $(\phi \ \theta \ \psi)$ . Desta forma, é impossível conseguir automatizar o ALIV porque a translação não é observável.

Uma vez que se faz uso da técnica de controlo moderno LQR (Secção 4.2) em que a realimentação é feita usando todos os estados que caracterizam o sistema, é necessário que o estimador construído saiba estimar a totalidade dos estados para uma futura realimentação. Esta técnica denomina-se por LQG, *linear quadratic gaussian* [16].

O filtro de Kalman é um estimador usado para navegação ou outras aplicações que requerem a reconstrução dos estados através de medições do ruído, baseado num tratamento probabilístico do processo e medição do ruído [16]. Com base na teoria amplamente explicada na referência adoptada como base, e partindo desta vez de uma dinâmica estocástica de um sistema em espaço de estados caracterizada por:

$$\begin{cases} \dot{x} = Ax + Bu + Gw \\ y = Cx + Du + v \end{cases} \quad (61)$$

em que  $w(t)$  é um processo de ruído desconhecido que actua no sentido de perturbar a aeronave, como por exemplo o vento, e  $v(t)$  uma medição de ruído desconhecida geralmente associada ao ruído dos sensores. Uma vez que a dinâmica do sistema (Equação 61) é constituída por a caracterização de ruídos associados à aeronave, o estado  $x(t)$  é agora um estado probabilístico, assim como o são as saídas  $y(t)$ , os seus valores são obtidos assim, partindo de valores expectáveis para as variáveis  $w(t)$  e  $v(t)$  [16]. Assumindo que  $w(t)$  e  $v(t)$  são processo de ruído branco, caracterizados em termos das matrizes espectrais  $Q_K = E\{ww^T\}$  e  $R_K = E\{vv^T\}$ , ficando assim estabelecidos os pressupostos para a construção de um estimador na forma da Equação 34 para o sistema estocástico em espaço de estados referido na Equação 61, em que agora a estimativa da saída é dada por:

$$\hat{y} = E\{Cx + v\} = C\hat{x} \quad (62)$$

Utilizando a dinâmica do sistema da Equação 61, a derivada do erro da dinâmica é dada por:

$$\dot{\tilde{x}} = (A - LC)\tilde{x} + Gw - Lv \quad (63)$$

em que o erro da saída definido como  $\tilde{y} = \hat{y} - y$  é igual a:

$$\tilde{y} = C\tilde{x} + v \quad (64)$$

e o erro da covariância dado por:

$$P(t) = E\{\tilde{x}\tilde{x}^T\} \quad (65)$$

O filtro de Kalman procura assim calcular a matriz de ganho óptimo  $L$  que minimize o erro da covariância  $P$ . O processo de obtenção da matriz  $L$  para o caso do filtro de Kalman é semelhante ao problema demonstrado na teoria referente ao controlo moderno LQR (Secção 4.2) na Equação 57. Pelo que, e mais uma vez por o MATLAB a calcular à semelhança da matriz de realimentação do ganho  $K$ , a teoria envolvida não vai ser novamente ilustrada, podendo ser consultada na referência adoptada [16].

O filtro de Kalman é assim obtido com base no comando *kalmf* do MATLAB, que tem como parâmetros de entrada o sistema em espaço de estados a ser observado, as matrizes espectrais  $Q_K$  e  $R_K$  e o período de amostragem  $t_s$ . O mesmo comando devolve um sistema em espaço de estados, *KEST*, constituído pelo filtro de Kalman e que tem como saídas não só a estimativa dos estados e das próprias saídas.

## 5 Implementação em SIMULINK

Como foi explicitado na Secção 4.3, de modo a tornar a simulação mais realística, recorreu-se a uma realimentação da estimativa dos estados obtidos através do filtro de Kalman com base nas saídas provenientes dos sensores a bordo do ALIV. Sendo assim, toda a filosofia adoptada na construção dos diferentes módulos de simulação baseou-se no mesmo princípio, como ilustra a Figura 19.

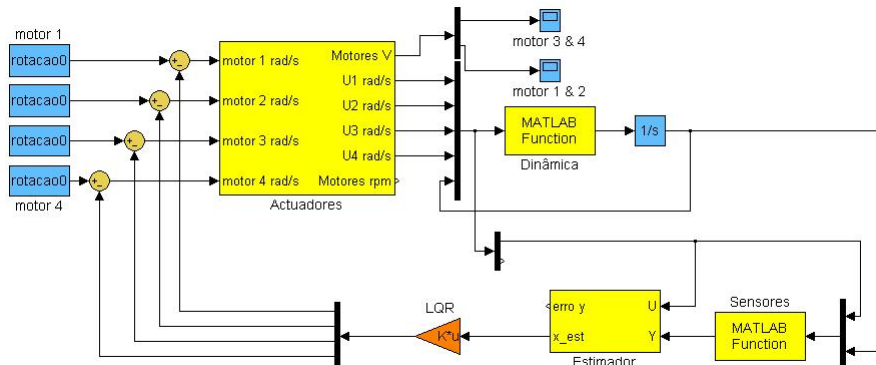


Figura 19: Filosofia dos módulos durante a fase de simulação

Em termos gerais existem três blocos principais, o primeiro bloco representa a dinâmica de um Quadrirotor convencional e é constituído pelas treze equações deduzidas na Secção 3.2.2. A estas treze equações estão associadas treze variáveis, que após integração e passagem de quaterniões a Euler constituem os doze estados de um Quadrirotor convencional para o controlo. O segundo bloco representa os sensores embarcados, neste bloco é simulado o comportamento dos sensores com base nas equações deduzidas na Secção 3.5, obtendo-se como saídas as leituras efectuadas pelos sensores consoante o módulo a ser simulado. Em ambos os blocos são recebidos como variáveis de entrada os estados actuais do Quadrirotor e a rotação de cada um dos rotores para que os cálculos possam ser efectuados consecutivamente. O terceiro bloco principal consiste na modelação do controlador de acordo com o módulo a ser simulado, Secção 4.3. Este bloco é responsável pela criação do filtro de Kalman e pela realimentação dos estados estimados, para obtenção do factor correctivo ideal a aplicar a cada um dos rotores. Por fim existe ainda um bloco que representa os actuadores. Neste bloco, como foi explicado na Secção 3.3, são aplicados os parâmetros do modelo dos motores adoptado, de modo a simular de uma forma mais realística o funcionamento dos mesmos. Todos estes blocos são iniciados através de um ficheiro<sup>5</sup> responsável pela construção dos valores iniciais dos estados, das matrizes e dos ganhos envolvidos de acordo com a simulação e a situação pretendida.

Uma vez que o ALIV não possui qualquer tipo de sensor para o controlo da posição, foram assumidas duas alternativas de modo a poder controlar integralmente os dozes estados do ALIV.

A primeira das situações consistiu em utilizar apenas um joystick para o controlo da posição e os sensores a bordo para o controlo da atitude: módulo com joystick (Secção 5.1.1).

A segunda hipótese consistiu na modelação de uma câmara a bordo do ALIV, de modo a torná-lo completamente autónomo através da visualização e dos sensores anteriormente considerados: módulo de visualização (Secção 5.1.2).

<sup>5</sup>O Apêndice C apresenta entre outros, o ficheiro *inicio.m*, responsável pela criação dos parâmetros iniciais envolvidos no começo da simulação, e das matrizes de ganho ideais para o tipo de simulação pretendida.

Para finalizar foi ainda criado um módulo que engloba os dois anteriores: módulo integrado (Secção 5.1.3). Neste módulo é conduzido o Quadrirotor até uma posição em que possa observar os pontos de referência ou alvo, através do módulo com joystick, e a partir daí passe a voo autónomo através do módulo de visualização.

## 5.1 Módulos finais

### 5.1.1 Módulo com Joystick

Como já foi referido anteriormente na Secção 3.5, o ALIV possui actualmente dois tipos de sensores, três acelerómetros que medem a aceleração global num dado ponto, e um magnetómetro responsável primordialmente pelo controlo da guinada. Com base na modelação matemática desenvolvida na Secção 3.5 através destes dois conjuntos de sensores temos seis variáveis de saída. Três das variáveis representam a aceleração total num dado ponto para cada um dos eixos  $e_1$ ,  $e_2$ ,  $e_3$  respectivamente, e as outras três variáveis a componente do vector unitário de acordo com a atitude do ALIV inicialmente calibrado a apontar para norte,  $y = (a_{px}, a_{py}, a_{pz}, N_{mag}, E_{mag}, D_{mag})$ .

Através única e exclusivamente destas saídas verificou-se que a translação não é observável, isto é, o filtro de Kalman não consegue observar os doze estados para o controlo integral do Quadrirotor. Uma vez não ter qualquer informação que possa ser restituída para estimar a posição, as velocidades lineares do mesmo também não são conseguidas. Como tal, dos dozes estados iniciais, com base nos sensores a bordo do ALIV o filtro de Kalman apenas consegue estimar seis deles, as três velocidades angulares e os ângulos de Euler,  $x = (p, q, r, \phi, \theta, \psi)$ . Assim surgiu como hipótese a criação de um módulo que controlasse a posição através do joystick de modo a tornar o mesmo mais versátil. A ideia foi atribuir a cada um dos graus de liberdade do joystick, um dos graus de liberdade do Quadrirotor, controlados através da atribuição das acelerações lineares e da aceleração angular segundo  $e_3$ .

Para fazer variar a posição do Quadrirotor, de acordo com a Secção 2.1, é necessário fazer variar a rotação entre cada par de rotores. Assim, é possível fazer o Quadrirotor deslocar-se ao longo dos três eixos inerciais pela introdução de velocidades lineares. Devido às grandes instabilidades inerentes ao movimento e estabilização de um Quadrirotor convencional, ao movimento do joystick não basta estarem associadas as velocidades lineares, é necessário que essas mesmas velocidades conduzam à estabilização da atitude do Quadrirotor de modo a torná-lo controlável. Sendo assim, estamos interessados em obter uma matriz, que para uma dada velocidade linear introduzida através do joystick, não só desloque o Quadrirotor sobre uma determinada direcção, como ao mesmo tempo o faça posteriormente estabilizar.

Para o cálculo dessa matriz foi considerado mais uma vez o sistema em espaço de estados original:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \quad (66)$$

em que a matriz da dinâmica  $A$  tem dimensões  $[6 \times 6]$  correspondente aos seis estados  $x = (p, q, r, \phi, \theta, \psi)$ , e a matriz de entrada  $B$   $[6 \times 4]$  correspondente às quatro entradas dos motores  $u = (u_1, u_2, u_3, u_4)$ . Considerando a realimentação deste sistema temos que a matriz em anel fechado correspondente é dada por:

$$A_f = (A - BK) \quad (67)$$

o que conduz à equação:

$$\dot{x} = (A - BK)x + Bu \quad (68)$$

Como estamos interessados que a uma perturbação inicial causada pela variação da rotação dos motores, o ALIV reaja com uma deslocação transversal e a conseqüente estabilização, estamos interessados em resolver:

$$0 = (A - BK)x + Bu \Leftrightarrow (A - BK)x = -Bu \Leftrightarrow x = (A - BK)^{-1}(-Bu) = -A_f^{-1}Bu \quad (69)$$

De modo a conseguirmos relacionar as saídas do joystick em termos de acelerações lineares e aceleração angular segundo  $e_3$ ,  $y_j = (\dot{U}, \dot{V}, \dot{W}, \dot{R})$ , com as entradas aos motores  $u = (u_1, u_2, u_3, u_4)$ , pretendemos resolver o seguinte sistema:

$$y_j = \begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \\ \dot{R} \end{bmatrix} = C_j x + D_j u \quad (70)$$

em que  $C_j$  é uma matriz com dimensões  $[4 \times 6]$  e  $D_j$  uma matriz com dimensões  $[4 \times 4]$ . Ambas obtidas mais uma vez com base no Jacobiano, desta vez entre as equações provenientes dos sensores com os estados  $x = (p, q, r, \phi, \theta, \psi)$ , caso da matriz  $C_j$ , e entre as equações provenientes dos sensores com as entradas  $u = (u_1, u_2, u_3, u_4)$ , caso da matriz  $D_j$ . No fundo estas matrizes são obtidas com base no conjunto de linhas e colunas pretendidas das matrizes originais  $A$  e  $B$  do sistema da Equação 66 para se obter as saídas pretendidas  $y_j = (\dot{U}, \dot{V}, \dot{W}, \dot{R})$ .

No entanto, como para este caso as saídas provenientes do joystick de acordo com os movimentos pretendidos são as saídas definidas na Equação 70, de acordo com a Equação 69 estamos interessados em resolver o seguinte sistema:

$$\begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \\ \dot{R} \end{bmatrix} = -C_j (A - BK)^{-1} Bu + D_j u = [-C_j A_f^{-1} B + D_j] u \quad (71)$$

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = [-C_j A_f^{-1} B + D_j]^{-1} \begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \\ \dot{R} \end{bmatrix} = J \begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \\ \dot{R} \end{bmatrix} \quad (72)$$

em que a matriz  $J$  é a matriz que relaciona o movimento do joystick com a deslocação transversal do Quadrirotor em termos dos parâmetros de entrada aos motores  $u = (u_1, u_2, u_3, u_4)$ , e  $y_j = (\dot{U}, \dot{V}, \dot{W}, \dot{R})$  as saídas provenientes do joystick. Os valores da Equação 72 serão então adiciona-

dos aos valores provenientes da realimentação dos seis estados. Isto faz com que sejam atribuídos ao ALIV velocidades lineares que permitam a sua deslocação transversal e a consequente estabilização. Com base nisto, a Figura 20 demonstra o raciocínio utilizado na construção deste módulo durante a fase de simulação.

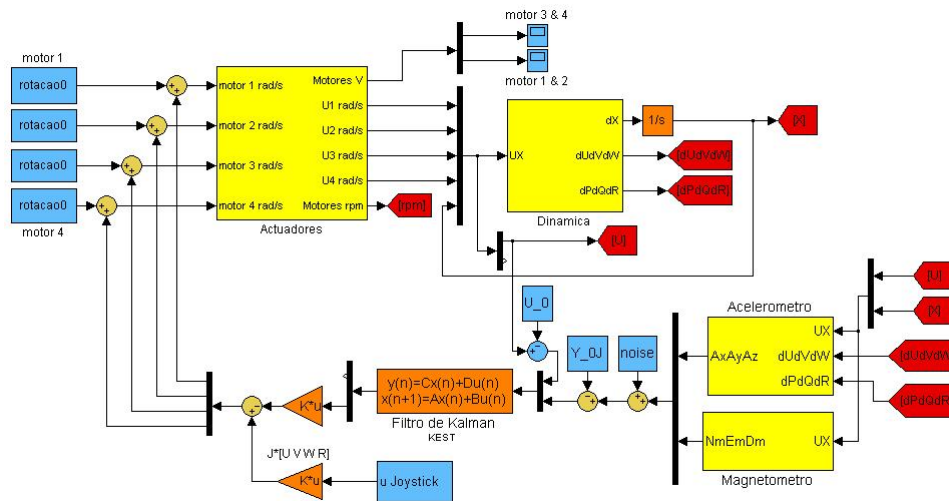


Figura 20: Ilustração do módulo com joystick durante a fase de simulação

### 5.1.2 Módulo de visualização

Um dos problemas no controlo de um Quadricóptero é conseguir a sua estabilização, e o principal objectivo devido às suas dificuldades em termos de pilotagem, é que o mesmo consiga ser o mais autónomo possível. Foi com base nestes dois pressupostos que a utilização do modelo desenvolvido para a câmara embarcada (Secção 3.5.3) revelou ser de extrema importância.

Na solução desenvolvida anteriormente, devido à inexistência de um sensor que conseguisse controlar a posição, a mesma foi conseguida através de um joystick. Embora a simulação resultasse, não é o mais ideal para os objectivos a que se propõe um Quadricóptero. Ao Quadricóptero desenvolvido nestes moldes é exigido uma posição de permanência a dada altitude, quer seja para a observação de um elemento estacionário quer seja para um objecto com movimento inferior às capacidades de movimentação do Quadricóptero. Se tivermos um controlo da posição, esse objectivo é conseguido sem que o utilizador tenha de fazer um constante ajuste através do joystick, tornando o Quadricóptero totalmente autónomo e muito mais eficaz.

Com o modelo da câmara desenvolvido na Secção 3.5.3, para além das seis saídas já consideradas para o caso dos acelerómetros e do magnetómetro, temos mais oito saídas correspondentes às projecções no referencial  $R_{xyz}$  dos quatro pontos alvo. Quatro projecções segundo  $x$ , e mais quatro segundo  $y$ . Utilizando o total das catorze saídas provenientes dos sensores, o filtro de Kalman está agora em situação de conseguir observar os doze estados globais. É assim possível obter um controlo tanto da atitude como da posição do ALIV com base nas estimativas calculadas.

Para o desenvolvimento deste módulo, ao contrário do módulo anterior, foram utilizadas as matrizes Jacobiana para a totalidade dos estados. Desta vez a matriz da dinâmica  $A$  para o sistema em espaço de estados comuns terá dimensões  $[12 \times 12]$  e a matriz de entrada  $B$   $[12 \times 4]$ ,



isto é  $x=(u, v, w, p, q, r, x, y, z, \phi, \theta, \psi)$ , e  $u=(u_1, u_2, u_3, u_4)$ . Devido à adição das saídas referentes à projecção dos pontos, a matriz  $C$  terá dimensões  $[14 \times 12]$  e a matriz  $D$   $[14 \times 4]$ ,  $y=(a_{px}, a_{py}, a_{pz}, x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, N_{mag}, E_{mag}, D_{mag})$ . Seguindo os moldes habituais para a obtenção da realimentação através da matriz de ganho  $K$  do LQR e do sistema característico em sistema espaço de estados que define o filtro de Kalman (Secção 4), foram atribuídas as matrizes de ponderação  $Q$  e  $R$ , e as matrizes de covariância  $Q_K$  e  $R_K$  que melhor satisfizessem o sistema<sup>6</sup>.

Como o filtro de Kalman procura minimizar o erro ao longo do tempo entre o estado real e o estimado, à projecção dos pontos pode ser atribuído um movimento característico ao longo do tempo. Isto faz com que o ALIV reaja de forma a minimizar o erro existente entre a projecção dos pontos e a sua posição, atribuindo-lhe capacidades autónomas na observação, por exemplo, de um objecto em movimento.

De acordo com a filosofia utilizada no desenvolvimento dos módulos de simulação, tanto o filtro de Kalman como o controlador são constantes e construídos de acordo com a posição inicial de referência. A altitude de observação bem como qualquer outro estado é portanto definida no ficheiro inicial antes de lançar a simulação. A Figura 21 ilustra o raciocínio utilizado na construção deste módulo.

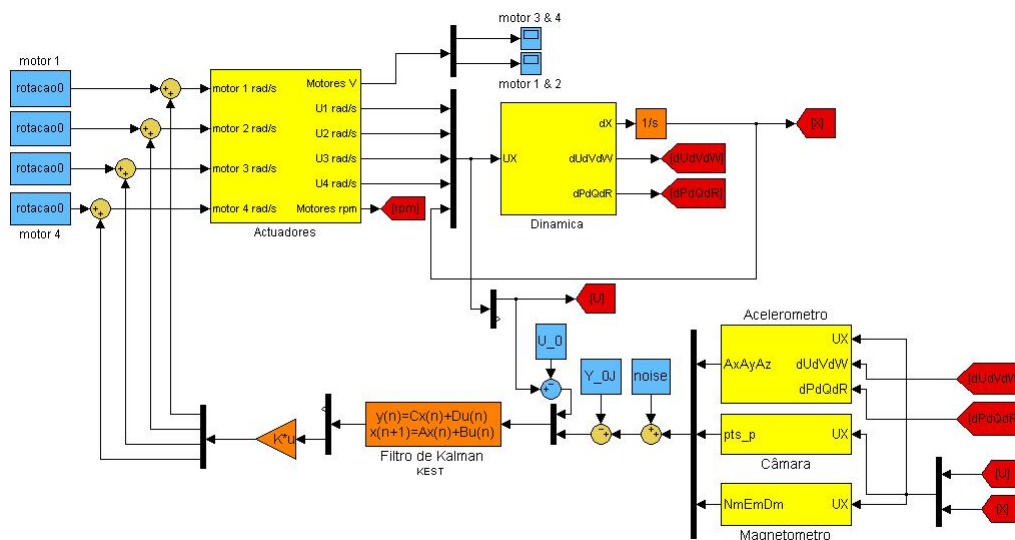


Figura 21: Ilustração do módulo de visualização durante a simulação

### 5.1.3 Módulo integrado

Um inconveniente do primeiro módulo de simulação é a dificuldade de estabilizar o ALIV numa dada posição, bem como a necessidade da permanência do utilizador no controlo do mesmo através do joystick. Um inconveniente do segundo módulo de simulação é a necessidade do ALIV ter de estar a uma dada altitude de modo a que possa observar os pontos ou o objecto de referência. Deste modo, foi construído um módulo integrado, em que inicialmente é conduzido o Quadrirotor através do módulo com joystick até uma posição em que consiga observar os pontos de referência, e a partir daí passa a voo autónomo através do módulo de visualização.

Dado que a dinâmica e os sensores se mantêm inalterados para cada um dos módulos, à excepção

<sup>6</sup>A construção do controlador pode ser analisada no ficheiro *controlo.m* inserido no Apêndice C.

da integração da câmara embarcada, a diferença reside no bloco do controlador, responsável pela estimativa dos estados e pela realimentação das entradas dos motores. Como estamos a lidar com duas situações distintas, em que as saídas dos sensores e os estados a serem estimados são de dimensões diferentes, é necessário então utilizar dois controladores. Um deles é o construído no módulo com joystick, o outro, o construído no módulo de visualização.

A altitude final de observação é mais uma vez definida no ficheiro inicial, e a permuta entre os dois módulos (controladores) é efectuada através do “*throttle*” do joystick. Na primeira fase da simulação, é necessário elevar o Quadrirotor e posicioná-lo de uma forma mais ou menos estabilizada numa posição em que possa visualizar os pontos alvo e se situe na proximidade da altitude de observação. A partir daí, e com a alteração do “*throttle*” do joystick, utilizando agora o controlador desenvolvido no módulo de visualização, o mesmo estabiliza na posição escolhida como referência. A partir deste momento é possível o ALIV comportar-se de uma forma completamente autónoma. A Figura 22 ilustra o raciocínio utilizado na construção deste módulo durante a simulação.

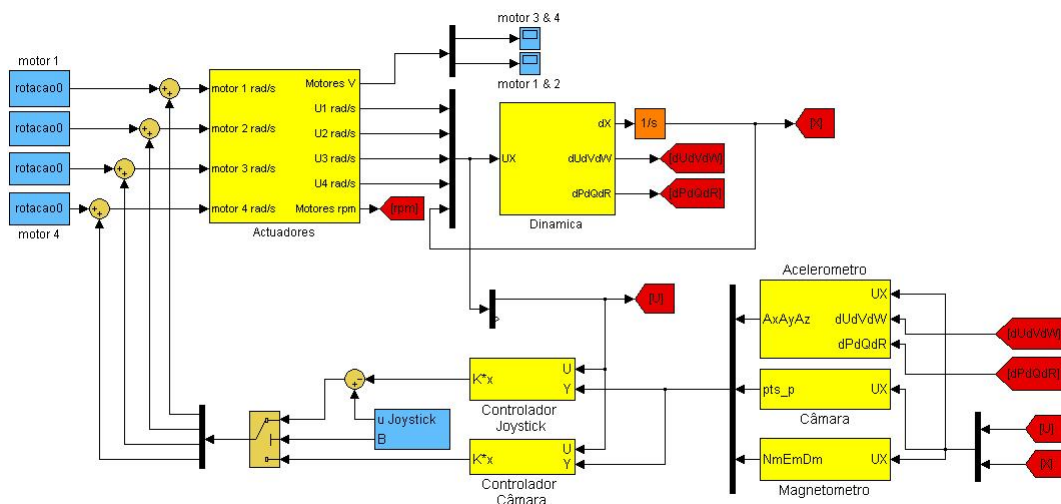


Figura 22: Ilustração do módulo integrado durante a simulação

## 5.2 Tempo real

Como tem vindo a ser descrito ao longo deste trabalho, é objectivo do mesmo aproximar o mais possível as simulações obtidas com a realidade, tendo em vista a futura implementação do estudo em ambiente real. Assim, de modo a tornar a reacção do ALIV ajustada em relação ao envio dos dados por parte dos sensores e à consequente correcção de tensão aplicada a cada um dos rotores, foi criado um modelo em tempo real através do MATLAB e SIMULINK, usando as ferramentas *Real-Time Workshop* e *Real-Time Windows Target*.

### 5.2.1 Criação de um executável através do *Real-Time Workshop*

O *Real-Time Workshop* é o primeiro passo para a construção de um sistema em tempo real. O *Real-Time Workshop* é uma extensão das capacidades do Simulink e do MATLAB que automaticamente gera, agrega e compila um determinado executável através de um sistema em diagrama de blocos. Este executável tem a capacidade de produzir posteriormente um *MAT-file* contendo os resultados da

execução do modelo em tempo real. Neste capítulo, por não ser relevante, não será debatido o modo de funcionamento e os princípios do *Real-Time Workshop*, podendo ser consultada essa informação na documentação que acompanha o MATLAB. No entanto, ao longo do Apêndice D, serão ilustrados os aspectos mais importantes que levaram à construção do executável.

Uma das limitações existentes actualmente no processo de criação de um executável a partir de um sistema em SIMULINK, é que a totalidade do sistema seja constituído por diagrama de blocos. Isto faz, com as *MATLAB Functions* utilizadas até ao momento tenham de ser convertidas para diagrama de blocos (Apêndice D).

Depois destas conversões, o sistema desenvolvido em ambiente Simulink está em condições de ser utilizado pelo *Real-Time Workshop*. Para tal, é necessário definir os parâmetros de resolução do sistema e proceder posteriormente à compilação do executável (Apêndice D).

Um dos problemas do ficheiro criado é de ter apenas a resolução do sistema para uma situação pontual. Para se verem os resultados de outra simulação, a mesma teria de ser definida inicialmente, o sistema novamente compilado, e por fim o executável novamente lançado. Para a simulação do ALIV, e da forma como ele foi desenvolvido, estamos interessados em poder alterar constantemente não só as situações iniciais, bem como efectuar a permuta dos controladores, de modo a observar instantaneamente e em tempo real as alterações provocados no ALIV. Para isso ser possível, foi utilizado posteriormente à criação do executável, o *Real-Time Windows Target*.

### 5.2.2 Real-Time Windows Target

O *Real-Time Windows Target* é uma solução para observar e testar os sistemas em tempo real. É um ambiente em que o computador que corre o ficheiro em tempo real e o computador onde se visualizam os resultados é o mesmo. O *Real-Time Windows Target* realiza a interacção entre o executável e o SIMULINK, permitindo ao utilizador usar o modelo desenvolvido em SIMULINK para visualizar os resultados, usando os mesmos blocos que em modo normal. O *Real-Time Windows Target* permite ainda alterar as constantes envolvidas durante a simulação, da mesma forma que se faz no modo normal, tornando assim a resolução do sistema em tempo real muito mais flexível.

À semelhança do que aconteceu em relação ao *Real-Time Workshop*, pela falta de relevância em relação ao trabalho desenvolvido não será descrito o modo de funcionamento do *Real-Time Windows Target*, que pode ser consultado vastamente por toda a documentação que acompanha o MATLAB. No entanto, o processo utilizado neste trabalho para se obter a interacção do *Real-Time Windows Target* com o modelo desenvolvido pode ser acompanhado em pormenor ao longo do Apêndice D.

Para se proceder à interacção entre o modelo em ambiente SIMULINK e o executável, é necessário fazer uma nova compilação do sistema, desta vez usando o *Real-Time Windows Target*. Depois, é necessário usar o modo externo do SIMULINK para conectar o sistema ao executável. Desta forma, quando se iniciar a simulação, a mesma decorre em tempo real, podendo os dados ser observados e alterados exactamente da mesma maneira que se faz quando se utilizam modelos que correm no modo normal.

Todo o controlo da simulação desenvolvida é caracterizada por determinados movimentos do joystick, quer seja para definir o módulo de simulação, quer seja para controlar a posição do ALIV quando estamos a operar no módulo do joystick. Através da interacção com executáveis a correr em tempo real, é necessário fazer a interacção em termos de hardware correspondente ao joystick de modo

a que este possa operar em tempo real. Sendo assim, os blocos tradicionais que procuram simular não só os eixos como os botões do joystick tiveram de ser substituídos por blocos característicos do “*Real-Time Windows Target*” para o efeito.

A definição dos eixos foi atribuída a blocos de “*analog in*” e os botões a “*digital in*”. Dentro de cada um destes blocos, foi atribuído o canal correspondente ao eixo ou ao botão pretendido respectivamente. Com este efeito, foi possível definir o joystick em tempo real, processo fundamental para a visualização e controlo dos movimentos do ALIV, caso contrário a simulação não faria sentido. A Figura 23 ilustra os blocos usados na definição do processo de selecção do módulo de simulação.

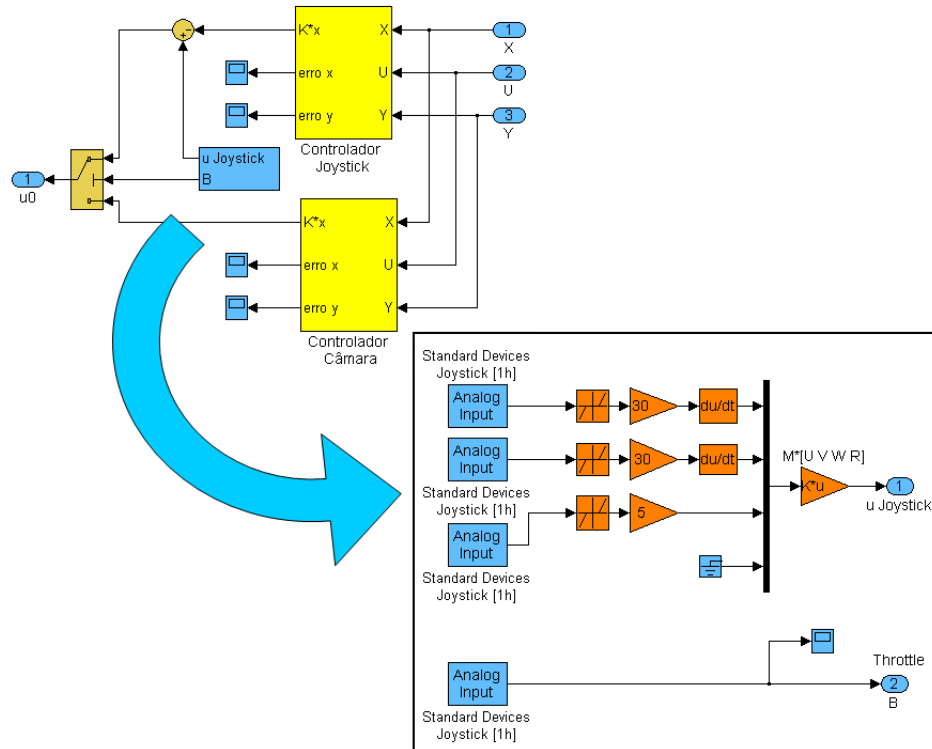


Figura 23: Implementação do joystick em tempo real

Nesta figura pode ser visualizado o bloco responsável pela permuta dos controladores, e ainda, o princípio adiantado na formulação do problema quando o ALIV opera no módulo com joystick, Secção 5.1.1.

Para além dos blocos de “*analog input*” característicos do “*throttle*” e da atribuição dos movimentos característicos do ALIV, rolamento, picada e altitude, existem ainda os “*digital input*” (botões do joystick), responsáveis no modelo desenvolvido, pela atribuição de movimento transversal aos pontos de referência, representando uma simulação de uma situação de vigilância, a um objecto com velocidades moderadas.

### 5.3 Visualização 3D dos resultados

Após o desenvolvimento da plataforma capaz de simular o comportamento do ALIV, e de acordo com o princípio de funcionamento do mesmo, ficou evidente que o principal objecto de estudo é controlar a atitude e a posição do mesmo face às diversas situações pretendidas. Isto implica observar o resultado obtido para os seis estados ( $X, Y, Z, \phi, \theta, \psi$ ) em relação ao referencial inercial  $R_{NED}$ .

Fazendo uso das técnicas tradicionais de visualização dos gráficos dos estados ao longo do tempo, verificou-se que os mesmos não eram muito descritivos da simulação ocorrida. Por isso, de modo a se poder observar concretamente a reacção do ALIV a determinadas simulações e objectivos, foi utilizado o simulador de voo *Opensource* FLIGHTGEAR [19].

De modo a incorporar o FLIGHTGEAR com a plataforma de simulação desenvolvida em SIMULINK, e com isso, poder visualizar o comportamento do ALIV em ambiente 3D fazendo uso dos grafismos do FLIGHTGEAR, foi utilizada a ferramenta *Aerospace Toolbox* [20] contida no MATLAB<sup>7</sup>.

Uma vez pretendermos seguir a trajectória do ALIV e ver a sua reacção ao longo da simulação, foi criado um modelo virtual desenvolvido em SOLIDWORKS [21] para incorporação no simulador. A Figura 24 ilustra o modelo construído em comparação com o modelo real.

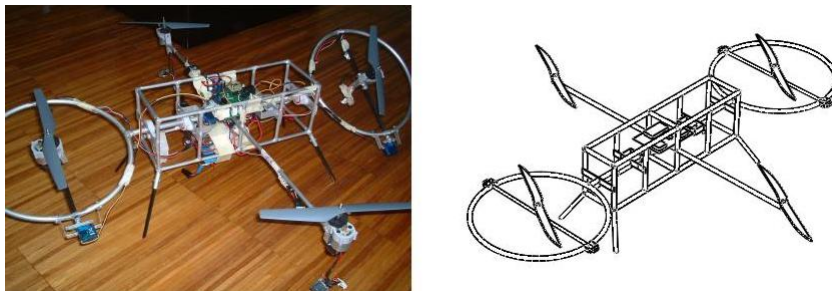


Figura 24: O ALIV real e virtual

Para a construção deste modelo foi tido como objectivo principal obter uma ideia clara da estrutura do ALIV não fazendo ênfase aos detalhes da electrónica existentes a bordo, sendo que os mesmos são desnecessários para a compreensão da visualização da posição e da atitude do ALIV. Ao longo do Apêndice E, podem ainda ser consultadas em pormenor as transformações necessárias para que o modelo inicialmente desenvolvido em SOLIDWORKS possa ser adoptado e correctamente animado pelo FLIGHTGEAR de acordo com a simulação que está a decorrer. A Figura 25 ilustra o FLIGHTGEAR quando o mesmo é iniciado com o UFO e durante uma fase da simulação. Nesta imagem pode ser observado que o modelo 3D assumido é o do ALIV e que este está alinhado de acordo com a posição inicial.

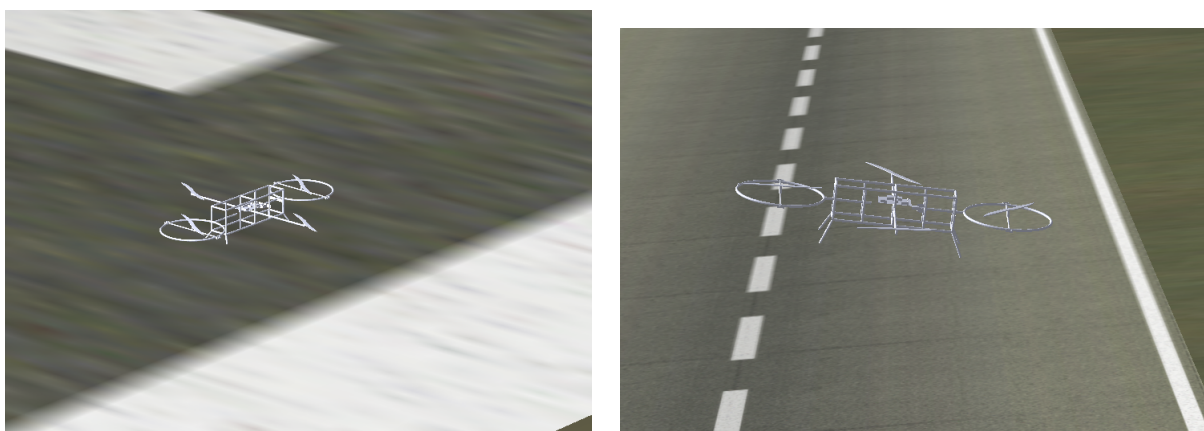


Figura 25: Simulação do ALIV no FLIGHTGEAR

<sup>7</sup>O Apêndice E contém a descrição do processo que tornou possível a interface entre o SIMULINK e o simulador de voo *Opensource* FLIGHTGEAR

## 6 Apresentação e discussão dos resultados

Nesta secção são apresentados os resultados obtidos no decorrer deste trabalho. De uma forma cronológica, os aspectos e os objectivos ultrapassados, são expostos como ponto de partida para as futuras adaptações e consequentes aproximações à realidade.

As discussões passam pelos aspectos mais relevantes no funcionamento de um Quadrirotor convencional desde a sua dinâmica ao tipo de missões a que se propõe, isto é, voo pairado e situações de vigilância. Inicialmente, e como não poderia deixar de ser, é validado o modelo matemático desenvolvido para a dinâmica do Quadrirotor, Secção 6.1. Depois, são utilizados os sensores para a realimentação dos motores através dos estados estimados, primeiro com um sensor de posição ideal para validação do modelo na Secção 6.2, e depois com o módulo do joystick na Secção 6.3.1, e por último com o módulo de visualização na Secção 6.3.2. Por fim, são discutidos os resultados finais deste trabalho com base no módulo integral na Secção 6.3.3.

### 6.1 Sistema em anel fechado ideal

Para a realização deste trabalho, e uma vez estar a ser utilizado um modelo de certa forma desconhecido, o primeiro objectivo a ser cumprido era sem dúvida a validação do modelo matemático e de toda a teoria a ele inerente. Sendo assim, de acordo com o modelo matemático desenvolvido na Secção 3, foi criada uma função que representasse a dinâmica do Quadrirotor, e com base no controlo moderno LQR (Secção 4.2), procedeu-se à construção da matriz de ganho  $K$  para a realimentação do sistema. O resultado foi a construção de um sistema em anel fechado característico do funcionamento de um Quadrirotor convencional controlado (Figura 26).

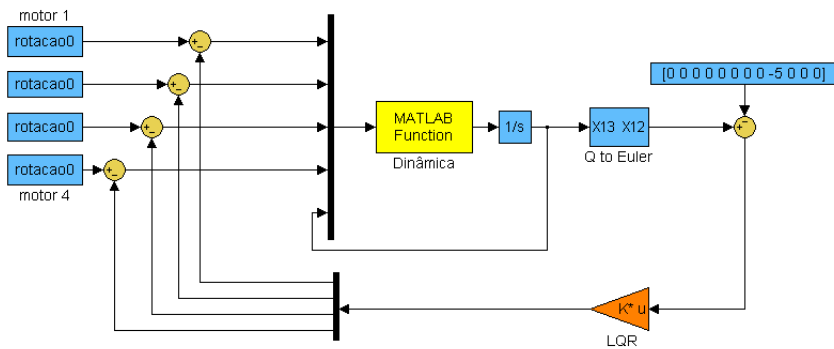


Figura 26: Sistema em anel fechado ideal

Como pode ser observado na figura, este primeiro sistema é simplesmente constituído por uma função que define a dinâmica de um Quadrirotor convencional apresentada ao longo da Secção 3, e por uma matriz de ganho  $K$  (controlo óptimo LQR), de modo a se conseguir a posição de estabilização pretendida. Inicialmente, estes motores possuem a rotação  $\omega_i$  necessária para que seja compensado o peso e seja igualada a força segundo a vertical causada pela gravidade, isto é, de acordo com a Secção 3.3:

$$F_{prop_i} = K_f \omega_i^2 = 1.1617 \times 10^{-5} \omega_i^2 \implies F_{prop_z} = -\sum_{i=1}^4 F_{prop_i} = 4.6468 \times 10^{-5} \omega_i^2 \quad (73)$$

$$\dot{w} = 0 \Leftrightarrow \frac{F_{propz}}{m} - S \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = 0 \Leftrightarrow \omega_0 = \sqrt{\frac{gm}{4.6468 \times 10^{-5}}} = \sqrt{\frac{9.81 \times 1.2}{4.6468 \times 10^{-5}}} = 503.3246 \text{ rad/s} \quad (74)$$

### 6.1.1 Cálculo da matriz de ganho $K$

Como foi apresentado na Secção 4 a matriz de ganho  $K$  pode ser obtida com base no comando *lqr* do MATLAB. Para isso, é apenas necessário definir as matrizes  $A$  e  $B$  que definem a dinâmica do sistema, as matrizes de ponderação  $Q$  e  $R$  que melhor se ajustam aos estados envolvidos durante a simulação, e o período de amostragem  $t_s$ .

De acordo de novo com a mesma secção, as matrizes  $A$  e  $B$  podem ser assumidas como matrizes Jacobianas das diferentes equações envolvidas em ordem a cada uma dos estados ou entradas. Por isso foi criado um ficheiro denominado *numerico.m* (Apêndice C) que realiza o cálculo tanto da matriz  $A$  como da matriz  $B$  para o caso da posição de referência escolhida. Assumindo que se pretende a estabilização do Quadrirotor a uma dada altitude, isto é  $X_0 = [u, v, w, p, q, r, e_x, e_y, e_z, \phi, \theta, \psi] = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ -h \ 0 \ 0 \ 0]$ , as matrizes  $A$  e  $B$  calculadas foram (em unidades SI):

$$A = \frac{\partial f_{dinamica}}{\partial X} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -9.81 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9.81 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \rightarrow \left\{ g = 9.81 \right. \quad (75)$$

$$B = \frac{\partial f_{dinamica}}{\partial U} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -0.0097 & -0.0097 & -0.0097 & -0.0097 \\ 0 & 0 & 0.1427 & -0.1427 \\ -0.1731 & 0.1731 & 0 & 0 \\ 0.0018 & 0.0018 & -0.0018 & -0.0018 \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \rightarrow \left\{ \begin{array}{l} \frac{2 \times K_f \times \omega_0}{m} = 0.0097 \\ \frac{2 \times K_f \times \omega_0 \times Ll}{I(1,1)} = 0.1427 \\ \frac{2 \times K_f \times \omega_0 \times Lc}{I(2,2)} = 0.1731 \\ \frac{2 \times K_f \times Kk \times \omega_0}{I(3,3)} = 0.0018 \end{array} \right. \quad (76)$$

Estas matrizes estão de acordo com as fórmulas da Secção 4.1 confirmando assim a validade do procedimento elaborado. Quanto às matrizes de ponderação  $Q$  e  $R$  foram assumidas de acordo com

o princípio de Bryson (Secção 4.2), o resultado foi o seguinte:

$$\begin{cases} u_{max} = v_{max} = w_{max} = 0.2 \text{ m/s} & p_{max} = q_{max} = r_{max} = 0.063 \text{ rad/s} \\ x_{max} = y_{max} = z_{max} = 0.4 \text{ m} & \phi_{max} = \theta_{max} = \psi_{max} = 0.049 \text{ rad} \\ u_{1_{max}} = u_{2_{max}} = u_{3_{max}} = u_{4_{max}} = 3.16 \text{ rad/s} \end{cases} \quad (77)$$

$$Q_{ii} = [25 \ 25 \ 25 \ 250 \ 250 \ 250 \ 6.25 \ 6.25 \ 6.25 \ 418 \ 418 \ 418] \quad R_{ii} = [0.1 \ 0.1 \ 0.1 \ 0.1] \quad (78)$$

### 6.1.2 Apresentação dos resultados para o anel fechado ideal

O objectivo dos resultados apresentados nesta secção é de demonstrar que a dinâmica assumida para o caso de um Quadrirotor convencional está de acordo com a teoria avançada na Secção 3.

- Controlo da posição transversal segundo  $xy$  após estabilização a uma dada altitude

De modo a não prolongar muito estes resultados preliminares, foi realizada uma trajectória que envolvesse os três eixos inerciais. Foi escolhida assim uma simulação que passasse pela estabilização a uma dada altitude ( $5 \text{ m}$ ) e pela deslocação transversal para  $x = 3 \text{ m}$  e depois para  $y = 3 \text{ m}$  através da alteração do vector de referência  $X_0$ . O resultado é o apresentado na Figura 27.

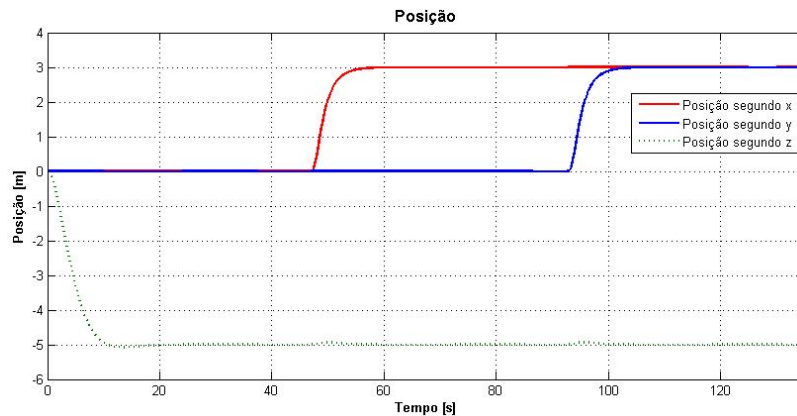


Figura 27: Posição para deslocação a  $5 \text{ m}$  de altitude e  $3 \text{ m}$  segundo  $x$  e  $y$  (anel fechado ideal)

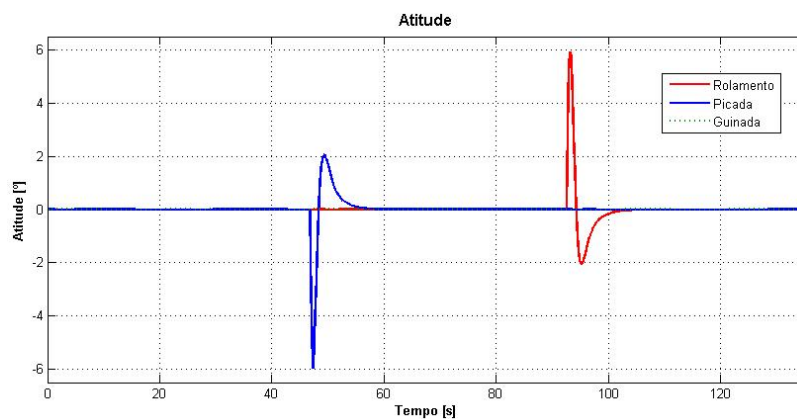


Figura 28: Atitude para deslocação a  $5 \text{ m}$  de altitude e  $3 \text{ m}$  segundo  $x$  e  $y$  (anel fechado ideal)



Para além da posição vertical, também a posição transversal pretendida foi conseguida com sucesso, mantendo-se o Quadrirotor estabilizado. A Figura 28 representa a variação da atitude. A azul pode ser observado o movimento de picada quando se pretende obter a deslocação segundo  $x$ , e a vermelho o movimento de rolamento quando o deslocamento é para ser efectuado segundo  $y$ . Ambos os resultados estão de acordo com os referenciais definidos na Figura 7, sendo os valores bastante parecidos pela quase simetria do Quadrirotor adoptado.

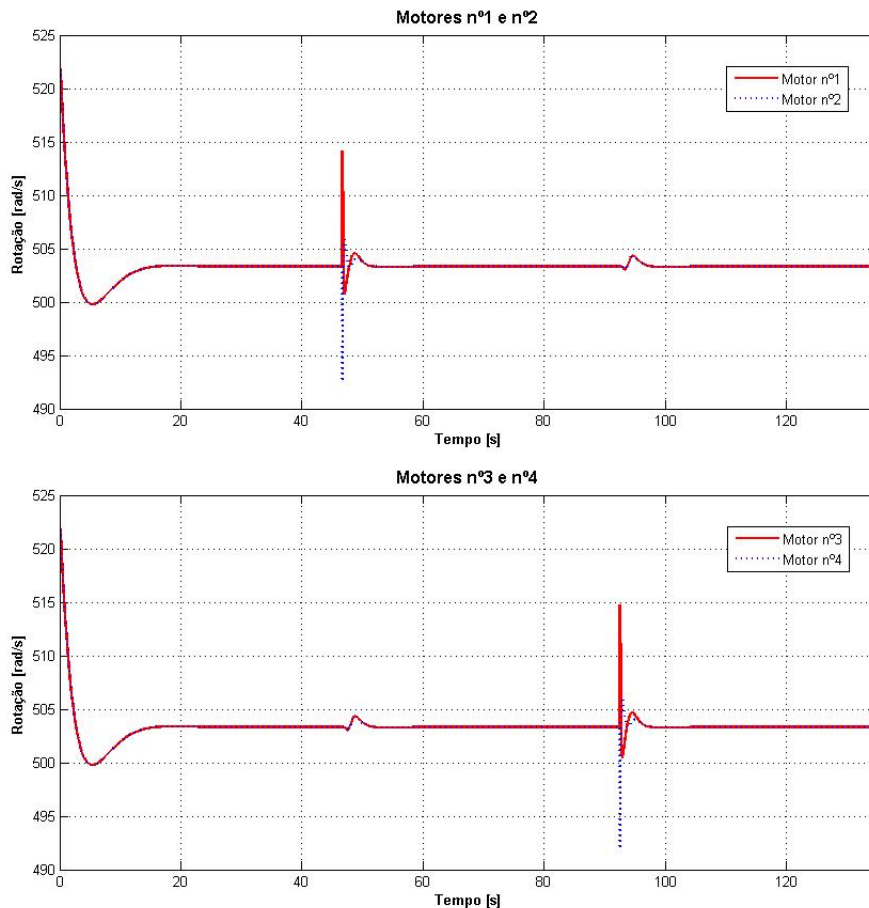


Figura 29: Rotação nos motores na deslocação a 5 m de altitude e 3 m segundo  $x$  e  $y$  (anel fechado ideal)

Em termos de motores, na primeira fase estamos a actuar no conjunto dos quatro rotores, de modo a que o Quadrirotor suba ou desça. Na Figura 29 pode ser observado que existe uma fase inicial em que existe um aumento repentino de todos os motores de modo a aumentar a altitude do Quadrirotor, seguida de uma fase de diminuição até que os mesmo atingam a velocidade característica que compense o peso corpo. Na segunda fase da simulação, estamos a actuar sequencialmente nos dois pares de rotores.

Primeiramente no par de rotores 3&4, aumentando o rotor 3 e diminuindo o rotor 4 na mesma ordem de grandeza de forma a manter constante a força vertical, e posteriormente no par de rotores 1&2, aumentando o 1 e diminuindo o 2 na mesma ordem de grandeza pelas razões já adiantadas, Figura 29.

Mais uma vez os resultados estão de acordo com o modelo desenvolvido para a dinâmica do Quadrirotor convencional, pelo que, apesar da simplicidade do sistema desenvolvido até aqui, este

revela ser de extrema importância para a continuação da implementação de modelos mais complexos.

## 6.2 Sistema com sensor de posição ideal

Após a verificação da dinâmica, os objectivos passam pela utilização do filtro de Kalman para a realimentação das entradas dos motores com base nos valores estimados dos estados, conferindo uma maior realidade ao sistema. Para isso, numa primeira fase, foi assumido um sensor de posição ideal dado o ALIV não possuir qualquer sensor capaz de observar a posição. A Figura 30 representa o modelo desenvolvido durante esta fase do projecto.

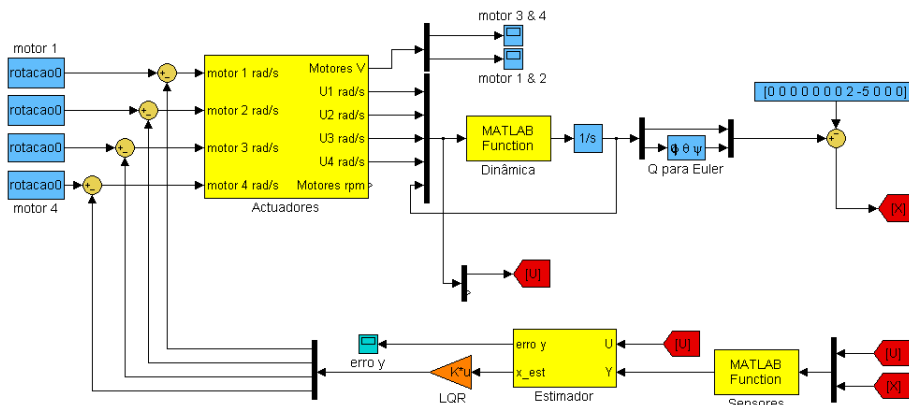


Figura 30: Sistema com sensor de posição ideal

Como pode ser constatado em relação ao modelo inicial, desta vez já estão incluídos os actuadores (Secção 3.3), e os blocos responsáveis pela construção da estimativa dos estados, os sensores (Secção 3.5) e o estimador (Secção 4.3).

### 6.2.1 Construção do estimador

Através do comando *kalm* do MATLAB é possível obter a construção de um estimador com base no filtro de Kalman em sistema espaço de estados (Secção 4.3). Para isso, é necessário fornecer ao comando o sistema em espaço de estados a ser observável, as matrizes espectrais  $Q_K$  e  $R_K$ , e o período de amostragem  $t_s$ . Desta vez, ao contrário do que se sucedeu para o caso do cálculo da matriz de ganho  $K$  (Secção 4.2) o sistema inicial em sistema espaço de estados é formado por:

$$\begin{cases} \dot{x} = Ax + Bu + Gw \\ y = Cx + Du + v \end{cases} \quad (79)$$

no entanto a teoria enumerada para o cálculo das matrizes numéricas  $A$  e  $B$  (Secção 6.1) aplica-se na íntegra ao cálculo das matrizes  $C$  e  $D$ , desta vez, utilizando as equações fornecidas pelos modelos matemáticos desenvolvidos para os sensores. Assim, mais uma vez as matrizes  $C$  de dimensão  $[12 \times 12]$  e  $D$  de  $[12 \times 4]$  podem também ser definidas como matrizes Jacobianas das equações definidas pelos sensores em ordem a cada um dos estados e entradas dos motores intervenientes na simulação, respectivamente. O resultado foi:

$$C = \frac{\partial f_{\text{sensores}}}{\partial X} = \begin{bmatrix} 0 & \cdots & 0 & 0 & 0 & 0 & 0 & -19.62 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 19.62 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cdots & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \rightarrow \begin{cases} 2 \times g = 19.62 \end{cases} \quad (80)$$

$$D = \frac{\partial f_{\text{sensores}}}{\partial U} = \begin{bmatrix} 0 & 0 & 0.0087 & -0.0087 \\ -0.0071 & 0.0071 & 0 & 0 \\ -0.0097 & -0.0097 & -0.0097 & -0.0097 \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{cases} \frac{2 \times K_f \times \omega_0}{m} = 0.0097 \\ \frac{2 \times K_f \times \omega_0 \times Ll}{I(1,1)} \times ddy = 0.0071 \\ \frac{2 \times K_f \times \omega_0 \times Lc}{I(2,2)} \times ddz = 0.0087 \end{cases} \quad (81)$$

As matrizes  $A$  e  $B$  são idênticas dado que a dinâmica não se alterou. Através da atribuição de valores razoáveis às matrizes  $G$ ,  $H$ ,  $Q_K$  e  $R_K$  pode ser finalmente calculado o estimador:

$$\begin{cases} G = eye(12, 6); & H = 0 \times eye(9, 6); \\ Q_{K_{ii}} = [dd \ dd \ dd \ dd \ dd \ dd] & R_{K_{ii}} = [dd \ dd \ dd \ dd \ dd \ dd \ dd \ dd \ dd], \quad dd = 1 \times 10^{-4} \end{cases} \quad (82)$$

$$sys = ss(A, [B \ G], C, [D \ H]); \quad [KEST, L, P, M] = kalmd(sys, Q_K, R_K, t_s);$$

Em que  $dd$  corresponde a uma perturbação associada ao ruído. Depois de se ter a construção do estimador, o mesmo foi inserido no diagrama de blocos apresentado na Figura 30. A Figura 31 ilustra o interior desse bloco. O sistema em espaço de estados para o caso do estimador é o resultado do  $KEST$  como explicado na Secção 4.3.

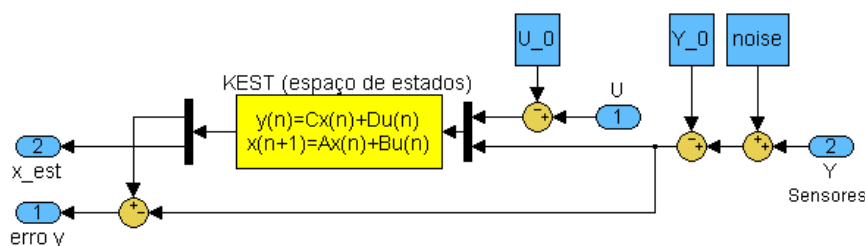


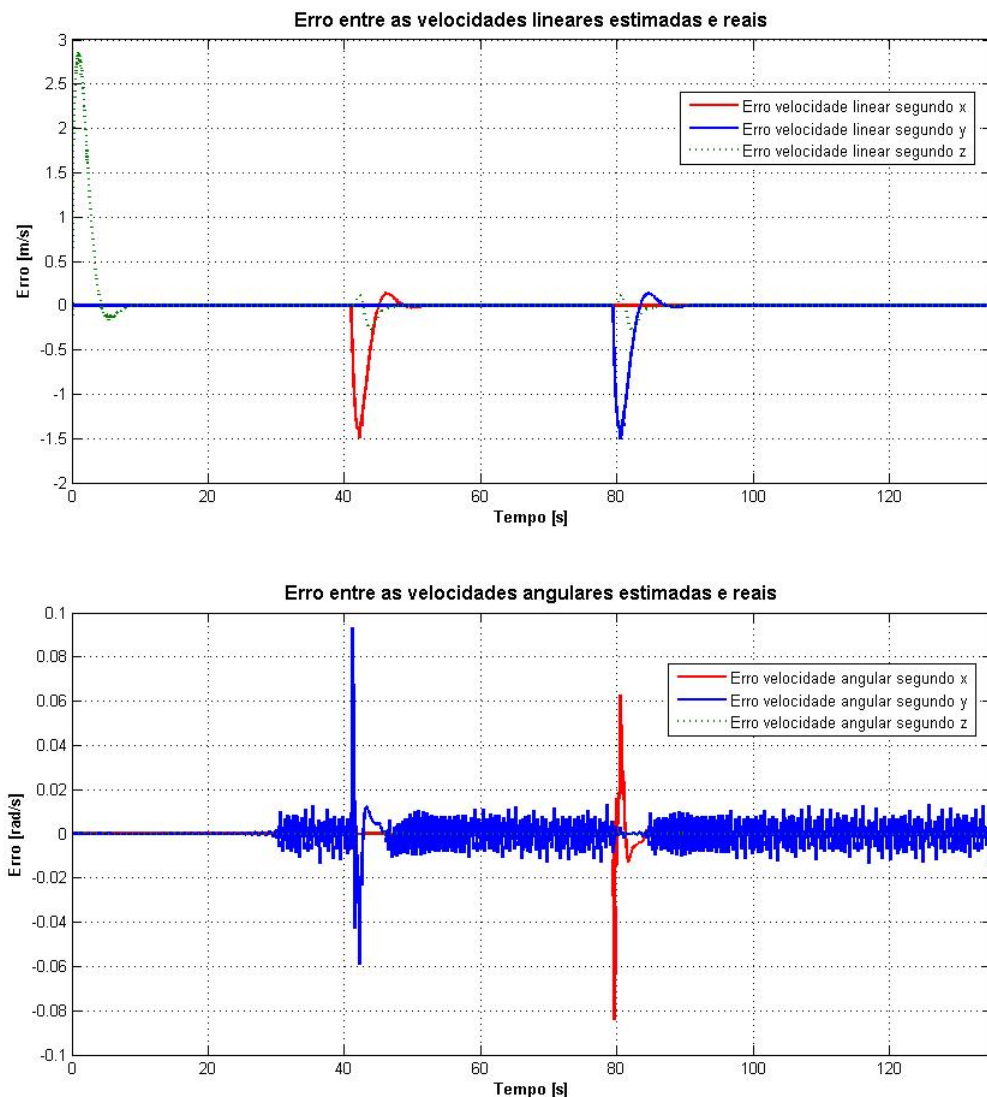
Figura 31: Estimador em diagrama de blocos para o caso do sistema com sensor de posição ideal

## 6.2.2 Apresentação dos resultados para o sensor de posição ideal

Para se obter a comprovação dos resultados alcançados, foi decidido realizar uma simulação idêntica à desenvolvida para a situação de anel fechado de modo a poder verificar se o comportamento através da utilização da estimativa dos estados era semelhante.

- Erro associado aos estados

Como foi referido ao longo da Secção 4.3, desta vez estamos a realizar a realimentação dos motores através das estimativas dos estados obtidos com base nas saídas disponíveis dos sensores e das capacidades do filtro de Kalman em recuperar com base nisso os estados pretendidos. Como tal, é objectivo deste primeiro modelo desenvolvido, conseguir que as estimativas dos estados sejam aproximadamente iguais aos estados reais de forma a confirmar o modelo. Isto é, como foi explicado na Secção 4.3, que com o passar do tempo o erro  $\tilde{x} = x - \hat{x}$  seja nulo, e que durante esse tempo o Quadrirotor esteja em condições de conseguir atingir os objectivos propostos sem que para isso se perda o controlo ou se atinjam valores inconvenientes de operação. Para a trajectória até então vinda a ser adoptada como referência, os resultados obtidos relativamente ao erro associado entre o estado estimado e o real foram os seguintes, Figura 32.



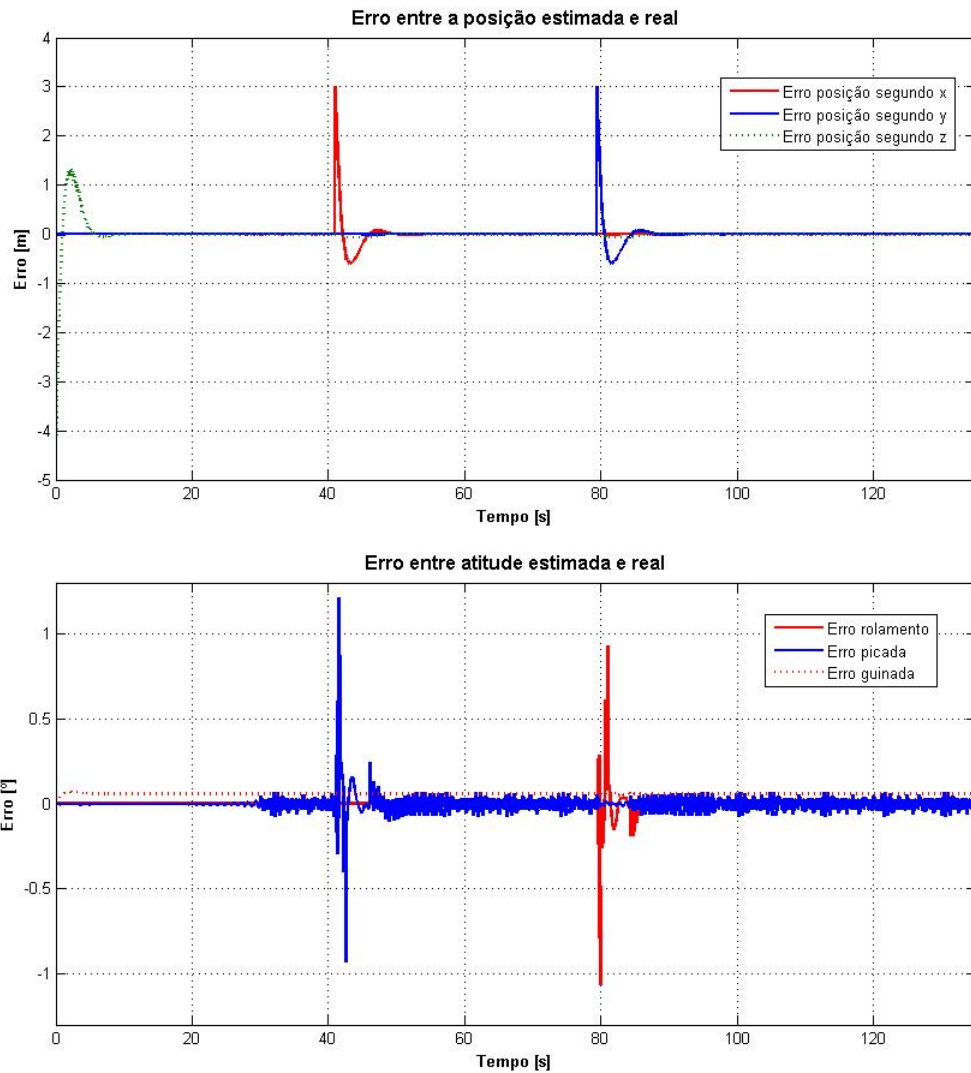


Figura 32: Erro entre os estados reais e estimados, sensor de posição ideal

Através destes gráficos é possível observar que o modelo tem o comportamento pretendido, dado que em qualquer um dos estados o erro associado tende para zero após ter ocorrido uma mudança na referência. Dado que  $x = X - X_0$  é de esperar que sempre que se alterar a posição de referência o erro inicie-se nesse mesmo valor e tenda depois para zero. Por isso é que no momento da alteração da posição de referência transversal, o erro passa para o valor da posição desejada, neste caso  $3\text{ m}$ . Pela mesma razão, os restantes erros  $\tilde{u}$ ,  $\tilde{v}$ ,  $\tilde{w}$ ,  $\tilde{p}$ ,  $\tilde{q}$ ,  $\tilde{r}$ ,  $\tilde{\phi}$ ,  $\tilde{\theta}$  e  $\tilde{\psi}$  apresentam um pico na altura dessa mudança da posição de referência.

- Controlo da posição transversal segundo  $xy$  após estabilização a uma dada altitude

Depois da observação do andamento em termos do erro associado entre os estados reais e os estimados, procede-se à ilustração dos estados globais de forma a confirmar os resultados previstos anteriormente. adiantados através simplesmente da observação dos erros.

Para o caso da posição global o resultado foi o seguinte (Figura 33).

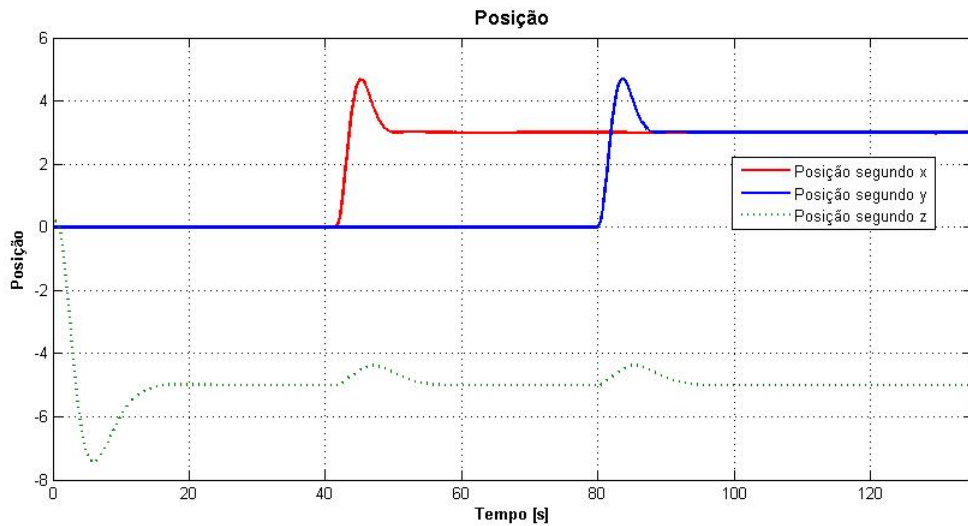


Figura 33: Posição para deslocação a 5 m de altitude e 3 m segundo  $x$  e  $y$ , sensor de posição ideal

Como pode ser observado na figura anterior, e comparando o resultado com o obtido na Figura 27, apesar de no final a posição ser idêntica e estar estabilizada, até alcançá-la, o Quadrirotor reagiu de forma diferente. Esta situação é de todo normal e aceitável, dado que desta vez estamos a realizar uma realimentação através da estimativa dos estados e não num anel fechado ideal, não sendo os valores atingidos um inconveniente à modulação até aqui realizada. Esta situação podia ser contornada com a diminuição da ponderação dos estados referentes à posição na matriz  $Q$ , no entanto, essa hipótese revelou uma maior dificuldade por parte do Quadrirotor em se estabilizar.

Em relação à atitude o resultado foi o seguinte, Figura 34.

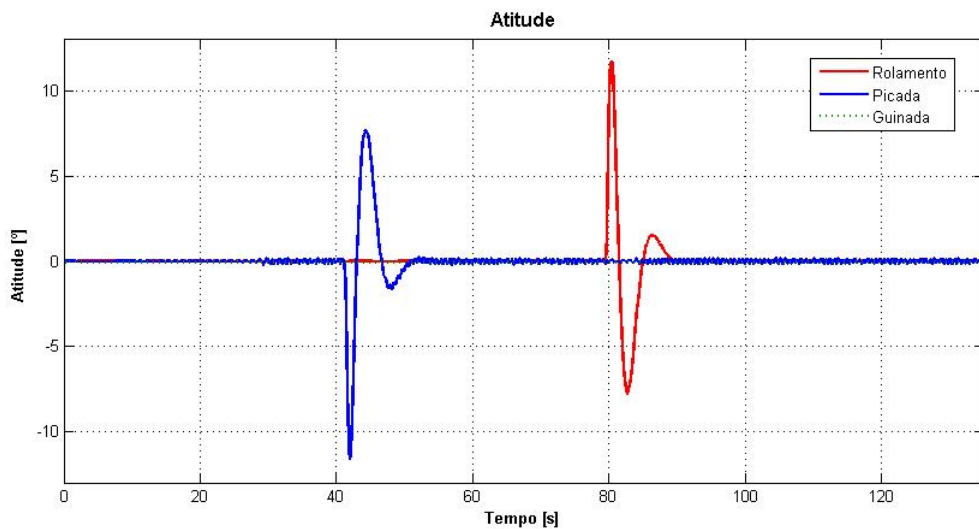


Figura 34: Atitude para deslocação a 5 m de altitude e 3 m segundo  $x$  e  $y$ , sensor de posição ideal

O que importa aqui destacar comparando os resultados com a situação em anel fechado representado na Figura 28, é que desta vez, apesar da posição de referência ser à mesma distância, o Quadrirotor precisar de uma atitude mais acentuada para conseguir realizar a mesma trajectória. Enquanto que na situação anterior atingia uma picada e um rolamento na ordem dos  $6^\circ$ , desta vez, para

atingir a mesma distância precisa aproximadamente do dobro,  $12^{\text{º}}$ , muito provavelmente associado ao erro inicial existente na atitude, Figura 32. Isto faz estarmos em condições de assumir que o Quadrirotor é mais limitativo na sua gama de operatividade do que o esperado inicialmente. Apesar disso, o objectivo de alcançar uma determinada posição estabilizada é conseguido e está em consonância com os resultados obtidos na situação ideal em anel fechado. O mesmo acontece para o comportamento dos motores intervenientes ao longo da simulação, pelo que, por motivos de interesse e economia de espaço, as figuras referentes a essa situação não vão ser apresentadas.

- Estabilização numa posição bastante diferente da inicialmente conhecida por parte do estimador

De modo a finalizar esta secção, é ainda apresentada uma simulação para demonstrar as capacidades do Quadrirotor e a fiabilidade do modelo desenvolvido até ao momento. Para isso, foi colocado o Quadrirotor numa posição diferente da situação inicial, em que todos os estados globais são nulos, e é pedido uma nova posição de referência. Neste tipo de situação o estimador não tem qualquer informação sobre a posição de partida, logo é de esperar erros iniciais algo acentuados. Adoptando uma condição inicial definida por:

$$X_i = [u \quad v \quad w \quad p \quad q \quad r \quad x \quad y \quad z \quad q_0 \quad q_1 \quad q_2 \quad q_3]$$

$$X_i = [0 \quad 0.2 \quad 0 \quad 0 \quad 0 \quad 0 \quad 2 \quad -3 \quad -2 \quad 1 \quad 0.2 \quad 0 \quad 0.15] \quad (83)$$

e pretendendo uma condição final estabilizada:

$$X_0 = [u \quad v \quad w \quad p \quad q \quad r \quad x \quad y \quad z \quad \phi \quad \theta \quad \psi]$$

$$X_0 = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 2 \quad -5 \quad 0 \quad 0 \quad 0] \quad (84)$$

o resultado entre a posição e a atitude estimada e real está representado na Figura 35 e na Figura 36.

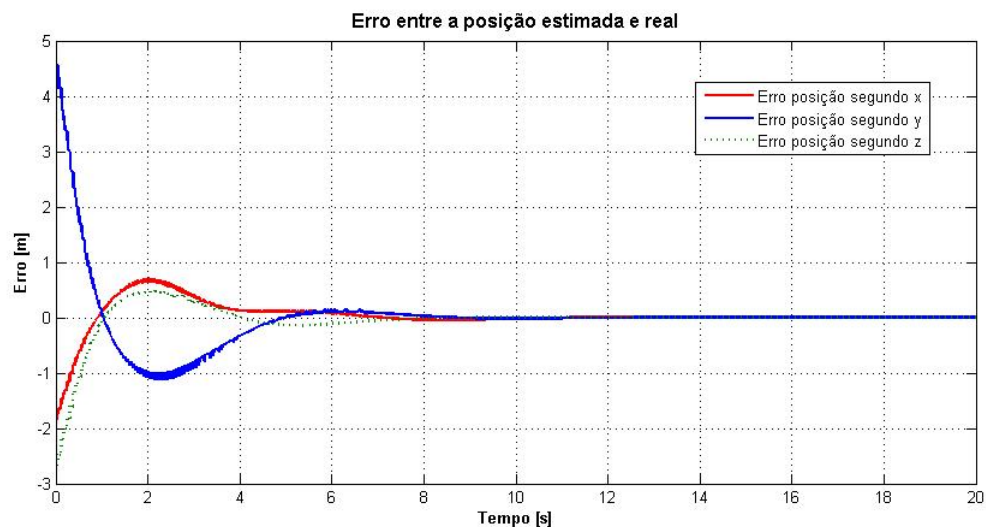


Figura 35: Erro entre a posição estimada e real, sensor de posição ideal

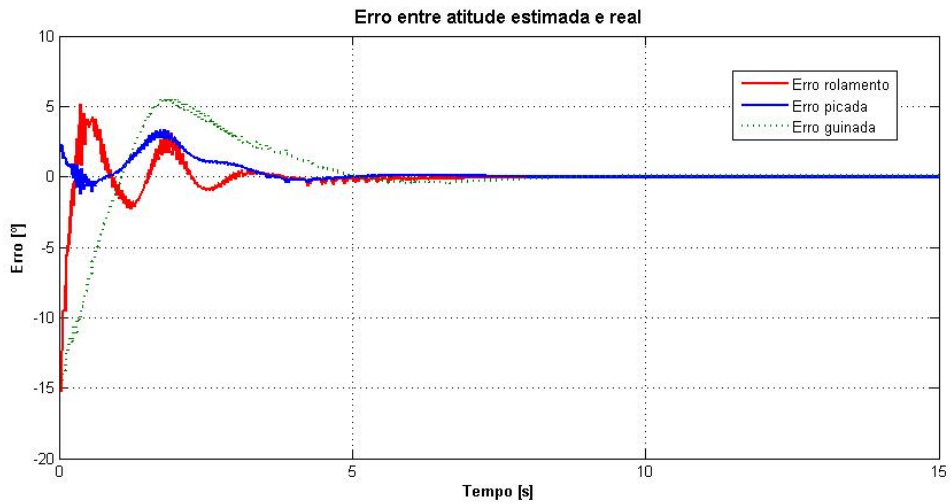


Figura 36: Erro entre a atitude estimada e real, sensor de posição ideal

Como se pode constatar o estimador conseguiu anular o erro inicial, pelo que é normal que tenha conseguido atingir a posição pretendida, Figura 37.

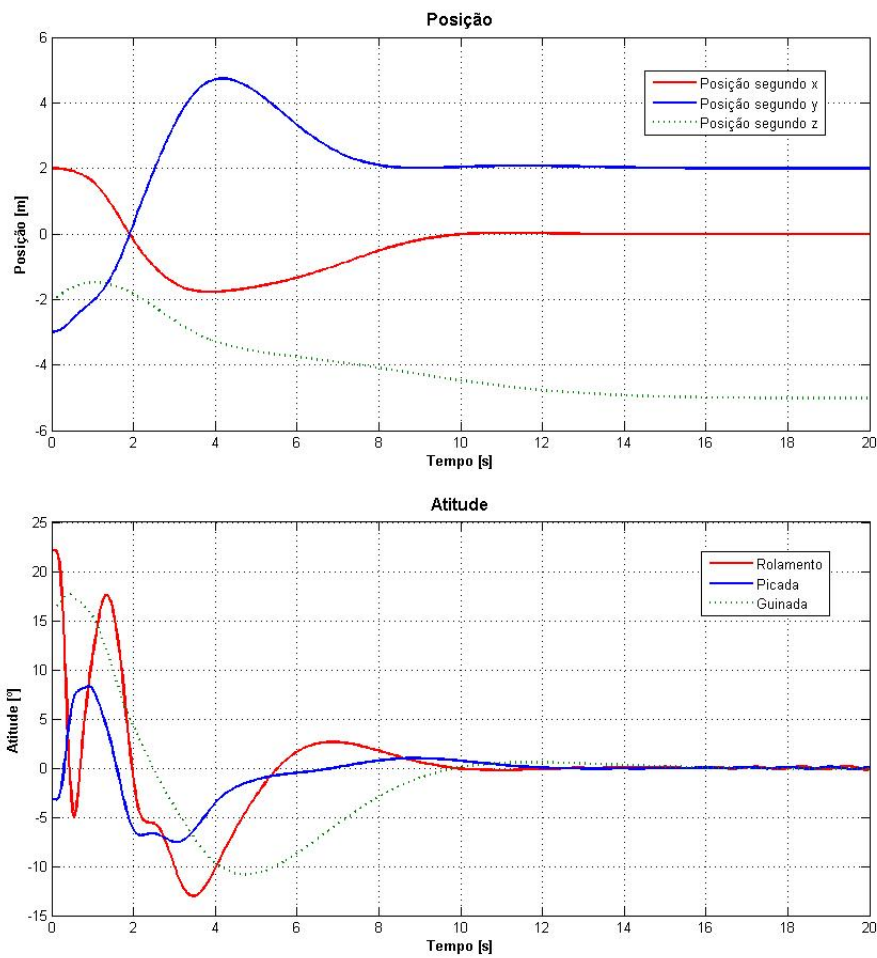


Figura 37: Posição e atitude para uma situação limite, sensor de posição ideal



Através dos gráficos anteriores é possível observar a rapidez com que o Quadrirotor conseguiu estabilizar numa posição final bastante diferente da inicialmente conhecida, isto revela ser de extrema importância na comprovação do modelo e da teoria adiantada nos resultados anteriores, estando assim reunidas as condições para o ponto de partida dos módulos finais.

### 6.3 Módulos finais

Até esta altura, conseguiram ser verificados os princípios fundamentais pelos quais se regem os módulos desenvolvidos. Isto é, tanto a dinâmica como o estimador revelaram estar de acordo com o esperado, possibilitando assim a implementação dos módulos finais característicos do MALIV - My Autonomous Locomotion Individual Vehicle, que consiste num modelo adaptado do ALIV original em que apenas é feito, uso das propriedades de um Quadrirotor convencional. Com base nos resultados obtidos nas duas últimas secções, estamos assim em condições de aplicar as alterações necessárias para a implementação dos módulos finais.

A partir desta altura, os resultados obtidos são apresentados com base na simulação em tempo real, usufruindo das capacidades do MATLAB e do SIMULINK para o efeito, Secção 5.2. O modelo apresentado na Figura 38 corresponde ao modelo final, um módulo integrado, onde é possível realizar a permuta entre os dois módulos inicialmente concebidos: o módulo com joystick e o módulo de visualização, Secção 5.1.

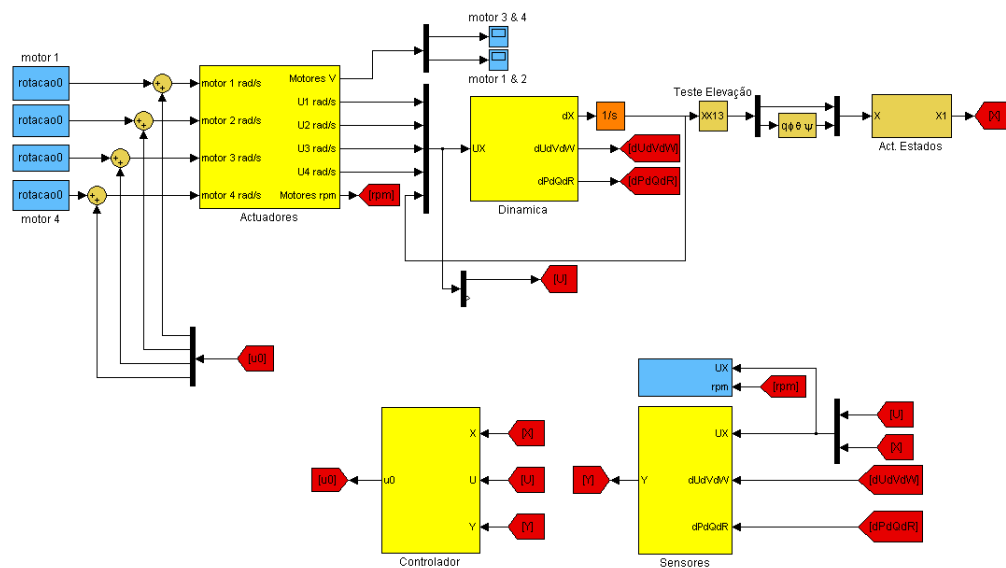


Figura 38: Diagrama de blocos do módulo integrado

Após a validação independente dos dois módulos anteriores, é apresentado os resultados do módulo integrado, responsável pela simulação final adoptada para o controlo versátil e integral do MALIV<sup>8</sup>.

<sup>8</sup>Nesta secção, à semelhança do que se vai passar nas seguintes, será dado ênfase aos resultados obtidos e não ao modo como o módulo foi construído, para isso é que foi criada a Secção 5.1 de explicação da filosofia adoptada em cada um dos módulos principais.

### 6.3.1 Módulo com joystick

O primeiro módulo a ser testado é o módulo com joystick. Este módulo caracteriza-se por fazer uso das saídas disponíveis nos sensores existentes a bordo do ALIV original e pela utilização de um joystick para o controlo da sua posição, Secção 5.1.1.

Com base na Figura 20, exemplificativa da filosofia adoptada na construção deste módulo, o estimador é composto pelo filtro de Kalman à semelhança do que aconteceu para o caso do sensor de posição ideal, Secção 6.2, mas desta vez, apenas estão disponíveis as saídas provenientes do acelerómetro e do magnetómetro, Figura 39. Isto faz com que para a construção do novo LQG, tanto as novas matrizes  $A$   $[6 \times 6]$ , e  $B$   $[6 \times 4]$ , sejam agora formadas pelo conjunto de linhas e colunas das matrizes originais referentes às seis variáveis de estado  $[p \ q \ r \ \phi \ \theta \ \psi]$ , e as matrizes  $C$   $[6 \times 6]$  e  $D$   $[6 \times 4]$  formadas pelo conjunto de linhas e colunas das matrizes originais referentes às saídas do acelerómetro e do magnetómetro<sup>9</sup>.

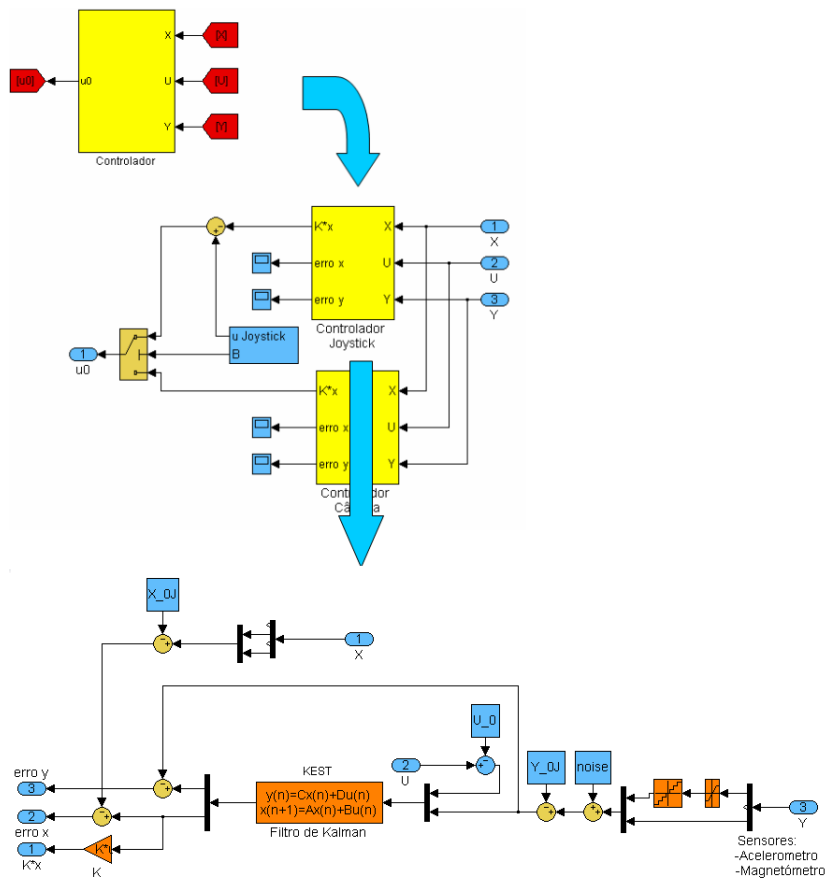


Figura 39: Controlador em diagrama de blocos adoptado para o módulo com joystick

A nova matriz de ganho  $K$  e o filtro de Kalman, são calculados com base nos mesmos comandos originais e da descrição anterior:

$$\begin{cases} K = lqrd(A, B, Q, R, t_s) \\ sys = ss(A, [B \ G], C, [D \ H]); \end{cases}$$

<sup>9</sup>A construção destas matrizes pode ser acompanhada no ficheiro *controlo.m* inserido no Apêndice C

Com desta vez:

$$\left\{ \begin{array}{ll} Q_{ii} = 1 \times 10^{-2} [364 \ 364 \ 364 \ 364 \ 364 \ 364] & R_{ii} = [10 \ 10 \ 10 \ 10] \\ p_{max} = q_{max} = r_{max} = 0.52 \text{ rad/s} & u_{1_{max}} = u_{2_{max}} = u_{3_{max}} = u_{4_{max}} = 0.32 \text{ rad/s} \\ \phi_{max} = \theta_{max} = \psi_{max} = 0.52 \text{ rad} & \\ Q_{K_{ii}} = [dd \ dd \ dd \ dd \ dd \ dd] & R_{K_{ii}} = [dd \ dd \ dd \ dd \ dd \ dd] \\ G = eye(6, 6) & H = 0 \times eye(6, 6); \end{array} \right. \quad (85)$$

Depois da realimentação, é exercido o controlo da posição através do joystick como explicado ao longo da Secção 5.1.1 e ilustrado na Figura 20.

Uma vez o joystick só possuir três graus de liberdade, teve de se proceder à escolha de três dos quatro estados de controlo  $[\dot{U} \ \dot{V} \ \dot{W} \ \dot{R}]$ . A escolha recaiu na eliminação do último estado. Sendo assim, ao movimento de rolamento do joystick está associado a deslocação do MALIV ao longo do eixo  $x$ , e ao movimento de picada a deslocação segundo o eixo  $y$ . Como é constatado, esta convenção não está de acordo com o princípio de funcionamento do Quadrirotor adoptado, mas revelou ser uma maneira mais eficaz no controlo da posição do MALIV através do joystick. Assim quando o joystick se desloca ao longo de um dos seus eixos o movimento do MALIV processa-se ao longo do mesmo eixo (Apêndice D).

- **Apresentação dos resultados**

De modo a ilustrar a validade do modelo desenvolvido, optou-se por definir uma trajectória e verificar se se conseguia efectua-la com o controlo do joystick. Isto é, procurou ser verificado a estabilidade e versatilidade do MALIV neste módulo de simulação uma vez já se ter confirmado que tanto a dinâmica como o estimador faziam o Quadrirotor reagir como o esperado.

- Controlo da posição transversal segundo  $xy$  após estabilização a uma dada altitude

Esta simulação passou por realizar pequenos ajustes no movimento do joystick, na totalidade dos seus três canais, para conseguir observar a alteração da posição vertical e horizontal ao longo do tempo, isto é, adoptando novas posições de referência. Para sintetizar os resultados, procedeu-se à elaboração de uma trajectória ao longo dos três eixos inerciais. Depois da estabilização vertical, foram realizadas deslocações transversais para ver como o MALIV reagia.

Procedeu-se então à simulação da seguinte trajectória:

1. Elevação do MALIV até uma altitude estabilizada na ordem dos  $12 \text{ m}$  de altitude
2. Manter a posição de altitude durante aproximadamente  $30 \text{ s}$
3. Deslocar o MALIV para a posição de  $y = 12 \text{ m}$
4. Permanecer em  $y = 12 \text{ m}$  e deslocar o MALIV para  $x = -12 \text{ m}$
5. Manter essa posição durante aproximadamente  $25 \text{ s}$

6. Voltar à posição inicial com  $x = y = 0$
7. Descer até ao solo com velocidades moderadas

A Figura 40 procura ilustrar o resultado da posição do MALIV em termos de  $xyz$  ao longo do tempo durante a simulação:

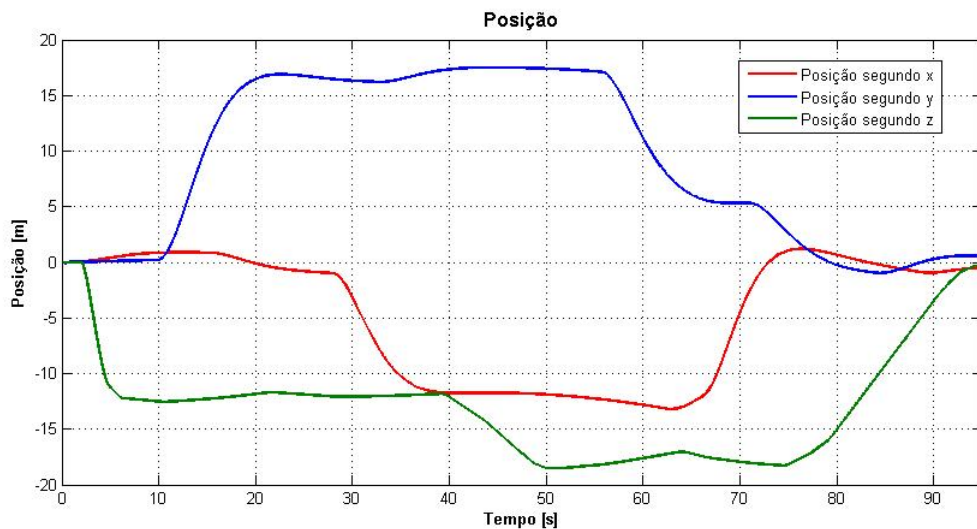


Figura 40: Posição  $xyz$  ao longo da segunda trajectória

Um dos inconvenientes deste módulo de simulação, como já foi referido ao longo deste trabalho, consiste na necessidade do constante ajuste por parte do utilizador da posição de referência pretendida. Isto faz com que neste módulo não sejam de esperar posições de referência claramente estabilizadas, no entanto, e como pode ser constatado na Figura 40, consegue-se realizar as trajectórias pretendidas sem que os valores oscilem muito.

A Figura 41 ilustra os ajustes referidos de modo a possibilitar a permanência ao longo das posições de referência pretendidas.

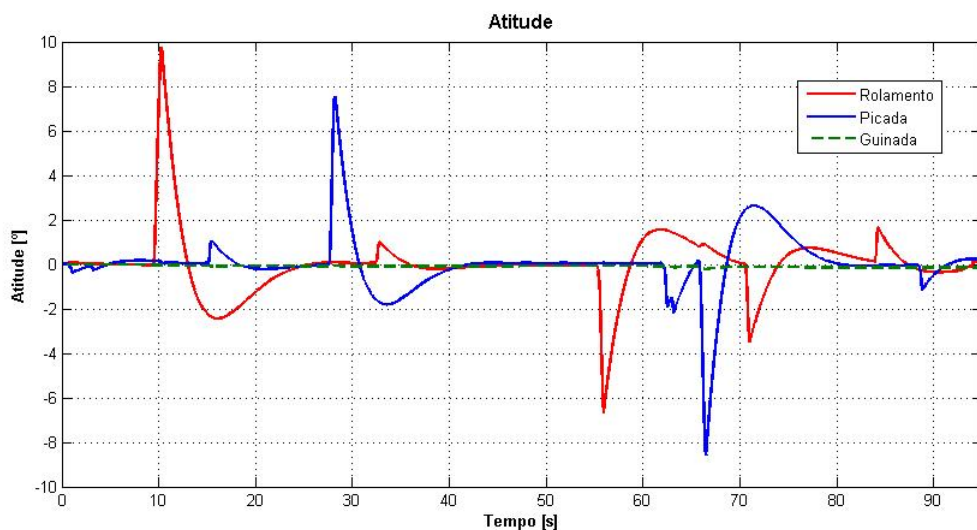


Figura 41: Atitude ao longo da segunda trajectória

Aqui podem ser constatados os movimentos de rolamento e picada necessários para a deslocação transversal à semelhança do que tem vindo a ser demonstrado, mas desta vez associados a movimentos do joystick. Foram alcançados aproximadamente valores de  $10^\circ$  para se obter uma trajectória de mais ou menos  $10\text{ m}$  de distância. Este valor até é inferior ao caso apresentado para o sensor de posição ideal, em que o Quadrirotor precisava de mais de  $10^\circ$  de rotação para atingir os  $3\text{ m}$  de distância. Isto deve-se ao facto, de anteriormente trabalharmos directamente na posição de referência, e não alterarmos gradualmente a velocidade transversal como o caso deste módulo. Neste aspecto este modelo revela ser menos limitativo. Em relação aos motores, para a obtenção dos movimentos transversais e como tem vindo a ser referido temos de actuar nos pares de rotores. Na Figura 42 podem ser observados em detalhe os diferenciais existentes entre o mesmo par de rotores para que as posições transversais sejam obtidas. Optou-se por este sistema de visualização uma vez a rotação necessária para as deslocações verticais ser consideravelmente maior. Assim, consegue-se identificar e verificar o correcto funcionamento dos mesmos. Isto vem provar a extrema susceptibilidade do MALIV em adoptar posições estabilizadas, já que um diferencial reduzido no mesmo par de rotores provoca deslocações horizontais algo consideráveis.

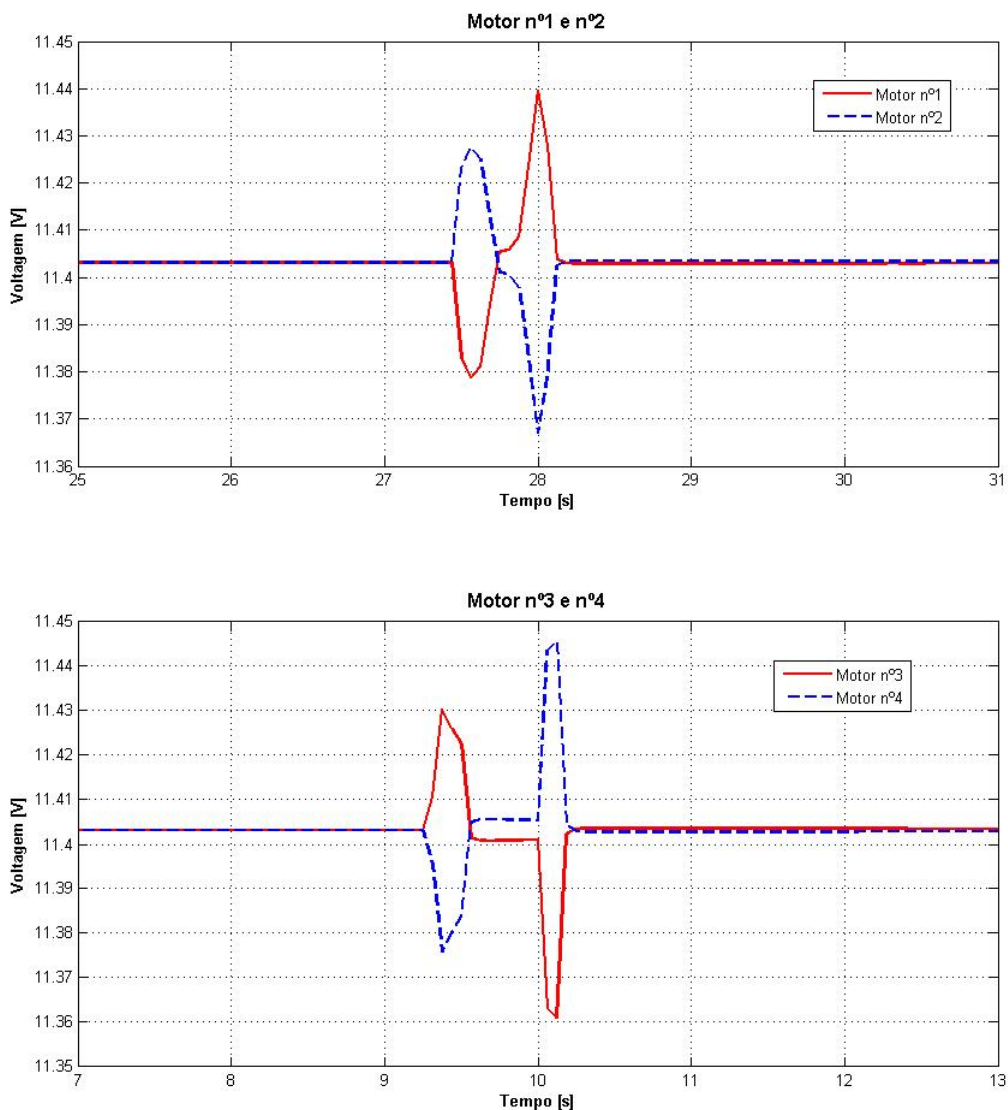


Figura 42: Voltagem aplicada aos motores ao longo da segunda trajectória

Como pode ser observado, por volta dos 27.5 s existe um aumento de voltagem no motor n<sup>o</sup>2 e uma diminuição no motor n<sup>o</sup>1 na mesma ordem de grandeza (Figura 42) assim é obtido um deslocamento negativo segundo  $x$ , Figura 40. O mesmo acontece por volta dos 9 s onde desta vez para se obter uma deslocação positiva segundo  $y$  (Figura 40) actuou-se no par de rotores n<sup>o</sup>3 e n<sup>o</sup>4, aumentando a tensão do motor n<sup>o</sup>3 e diminuindo a do motor n<sup>o</sup>4 (Figura 42). Isto foi modelado através da utilização de um bloco de derivada associado ao movimento do joystick. Após esta fase, existe a provocação de uma aceleração idêntica mas de sentido contrário para que o MALIV se consiga estabilizar, que ocorre mais ou menos nos 11.4 V, Figura 42, que equivale aos:

$$\frac{503.3246}{K_v} + 9 = 11.4032 V$$

onde  $K_v$  corresponde ao factor de conversão entre tensão e velocidade e 9 ao fim da zona morta do motor adoptado, Secção 3.3. Em termos de altitude não se procedeu da mesma maneira por inaptidão dos resultados com esse sistema, sendo assim, a aceleração vertical é sempre anulada com movimentos de guinada por parte do joystick, contrários aos anteriormente realizados.

Em relação às velocidades, e devido aos movimentos transversais, as velocidades transversais  $U$  e  $V$  apresentam valores na ordem dos 3 m/s, enquanto que a velocidade vertical  $W$  ronda os 6 m/s, Figura 43.

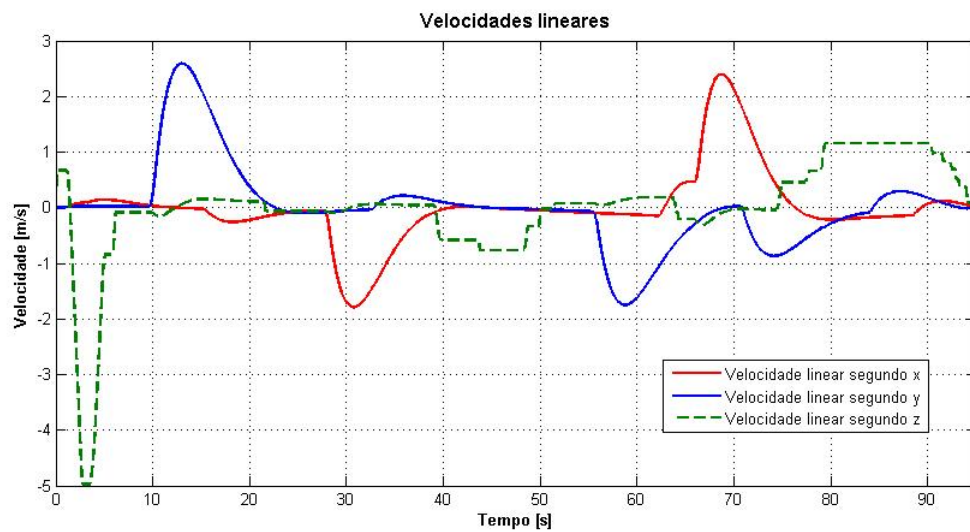


Figura 43: Velocidades lineares ao longo da segunda trajetória

É de notar ainda a diminuição da velocidade vertical à medida que o MALIV se aproxima do solo, conseguindo assim com este módulo efectuar uma descida mais ou menos controlada e com velocidades verticais convenientes. Todas as velocidades envolvidas durante a simulação, provaram ser de grandezas bastante aceitáveis para o tipo de UAV e de movimentos alcançados.

Este módulo apesar de possibilitar uma versatilidade ao MALIV de acordo com os sensores nele embarcados, revelou ser pouco funcional, dado ser necessária alguma prática para conseguir operar o MALIV em condições. Todos os valores retirados podem ser alterados consoante a rapidez de resposta que pretendemos, por isso, o critério de selecção para este tipo de movimentos, está relacionado com aquilo que se pensa ser uma situação considerada normal e ajustada ao tipo de funcionalidade que

um UAV deste tipo poderá vir a ter.

### 6.3.2 Módulo de visualização

O principal problema do módulo anterior é a necessidade do utilizador permanecer em constante alerta de modo a conseguir obter o controlo posicional do MALIV. De forma a corrigir essa situação foi criado o módulo de visualização que tem como base o modelo da câmara desenvolvida e apresentada ao longo da Secção 3.5.3. Este modelo consiste numa câmara embarcada no MALIV que através da visualização, de quatro pontos de referência consegue a estabilização numa posição NED. Desta forma, o MALIV passa a ser autónomo, não havendo interacção entre o utilizador e a missão a que o MALIV se destina. Essa missão passa pela escolha da altitude de observação, e depois por missões de vigilância associadas a objectos com velocidades moderadas.

O importante neste módulo é verificar o modelo da câmara desenvolvido, e saber se em conjunto com as saídas dos sensores já verificadas, o filtro de Kalman está em condições de recuperar a totalidade dos doze estados globais. Basicamente a ideia é substituir o sensor de posição ideal já verificado (Secção 6.2), pelo modelo da câmara desenvolvido. Dado que se optou pela utilização de pontos de referência, a visualização dos mesmos no referencial da imagem,  $R_{xyz}$ , corresponde a um par de coordenadas  $(x, y)$ , desprezando o eixo vertical. Isto equivale para os quatro pontos de referência, a oito saídas provenientes da câmara, quatro pares  $(x, y)$ , Figura 44.

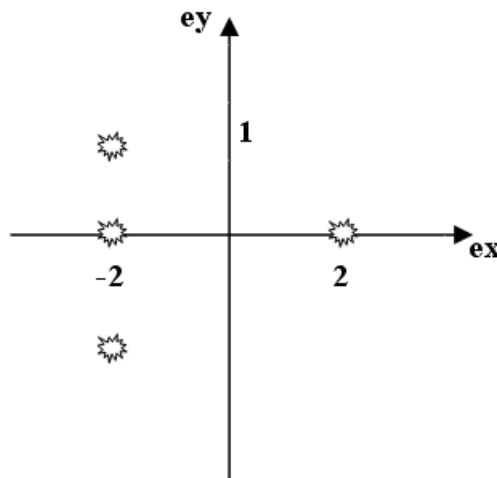


Figura 44: Pontos de referência no referencial NED,  $R_{NED}$

Sendo assim, mais uma vez é necessário criar as novas matrizes  $C$  e  $D$  provenientes do Jacobiano das equações dos sensores, enquanto que as matrizes  $A$  e  $B$  permanecem inalteradas pela dinâmica ser a mesma. É de realçar, que as primeiras e últimas três linhas, em conjunto com as primeiras e últimas três colunas, correspondem às matrizes  $C$  e  $D$  já calculadas no módulo com joystick, em que apenas era utilizado o acelerómetro e o magnetómetro<sup>10</sup>.

Considerando os quatro pontos de referência, a nova matriz  $C$   $[14 \times 6]$  obtida foi:

<sup>10</sup>Tanto a construção destas matrizes como de todos os outros parâmetros envolvidos durante esta simulação podem ser consultados ao longo do código desenvolvido e apresentado em anexo no Apêndice C.

$$C = \frac{\partial f_{\text{sensores}}}{\partial X} = \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & 0 & 0 & -19.62 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 19.62 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & -0.0001 & -0.0001 & 0 \\ 0 & \dots & 0 & 0 & -0.3577 & 0.0512 & 2.5547 & 0.1023 & 0.7153 \\ 0 & \dots & 0 & 0.3577 & 0 & 0.1028 & 0.1028 & 2.7093 & -0.3577 \\ 0 & \dots & 0 & 0 & -0.3577 & 0 & 2.5036 & 0 & -0.7153 \\ 0 & \dots & 0 & 0.3577 & 0 & -0.1018 & 0 & 2.7072 & 0 \\ 0 & \dots & 0 & 0 & -0.3577 & 0 & 2.5036 & 0 & 0.7153 \\ 0 & \dots & 0 & 0.3577 & 0 & 0.1028 & 0 & 2.7093 & 0 \\ 0 & \dots & 0 & 0 & -0.3577 & -0.0512 & 0 & -0.1023 & 0.7153 \\ 0 & \dots & 0 & 0.3577 & 0 & 0.1028 & 0 & 2.7093 & 0.3577 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \dots & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \quad (86)$$

A matriz  $D$  apresenta resultados diferentes em comparação com o caso do sensor de posição ideal, porque aí ainda não tinha sido adoptada a posição correcta do acelerómetro na estrutura do MALIV. No entanto o raciocínio é o mesmo. A matriz  $D$   $[14 \times 4]$  obtida foi:

$$D = \frac{\partial f_{\text{sensores}}}{\partial U} = \begin{bmatrix} 0 & 0 & 0.0052 & -0.0052 \\ -0.0043 & 0.0043 & 0 & 0 \\ -0.0105 & -0.0090 & -0.0097 & -0.0097 \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (87)$$

A partir daqui, tanto a matriz de ganho  $K$  como o filtro de Kalman em espaço de estados foram calculados com base nos comandos já explicitados:

$$\begin{cases} K = \text{lqrd}(A, B, Q, R, t_s) \\ \text{sys} = \text{ss}(A, [B \ G], C, [D \ H]); \end{cases}$$

Com<sup>11</sup>:

$$\begin{cases} Q_{ii} = [25 \ 25 \ 25 \ 250 \ 250 \ 250 \ 6.25/10 \ 6.25/10 \ 6.25/10 \ 418 \ 418 \ 418] & R_{ii} = [0.1 \ 0.1 \ 0.1 \ 0.1] \\ R_{K_{ii}} = [dd \ dd \ dd \ dd \ dd \ dd \ dd \ dd \ dd \ dd \ dd \ dd \ dd \ dd] & Q_{K_{ii}} = [dd \ dd \ dd \ dd \ dd \ dd] \\ G = \text{eye}(12, 6) & H = 0 \times \text{eye}(14, 6); \end{cases} \quad (88)$$

<sup>11</sup>Os valores máximos esperados para os estados envolvidos durante a simulação, para obtenção das matrizes  $Q$  e  $R$  são idênticos aos apresentados na Equação 77, à excepção de  $x_{max} = y_{max} = z_{max}$  agora considerados igual a  $1.26 \ m$



- **Apresentação dos resultados**

Mais uma vez estes resultados são obtidos com a simulação em tempo real e com o módulo integrado, fazendo uso desta vez do controlador construído para o caso do módulo de visualização ao contrário do que foi utilizado no módulo de joystick, Figura 39. O objectivo destes resultados passa por demonstrar as capacidades de vigilância do MALIV a um objecto animado com velocidades moderadas, a uma altitude definida inicialmente.

- Vigilância dum objecto ao longo dos eixos transversais  $xy$  após estabilização a uma dada altitude

Para que se possa testar este módulo é necessário começarmos a simulação numa posição em que o MALIV esteja em condições de observar os pontos de referência, assumindo os pontos de referência mencionados na Figura 44, é condição necessária que o MALIV esteja a uma dada altitude e perto da posição  $(x, y) = (0, 0)$ . A altitude escolhida para observação é definida no ficheiro *inicio.m* (Apêndice C). De modo a conseguirmos comparar os resultados obtidos com este módulo, com resultados anteriormente obtidos, procedeu-se à deslocação do MALIV numa trajectória semelhante. Isto é:

1. Estabilização à altitude definida no ficheiro *inicio.m*,  $7\text{ m}$
2. Deslocação transversal e positiva segundo o eixo  $y$
3. Estabilização na nova posição
4. Deslocação transversal e negativa segundo o eixo  $x$
5. Estabilização na nova posição
6. Deslocação transversal oposta, recuperação da posição inicial a  $7\text{ m}$  de altitude

Ao contrário do módulo com joystick, em que existe a liberdade por parte do utilizador em efectuar qualquer trajectória, neste módulo a altitude é fixa e definida no ficheiro inicial, e a posição transversal é controlada com a deslocação dos pontos de referência. Isto acontece porque o modelo foi construído de forma que o MALIV com base no filtro de Kalman procure anular o erro entre a posição dos pontos de referência e a sua posição actual. Desta forma, é possível verificar as capacidades de vigilância do MALIV a um objecto animado com velocidades transversais, desde que para tal, o mesmo consiga obter informação sobre elementos característicos desse objecto a ser vigiado, no presente caso, os pontos de referência.

Para efectuar esta simulação, o movimento dos pontos com coordenadas  $\begin{bmatrix} Q_x & Q_y \end{bmatrix}$  no referencial fixo foi associado a três botões do joystick, com três funções dependentes do tempo:

$$\begin{bmatrix} Q_x \\ Q_y \end{bmatrix} = \begin{bmatrix} Q_x \\ Q_y \end{bmatrix} + Tt \quad (89)$$

onde  $T$  corresponde a um vector associado ao botão pressionado do joystick. Assim, ao botão nº1 está associado uma deslocação negativa dos pontos de referência sobre o eixo dos  $x$ ,  $T = [-1\ 0]$ , ao botão nº2 uma deslocação positiva dos pontos de referência sobre o eixo dos  $y$ ,  $T = [0\ 1]$ , e ao botão número três uma deslocação do objecto segundo  $x$  e  $y$ ,  $T = [1\ -1]$ , Figura 45.

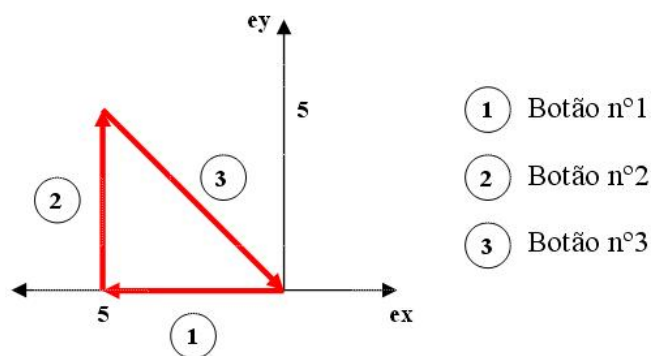


Figura 45: Trajectória do objecto ao longo da simulação no referencial NED,  $R_{NED}$

Isto quer dizer que para se efectuar a trajectória pretendida e consequentemente analisar a estabilidade do MALIV, é necessário simular o movimento de um objecto com essa trajectória através dos botões 1 a 3 do joystick, e verificar se o MALIV consegue acompanhá-lo.

Em termos da posição segundo os três eixos inerciais o resultado foi o seguinte, Figura 46.

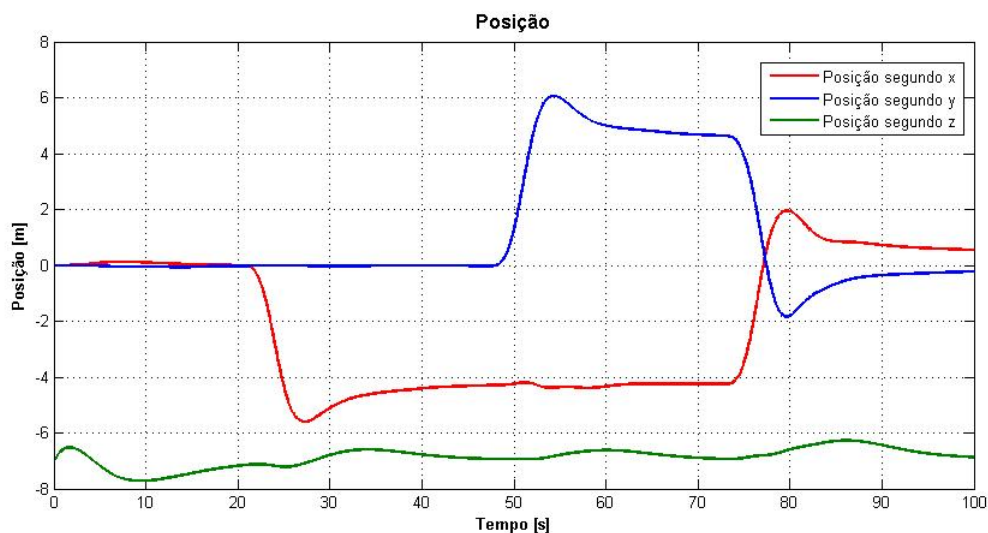


Figura 46: Posição de vigilância  $xyz$  ao longo da trajectória

Como pode ser observado, o MALIV consegue seguir o objecto animado com velocidade de  $1.5\text{ m/s}$ , e desta vez, ao contrário do que sucedia para o módulo com joystick, é possível o MALIV permanecer na mesma posição, estando esta totalmente estabilizada e não sendo necessária qualquer intervenção por parte do utilizador. A escolha da velocidade foi arbitrária, no entanto procurou-se ilustrar os resultados com uma velocidade em que o MALIV se comportasse dentro da normalidade. O MALIV consegue também acompanhar objectos com velocidades a rondar os  $2.5\text{ m/s}$ , mas desta forma, atinge valores de atitude e tensão aos motores bastante elevados.

A atitude é a esperada e está de acordo com a teoria até agora desenvolvida, isto é, após uma rotação inicial, existe uma rotação oposta para se obter a estabilização na posição escolhida. À picada está associada a deslocação segundo o eixo  $x$ , e ao rolamento a deslocação segundo  $y$ . No entanto, pode ser observado desta vez um acoplamento entre o rolamento e a guinada, Figura 47. Apesar disso, o MALIV consegue a obtenção da posição e orientação desejadas.

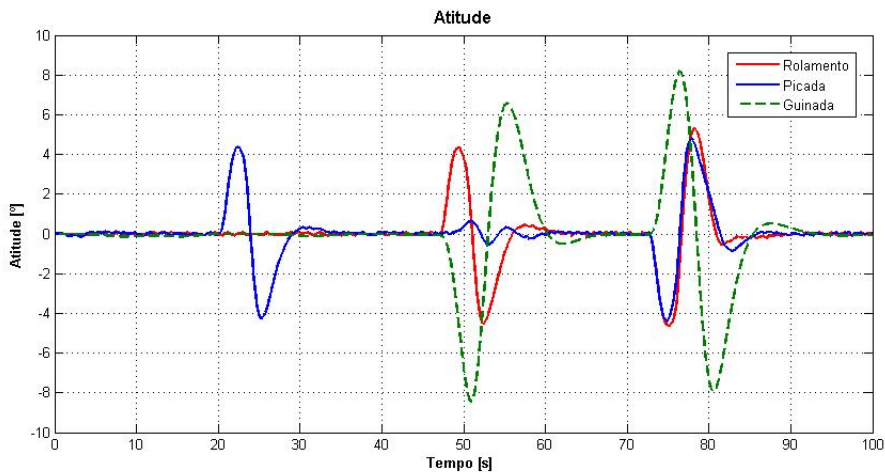


Figura 47: Atitude ao longo da trajetória de vigilância

Em termos de limites são atingidos desta vez mais ou menos 10 de amplitude mas só em termos da guinada, em rolamento e picada são atingidos apenas 5 de amplitude para a deslocação contínua.

Ao contrário das outras discussões, não vão ser apresentados os gráficos referentes aos motores e às velocidades para esta situação, uma vez essa teoria já ter sido amplamente discutida. Aqui, o importante era confirmar se o MALIV conseguiria manter uma altitude estabilizada e vigiar um objecto animado, algo que já foi comprovado.

- Vigilância de um objecto em constante movimento

Para além da simulação anterior, foi ainda elaborada uma simulação que se prendia pela vigilância de um objecto animado continuamente de uma velocidade de  $1.5\text{ m/s}$  ao longo de  $x$  e  $y$ . A diferença desta simulação consiste no prolongamento da trajetória ao longo dos eixos, de forma a poder observar como se comporta o MALIV em termos de atitude e velocidades. Esta simulação serve para ilustrar a base de muitas justificações adiantadas nas secções anteriores, quando se justificaram não só o andamento da tensão aos motores e a atitude do Quadrirotor quando este efectuava deslocamentos transversais. A trajetória foi a representada na Figura 48.

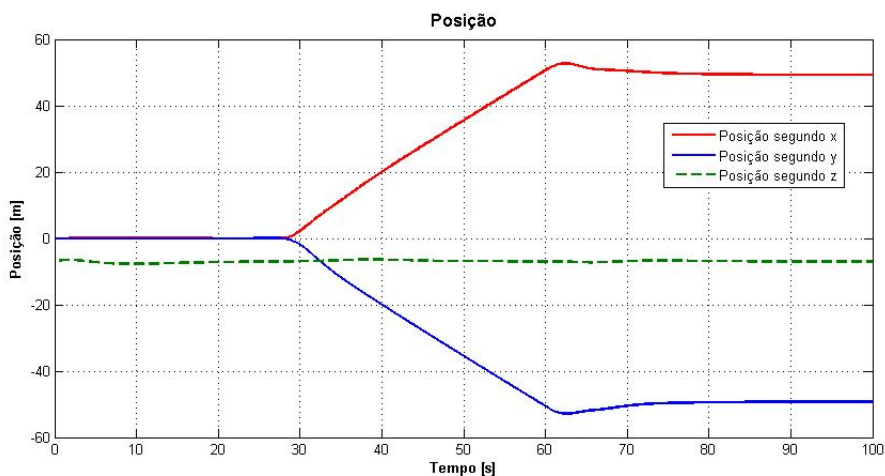


Figura 48: Posição de vigilância xyz ao longo da trajetória, segunda trajetória

Através da Figura 48 podemos observar um tipo de trajectória diferente das habituais, desta vez houve uma deslocação permanente do objecto durante mais ou menos 30 s segundo os dois eixos  $xy$ . A atitude equivalente a esta trajectória foi, Figura 49.

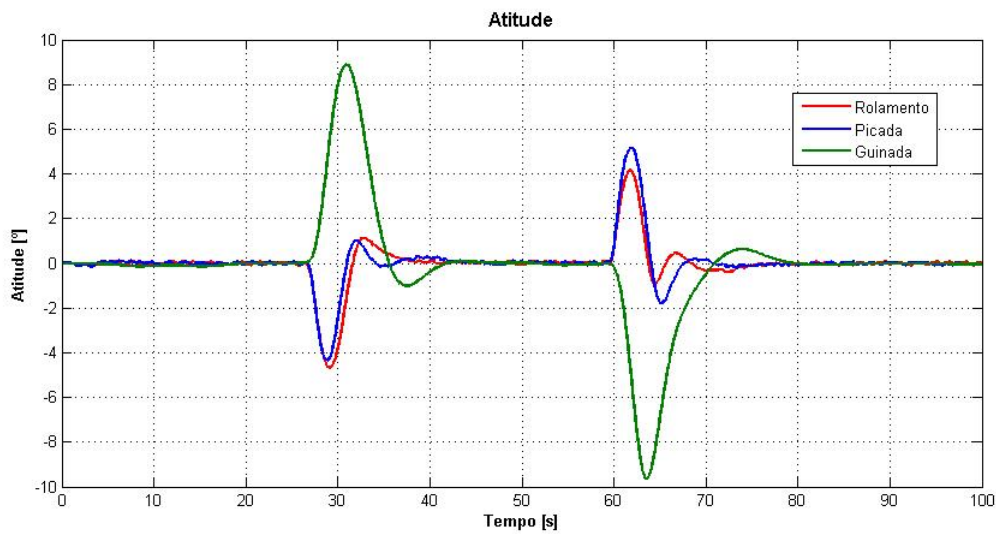


Figura 49: Atitude ao longo da segunda trajectória de vigilância

Para além do acoplamento já referenciado entre a guinada e o rolamento, na Figura 49 podemos ainda constatar que a anulação da rotação inicial é efectuada no momento da conclusão da deslocação sobre os eixos, isto faz com que até aí o MALIV esteja animado com uma determinada velocidade que o permite continuar a deslocar segundo os mesmos, Figura 50.

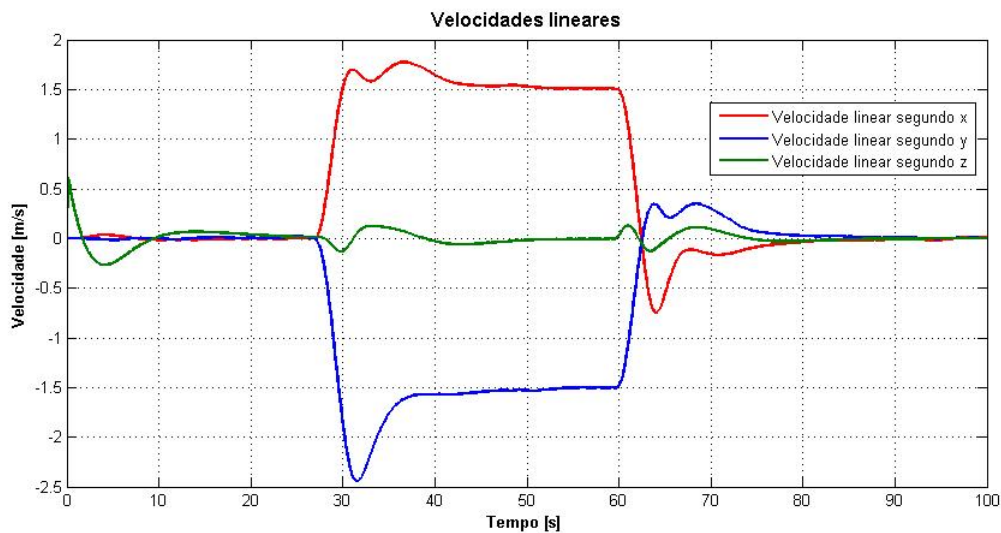


Figura 50: Velocidades lineares ao longo da segunda trajectória

Isto quer dizer que, quando o objecto a ser observado parou, o MALIV efectuou uma rotação contrária de modo anular a sua velocidade e com isso conseguir parar na posição devida.

Apesar das limitações inerentes à visualização dos pontos de referência numa situação inicial que fez com que cada uma destas simulações se inicie já a uma dada altitude, este módulo revela estar

bastante solidificado para os fins a que se propõe, estando assim reúnas as condições para a elaboração do módulo final: o módulo integrado.

### 6.3.3 Módulo integrado

Até agora têm vindo a ser apresentadas as discussões dos resultados de uma forma progressiva, obtidas ao mesmo tempo que diferentes objectivos foram ultrapassados e a proximidade com um ambiente real se tornava mais evidente. O módulo integrado corresponde ao módulo final adoptado para a simulação do MALIV, Figura 38. Este módulo é composto pelos módulos com joystick e módulo de visualização. Isto é, este módulo representa a totalidade de toda a formulação teórica desenvolvida até ao momento, desde a dinâmica, os sensores e os diferentes controladores adoptados. Foi por isso considerado essencial, fazer uma abordagem deste tipo para a secção das discussões. Este módulo funciona assim, como a conclusão de todos os outros resultados obtidos, em que tudo se enquadra para que este módulo corresponda de facto a uma situação possível de ambiente real.

Tal como foi debatido ao longo da Secção 5.1 em que se dá ênfase à teoria por de trás de cada um dos módulos principais, os dois módulos anteriores têm fragilidades que provocava a inaptidão dos mesmos para o controlo versátil e automatizado do MALIV em ambiente real, objectivo da presente tese. Se por um lado ao controlarmos o MALIV através do Joystick, temos uma maior liberdade em termos posicionais caracterizada pela não automatização do mesmo e à necessidade constante de intervenção por parte do utilizador, por outro, utilizando a câmara modelada para controlo posicional, estamos limitados à aptidão da mesma em observar os pontos de referência, ou seja, as simulações já se iniciam por cima dos mesmos.

Como foi explicado na Secção 6.3.3, a ideia por de trás deste módulo consiste na condução do MALIV através do módulo com joystick até uma posição em que o mesmo consiga observar os pontos de referência, e partir daí, use o controlador desenvolvido no módulo de visualização para se estabilizar e vigiar o objecto se o mesmo for animado de velocidades transversais moderadas. Essa transição é efectuada usando o “*throttle*” do joystick para permuta entre os módulos.

Como o facto de qualquer um dos controladores já ter sido apresentado e discutido nas secções anteriores, nesta discussão final serão já apresentados os resultados para os diferentes tipos de missões consideradas.

- **Apresentação dos resultados**

Uma vez se tratar do módulo final, e dado que o objectivo consiste na elaboração de situações plausíveis de serem consideradas ajustadas a um ambiente real, ao contrário das outras simulações, nesta última secção são consideradas três tipos de missões distintas que procuram exemplificar as capacidades do Quadrirotor desenvolvido em realizar objectivos destinados a um UAV deste tipo.

- **Missão nº1:** A primeira missão serve para verificar se a permuta entre os módulos funciona. É elevado o MALIV através do módulo com joystick, e após o mesmo se situar numa altitude que consiga visualizar os pontos de referência, procede-se à permuta dos controladores.

Para esta primeira missão é seleccionada a altura desejada de observação final no ficheiro inicial, *inicio.m*, e depois de correr o ficheiro é lançada a simulação. A Figura 51 ilustra a posição inercial

do MALIV ao longo da presente missão, e a Figura 52 o andamento do “*throttle*” do joystick para se visualizar o momento da transição entre os módulos.

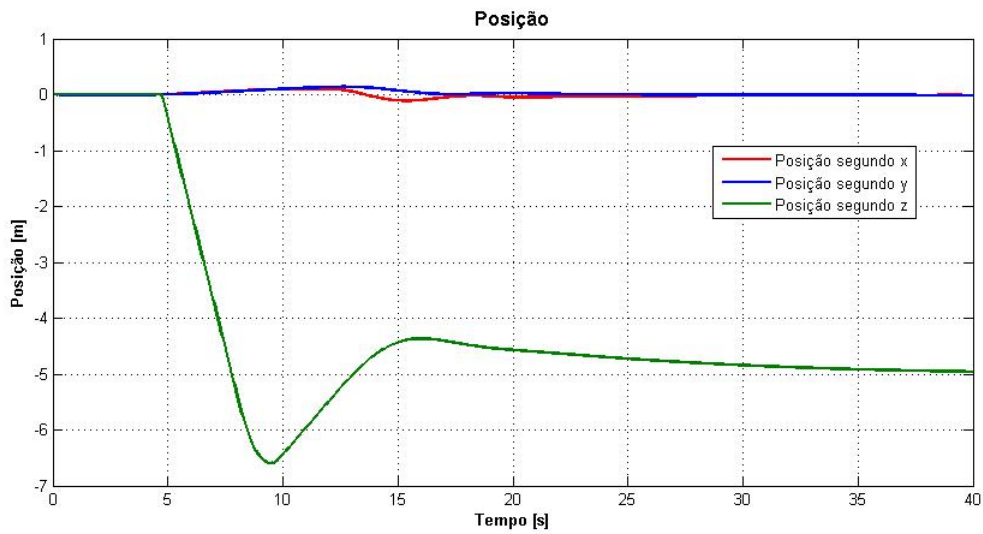


Figura 51: Posição do MALIV ao longo do tempo, Missão nº1

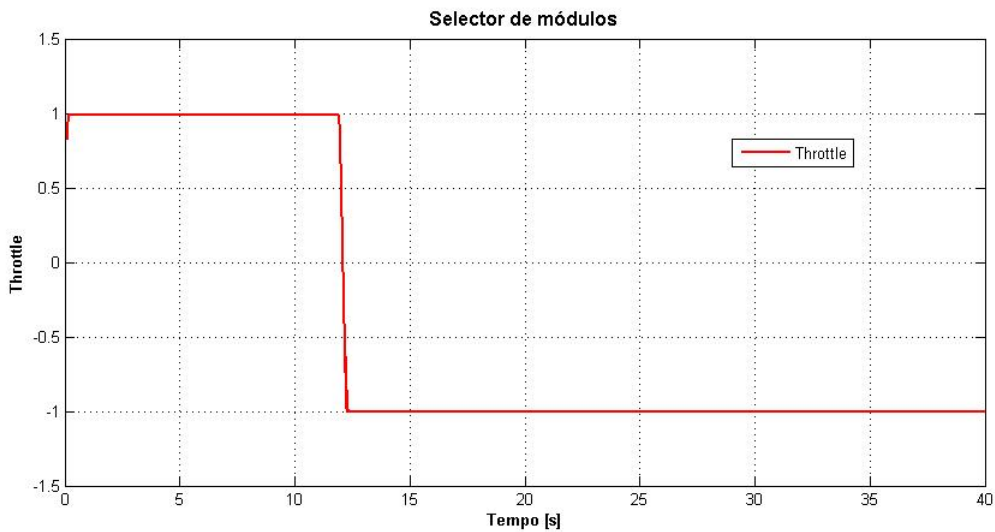


Figura 52: Selector dos módulos ao longo da primeira missão

Na Figura 52, é visível à passagem do segundo 12, a alteração do valor do “*throttle*” (por acionamento manual) para valores inferiores a zero, fazendo com que no selector ilustrado na Figura 23 o mesmo cause a realimentação dos motores através do controlador construído para o módulo de visualização. A partir deste momento o MALIV desloca-se no sentido de se estabilizar à altitude definida inicialmente (5 m). Como pode ser observado através da Figura 51, desta vez, e como foi justificado no sensor de posição ideal (Secção 6.2) já não há a oscilação tão evidente da posição vertical, dado que agora o MALIV foi conduzido até perto dela através do joystick. Desta maneira, o “salto” entre a posição actual e a de referência é muito menor, e por isso, muito menor é o erro entre as mesmas fazendo com que o MALIV não se afaste consideravelmente da posição final desejada antes de se estabilizar. O objectivo desta missão foi cumprido, e conseguiu-se verificar a validade do

simulador construído tendo em conta a permuta entre os dois controladores.

- **Missão nº2:** Esta missão consiste na verificação de uma situação em ambiente real em que há o objectivo de vigiar um objecto situado inicialmente numa posição afastada.

Ao contrário da missão anterior em que apenas foi demonstrada a aptidão do módulo integrado em realizar a permuta dos dois módulos discutidos nas secções anteriores, nesta missão os pontos de referência que pretendem simular um objecto, estão distantes do ponto de partida do MALIV cerca de  $5\text{ m}$  de distância segundo o eixo  $y$ . Para isso é necessário adicionar  $5\text{ m}$  à coordenada  $y$  dos pontos representados na Figura 44.

A Figura 53 e a Figura 54, representam o andamento da posição do MALIV no referencial  $R_{NED}$  ao longo da simulação, e o “*throttle*” do joystick para a selecção do controlador envolvido, ao longo do tempo.

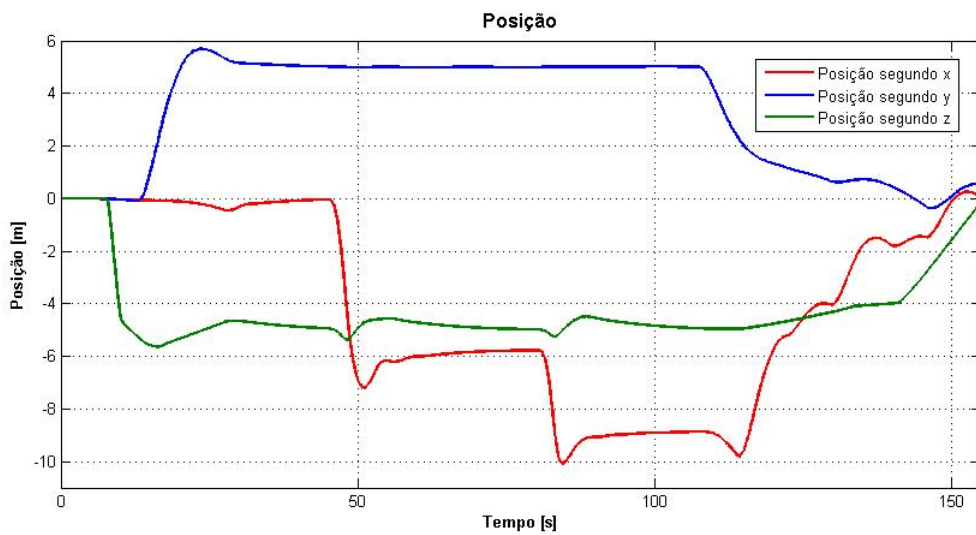


Figura 53: Posição do MALIV ao longo do tempo, Missão nº2

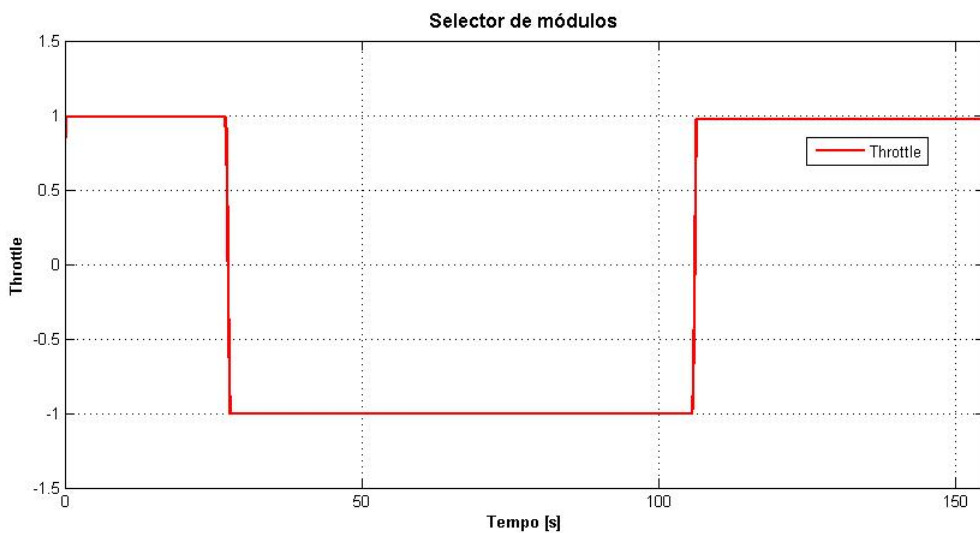


Figura 54: Selector dos módulos ao longo da segunda missão

Como pode ser observado nas figuras anteriores, no momento da primeira transição, aproximadamente 25 s, o MALIV encontra-se na posição  $(x, y, z) = (0, 6, -6)$  pelo que se seguirá uma correcção da mesma, para que ao fim de se conseguir estabilizar, o mesmo possua a posição final pretendida, isto é  $(x, y) = (0, 5)$  com 5 m de altitude. A Figura 55 ilustra o erro associado entre as saídas estimas e as saídas reais no momento da permuta entre os módulos para o caso de cada uma das coordenadas  $y$  dos pontos de referência.

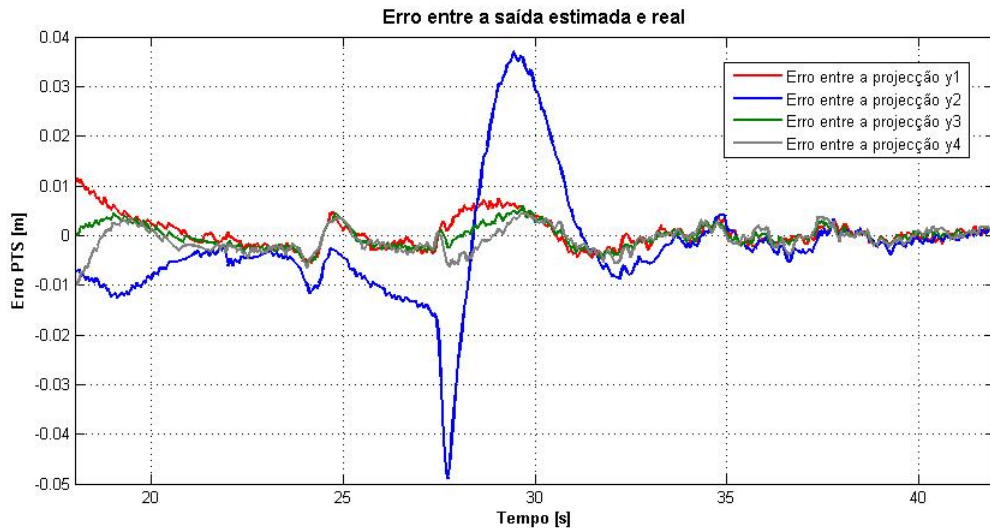


Figura 55: Erro associado entre as saídas estimas e as saídas reais para a coordenada  $y$  dos pontos de referência

O pico mais elevado corresponde ao ponto 2, e está de acordo com o esperado, uma vez esse ponto corresponder ao vértice isolado do triângulo formado pelo conjunto dos quatro pontos de referência, Figura 44. A Figura 56 apresenta as velocidades lineares envolvidas, e obviamente, que também nessa zona, a velocidade segundo  $y$ ,  $V$ , é a mais considerável.

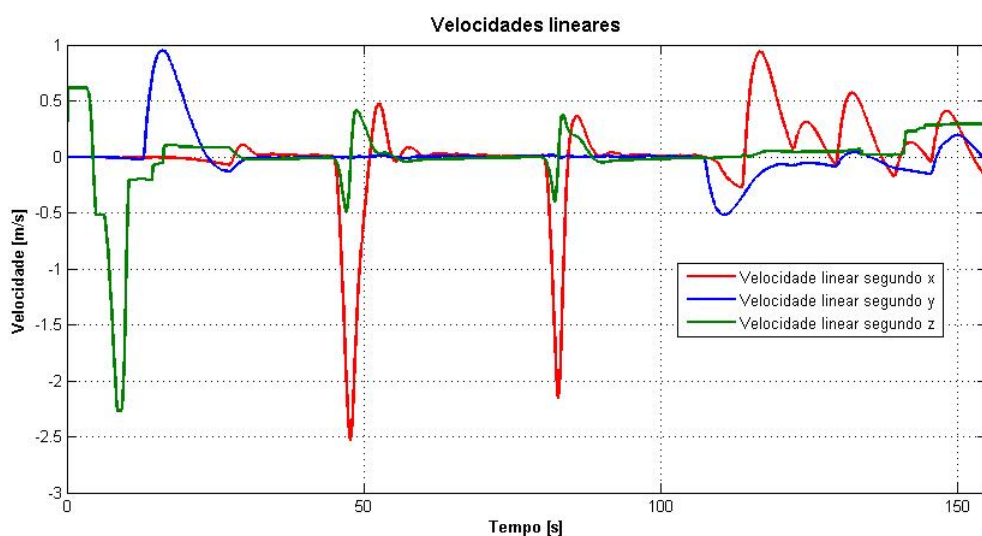


Figura 56: Velocidades lineares do MALIV ao longo do tempo, Missão nº2

A segunda parte da simulação consistiu na atribuição do movimento ao objecto (entre os 45 s e os



80 s aproximadamente), algo já debatido na secção do módulo de visualização, Secção 6.3.2. Como tal, apenas se deixa como referência, o andamento ajustado das velocidades, Figura 56, e a atitude envolvida, Figura 57, mais uma vez em consonância com as duas deslocações segundo  $x$  efectuadas, Figura 53.

A missão termina com uma nova permuta entre os módulos, perto dos 105 s, Figura 54, de modo a que se consiga aterrar o MALIV de uma forma suave na posição de partida, longe do local em que se situaria o objecto a ser vigiado, Figura 53.

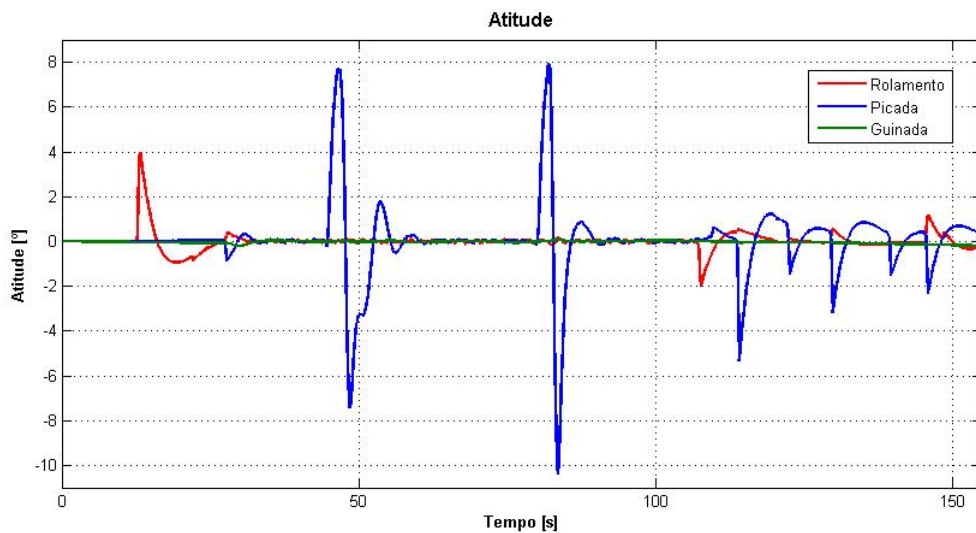


Figura 57: Atitude do MALIV ao longo do tempo, Missão nº2

O sucesso desta missão levou a que fossem testados alguns dos limites do MALIV, não só na primeira abordagem em relação ao objecto, mas também no tipo de trajectórias que o mesmo consegue realizar.

- **Missão nº3:** Esta última missão consiste na tentativa de demonstrar as capacidades do MALIV na vigilância de trajectórias mais complexas, bem com na perseguição de um objecto inicialmente animado com velocidades moderadas.

Nesta última missão as capacidades do MALIV são levadas ao extremo, e o objectivo é que após conduzi-lo de encontro a um objecto que já esteja a efectuar uma trajectória, o MALIV consiga seguir o objecto.

Para que tal simulação seja possível, desta vez é atribuído ao botão nº1 do joystick, uma função trigonométrica que simule uma trajectória circular. Essa trajectória circular é adoptada pelos pontos de referência enquanto se estiver a pressionar o botão nº1, quando o botão pressionado for o nº3, é atribuído um movimento linear segundo  $x$  à semelhança do que aconteceu na missão nº2. Da forma como a equação está construída, quando se iniciar a trajectória circular, o objecto estará sobre a posição  $(x, y) = (0, 10)$  e efectuará uma translação contrária ao sentido dos ponteiros do relógio, Figura 58.

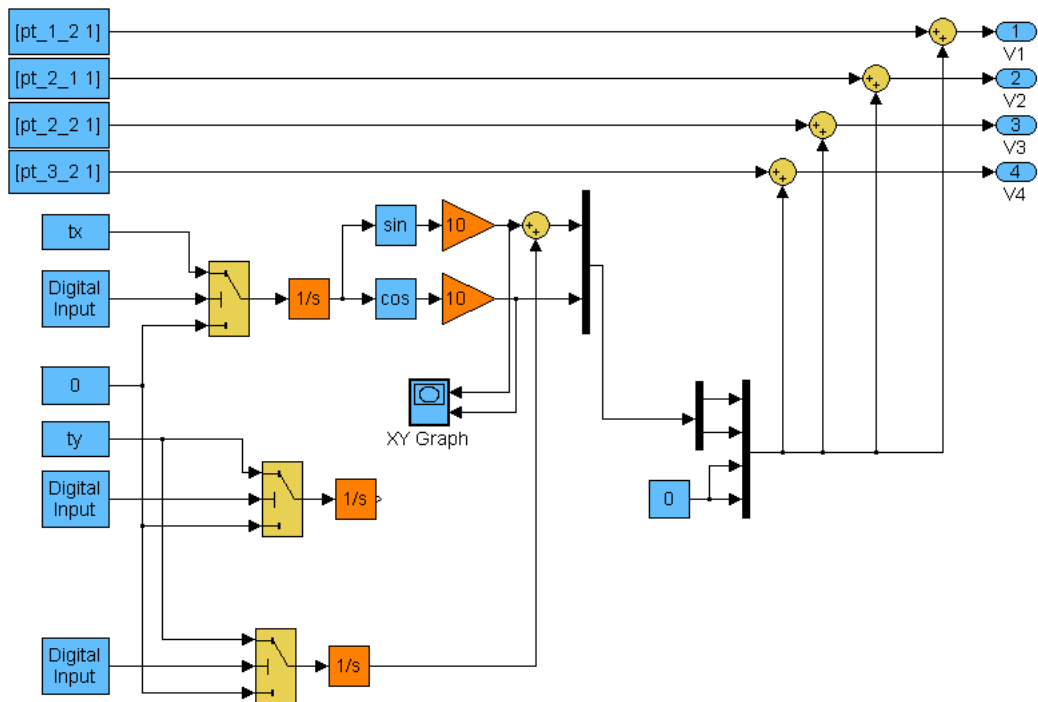


Figura 58: Alteração dos pontos de referência em diagrama de blocos

A ideia é elevar o MALIV até uma dada altura, próxima da altitude escolhida inicialmente, e fazê-lo posteriormente interceptar a trajetória do objecto, para que a partir daí, com a permuta dos controladores, averiguar as capacidades do MALIV em efectuar a mesma trajetória circular de 10 m de diâmetro, com base no sistema de visualização.

A missão consistiu em:

1. Elevação do MALIV para perto da altitude de vigilância
2. Condução do MALIV até um ponto de passagem da trajetória do objecto
3. Permuta dos controladores
4. Perseguição do objecto segundo uma trajetória circular de forma autónoma
5. Perseguição do objecto após paragem, segundo uma trajetória rectilínea e de forma autónoma
6. Perseguição do objecto segundo uma nova trajetória circular e de forma autónoma
7. Perseguição do objecto após paragem, segundo uma nova trajetória rectilínea e de forma autónoma
8. Permuta dos controladores
9. Aterragem suave por parte do MALIV

A Figura 59 representa a trajectória efectuada pelo objecto em termos de  $xy$  no referencial  $R_{NED}$ .

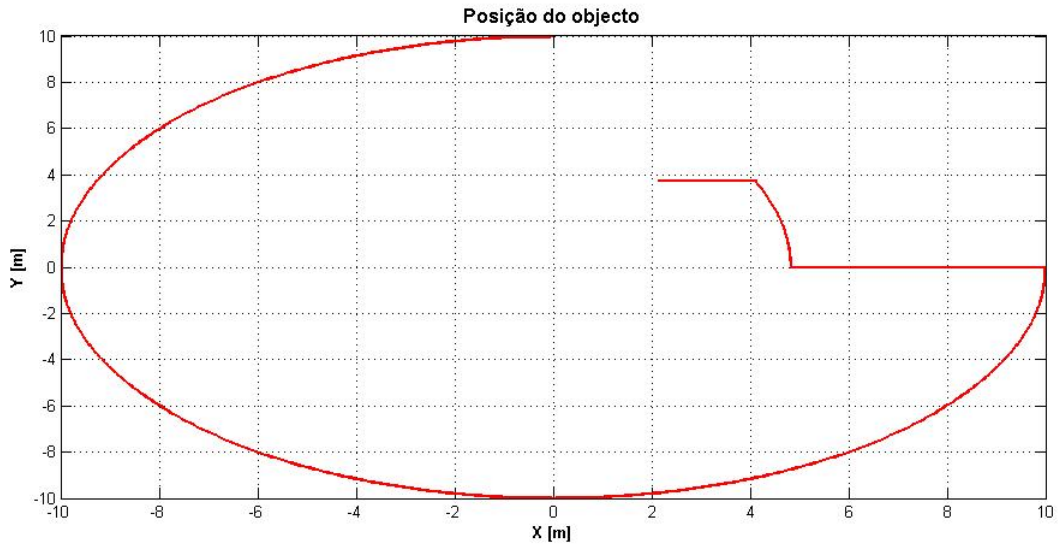


Figura 59: Trajectória do objecto a ser vigiado segundo  $xy$

A Figura 60 representa a posição  $xy$  no referencial  $R_{NED}$  ao longo do tempo por parte do objecto.

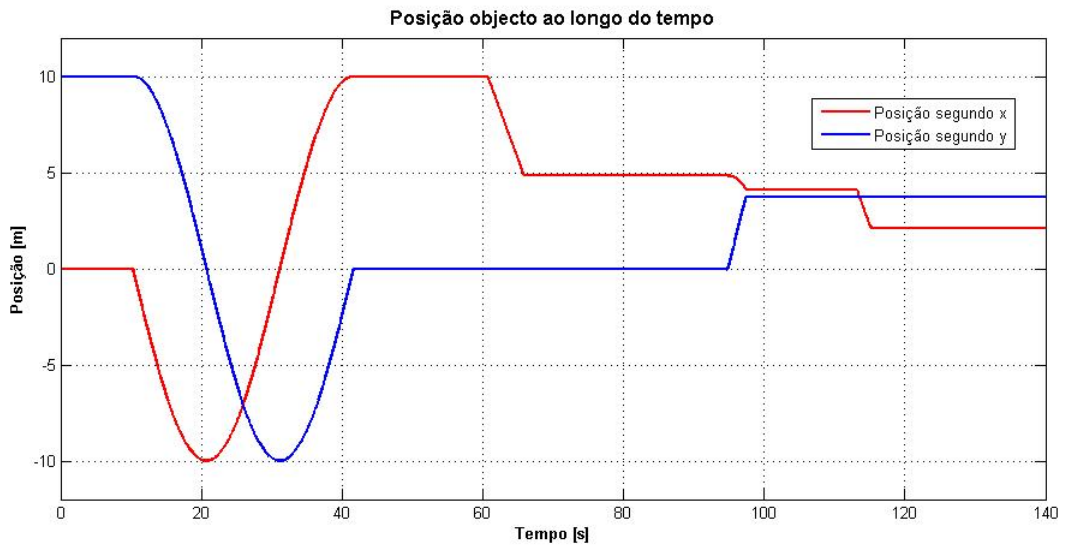


Figura 60: Trajectória do objecto a ser vigiado segundo  $xy$  ao longo do tempo

A ideia passa por conduzir o MALIV até perto da posição inicial do objecto, e depois atribuir o movimento ao objecto através do botão nº1 do joystick. Por fim realiza-se a transição entre os módulos com o “*throttle*” para ver se o MALIV consegue acompanhar o objecto, Figura 61.

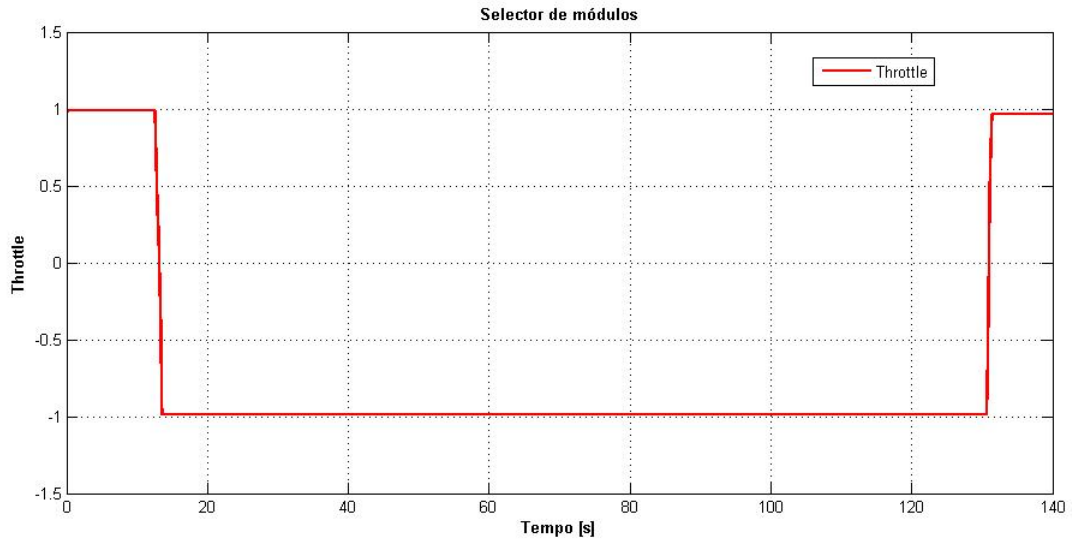


Figura 61: Selector dos módulos ao longo da terceira missão

Por comparação entre a Figura 60 e a Figura 61, é possível verificar que a transição entre os módulos, que ocorre perto dos 12 s, é posterior à atribuição do movimento do objecto que acontece perto dos 9 s .

Na altura em que decorre a transição dos módulos, perto dos 12 s, a posição do objecto é aproximadamente de  $(x, y) = (-2, 7)$ , Figura 60, e a posição do MALIV nessa altura é de aproximadamente  $(x, y) = (1, 6)$ , Figura 62.

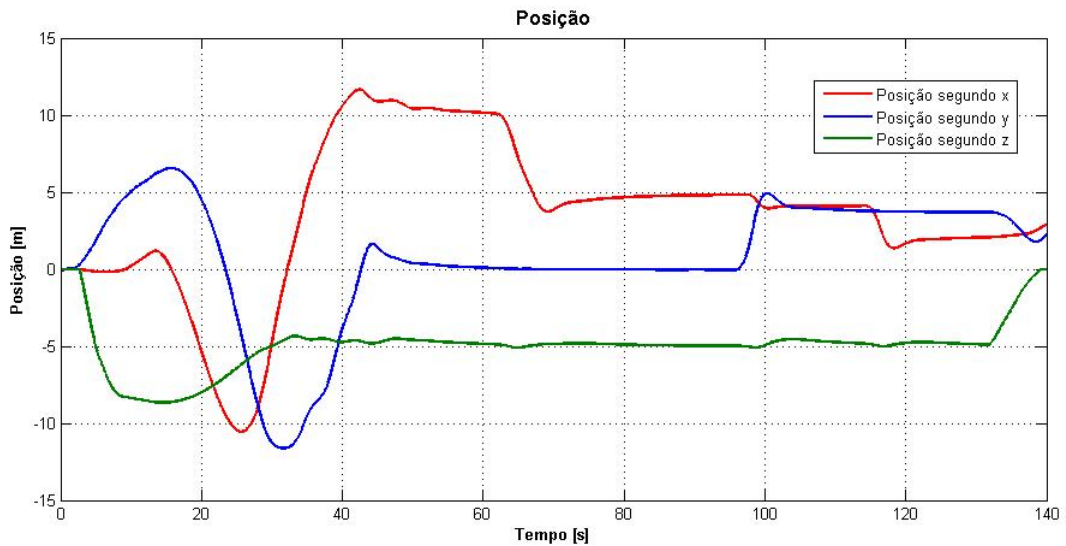


Figura 62: Posição do MALIV ao longo do tempo, Missão nº3

O que se verifica é que apesar de haver um erro inicial máximo em termos de posição  $x$  na ordem dos 3 m, o MALIV consegue recuperar praticamente a totalidade do erro e percorrer a mesma trajectória que o objecto a ser vigiado, ao ponto de no final da primeira trajectória circular, por volta dos 40 s, apresentar praticamente a mesma posição que o objecto, isto é,  $(x, y) = (10, -2)$ . Quanto mais tempo passar ou quanto menor seja a velocidade do objecto, menor é o erro.

Posteriormente à primeira trajectória, há uma zona de trajectória rectilínea seguida de uma trajectória circular, Figura 59. Nestas condições o MALIV comporta-se como o esperado e consegue acompanhar o objecto. Na fase final da primeira trajectória rectilínea (aproximadamente 65 s), o objecto está na posição  $(x, y) = (5, 0)$ , Figura 60, da mesma forma que também o MALIV se situa na mesma posição, Figura 62. Isto revela o melhor comportamento do MALIV em trajectórias rectilíneas. No final da segunda trajectória circular, mais uma vez há o acompanhamento total por parte do MALIV, sendo que à passagem do segundo 100, tanto o MALIV como o objecto se encontram bastante próximos do ponto de coordenadas  $(x, y) = (4, 4)$ . A trajectória do objecto termina no ponto de coordenadas  $(x, y) = (2, 4)$ , Figura 59, e foi a partir dos 130 s que se efectuou nova mudança nos módulos, Figura 61, para realizar a aterragem manual através do joystick, Figura 62.

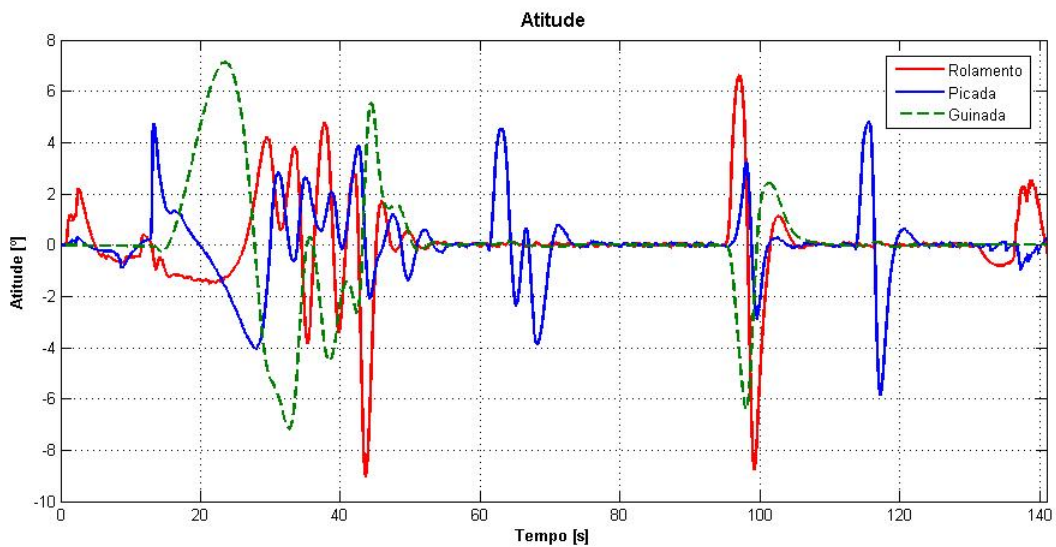


Figura 63: Atitude do MALIV ao longo do tempo, Missão nº3

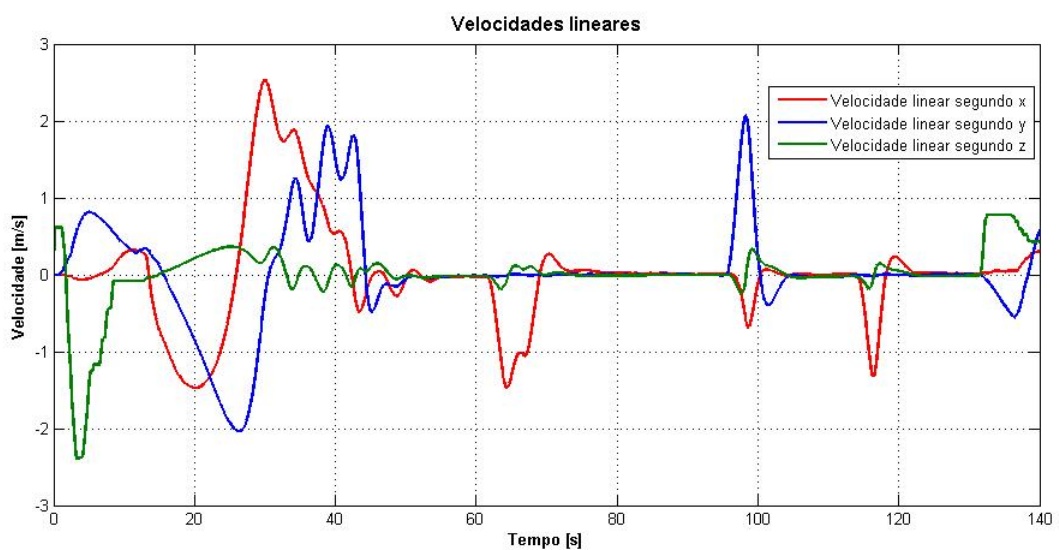


Figura 64: Velocidades lineares do MALIV ao longo do tempo, Missão nº3

Em relação à atitude existe um pico em termos de picada perto da primeira permuta dos módulos, 15 *s*, dado que aí como já foi mencionado, o erro entre a posição  $x$  era o maior e rondava os 3 *m*. Segue-se uma zona de rotações mais ou menos equilibradas correspondentes à trajectória circular, em que se ronda os 4, e rotações bem menos acentuadas quando o MALIV já se encontra numa posição estabilizada, 50 *s*, Figura 63 .

Em termos de velocidades (Figura 64) o resultado está ajustado aos movimentos efectuados pelo MALIV. Para as trajectórias circulares estamos na presença de velocidades segundo  $x$  e  $y$ , enquanto que na mudança da posição transversal  $x$ , só a velocidade  $U$  revela ser considerável, Figura 64)

Estas simulações servem não só para testar as diferentes possibilidades em termos de missões exigidas a um UAV como o MALIV, mas também, para demonstrar a robustez que o tipo de controlador e filosofia adoptadas possuem. Sendo que, houve a preocupação de acabar as presentes discussões com situações que se poderiam reflectir numa situação real, não deixando descorar aquilo que foi desde sempre o objectivo principal desta tese, conseguir controlar e simular um Quadrirotor convencional, aproximando-o mais possível de uma situação real em que o mesmo possa vir a ser substituído verdadeiramente pelo ALIV.

## 7 Conclusões

A presente tese teve como objectivo principal, estudar um UAV de asas rotativas composto por quatro rotores capazes de por si só controlar os seis graus de liberdade que caracterizam o movimento segundo os três eixos inerciais, o Quadrirotor. Foi proposto que, a partir de dados facultados de um Quadrirotor existente, o ALIV, se conseguisse simular situações em ambiente real para as quais o princípio de funcionamento do mesmo se justificasse.

O primeiro objectivo a ser cumprido, foi a modelação de uma dinâmica capaz de reproduzir os efeitos esperados pela variação da rotação de cada um dos quatro rotores, base fundamental na caracterização de um Quadrirotor. Com isto, através de dados estruturais do ALIV, conseguiu-se validar as capacidades do mesmo, em obter deslocações transversais, funcionando como se de um Quadrirotor convencional se tratasse.

Partindo do equipamento a bordo do ALIV, foram modelados os sensores responsáveis pelo controlo da atitude do mesmo. De modo a tornar o ALIV mais funcional e versátil, houve a necessidade de modelar um controlo posicional. Desta forma, foram criadas duas soluções, uma através do joystick, e outra através de uma câmara embarcada. Nos dois módulos construídos verificaram-se limitações prejudiciais ao funcionamento do ALIV em ambiente real. Se por um lado no primeiro havia a necessidade da presença constante do utilizador e uma grande sensibilidade em termos de estabilidade posicional, no segundo havia a necessidade de o mesmo partir numa posição em que já observasse os pontos de referência. Com base nisto, foi construído um módulo integrado capaz de utilizar o módulo com joystick numa fase inicial até à visualização dos pontos de referência, e o módulo de visualização, para a vigilância de um objecto animado com velocidades moderadas. Desta forma, foi possível apresentar simulações plausíveis de caracterizar missões a que um UAV deste género possa estar sujeito.

Tendo em conta que um dos objectivos proposto pela presente tese era aproximar o mais possível o sistema desenvolvido com a realidade para futura implementação no ALIV, os modelos foram adaptados para uma simulação em tempo real. Foi ainda criado um módulo de interface com o simulador FLIGHTGEAR, onde mais facilmente é possível observar a movimentação do ALIV ao longo do tempo. Com isto, apesar de os dados não puderem ser transpostos para o papel, revelaram ser uma mais valia na observação 3D da atitude do mesmo, ao invés da visualização temporal de cada um dos estados envolvidos na simulação.

Apesar de terem sido realizadas algumas simplificações com o intuito de conseguir construir um sistema mais adequado às necessidades, a solução final demonstrou ser bastante satisfatória, dado que vários objectivos foram sendo ultrapassados com a finalidade de aproximar cada vez mais a simulação de um ambiente real. Com isto, não só se conseguiram apresentar diferentes módulos de simulação, como no fim se conseguiu integrá-los de forma a tornar a simulação do MALIV mais versátil, apresentando alguns exemplos do comportamento do mesmo, face a diversas missões propostas.

### 7.1 Trabalhos futuros

Como tem vindo a ser referido, a presente tese consiste na implementação de um sistema capaz de no futuro ser implementado no ALIV de modo a que se possa por em prática todos os princípios exemplificados virtualmente ao longo desta tese.

Desta forma, em termos de continuidade do trabalho desenvolvido, estão um conjunto de processos que poderão aproximar ainda mais estas simulações do ambiente real exigido pelo ALIV. Entre as quais, destacam-se processos correctivos e de melhorias à modelação desenvolvida. Para isso poderá proceder-se à adopção de outros componentes do ALIV como sejam o caso do sistema motor mais hélice, modelados através de ensaios experimentais sobre os mesmos para que a modelação fosse mais exacta, e ainda a implementação dos graus de liberdade suplementares do ALIV, ganhando com isso novas possibilidades de simulação, tanto de estabilização como de deslocações transversais. E por fim, a ligação entre o sistema desenvolvido e o modelo real através da troca de informação entre os equipamentos embarcados e uma estação de trabalho. Com a construção de uma plataforma de testes, a integridade da estrutura do ALIV pode ser facilmente salvaguardada, começando depois a realização de pequenos testes entre a estação de trabalho e o ALIV que validem as simulações até então desenvolvidas.



## Referências

- [1] Office of the Secretary of Defence Unmanned Aircraft Systems Roadmap: 2005-2030, 2008
- [2] [http://en.wikipedia.org/wiki/Unmanned\\_aerial\\_vehicle](http://en.wikipedia.org/wiki/Unmanned_aerial_vehicle), Fevereiro 2008
- [3] <http://en.wikipedia.org/wiki/Quadrotor>, Fevereiro 2008
- [4] <http://forum.heli4.com/index.php?showtopic=4856>, Fevereiro 2008
- [5] J.Gordon Leishman. The Bréguet-Richet Quadrotor Helicopter of 1907, 2002
- [6] Tarek Hamel, Robert Mahony, Rogelio Lozano, & James Ostrowski. Dynamic Modelling and Configuration Stabilization for an X4-Flyer. Em *IFAC 15th Triennial World Congress, Barcelona, Espanha*, 2002
- [7] Tarek Hamel, Robert Mahony, & Abdelhamid Chriette. Visual servo trajectory tracking for a four rotor VTOL aerial vehicle. Em *Proceedings of the IEEE International Conference on Robotics and Automation*, 2002
- [8] N. Guenard, T. Hamel, V. Moreau. Dynamic modeling and intuitive control strategy for an “X4-flyer”. Em *ICCA, Budapeste, Hungria*, 2005
- [9] <http://www.rctoys.com/draganflyer5.php>, Março 2008
- [10] P. Pounds, R. Mahony, P. Hynes, & J. Roberts. Design of a Four-Rotor Aerial Robot. Em *Proceedings of the 2002 Australasian Conference on Robotics and Automation*, 2002
- [11] Paul Pounds, Robert Mahony, Joel Gresham, Peter Corke, & Jonathan Roberts. Towards Dynamically-Favourable Quad-Rotor Aerial Robots. Em *Proceedings of the 2004 Australasian Conference on Robotics and Automation*, 2004
- [12] Gabe Hoffmann, Dev Gorur Rajnarayan, Steven L. Waslander, David Dostal, Jung Soon Jang, & Claire J. Tomlin. The Stanford Testbed of Autonomous Rotorcraft for Multi Agent Control (STARMAC). Em *Proceedings of the 23rd Digital Avionics Systems Conference*, Stanford, 2004
- [13] E. Altug, J. P. Ostrowski, & R. Mahony. Control of a Quadrotor Helicopter Using Visual Feedback. Em *Proceedings of the IEEE International Conference on Robotics and Automation*, 2002
- [14] E. Altug, J. P. Ostrowski, & C. J. Taylor. Quadrotor Control Using Dual Camera Visual Feedback. Em *Proceedings of the IEEE International Conference on Robotics and Automation*, 2003
- [15] Samir Bouabdallah, Andre Noth, & Roland Siegwan. PID vs. LQ Control Techniques Applied to an Indoor Micro Quadrotor. Em *Proceedings of 2004 IEEE/RSJ International Conference On Intelligent Robots and Systems*, 2004
- [16] Brian L. Stevens, Frank L. Lewis. Aircraft Control and Simulation, 2nd Edition, New York: John Wiley & Sons, Inc., 2003
- [17] J. R. Azinheira. Controlo de Voo. Folhas da cadeira de Controlo de Voo, Instituto Superior Técnico, Lisboa 2007

- [18] Barnes W. McCormick . Aerodynamics, Aeronautics, and Flight Mechanics, 2nd Edition, New York: John Wiley & Sons, Inc., 1995
- [19] <http://www.FLIGHTGEAR.org>, Julho 2008
- [20] <http://www.mathworks.com/products/aerotb/>, Julho 2008
- [21] <http://www.SOLIDWORKS.com>, Julho 2008
- [22] <http://www.BLENDER.org>, Julho 2008

## Apêndice A

Nesta secção serão apresentados com detalhe todos os passos envolvidos na obtenção das equações que definem a dinâmica do problema.

**Dedução da matriz de rotação global:**

$$S_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad S_\theta = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad S_\psi = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S = S_\psi S_\theta S_\phi$$

$$S_E = \begin{bmatrix} \cos \psi \cos \theta & \sin \psi \cos \theta & -\sin \theta \\ \cos \psi \sin \theta \sin \phi - \cos \phi \sin \psi & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \cos \theta \sin \phi \\ \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi & \sin \psi \sin \theta \cos \phi - \sin \phi \cos \psi & \cos \theta \cos \phi \end{bmatrix}$$

**Dinâmica**

$$F_T = \frac{d}{dt} [mV_T]_B + \omega \times [mV_T]_B$$

$$M_T = \frac{d}{dt} [I\omega]_B + \omega \times [I\omega]_B$$

Dado que a massa é constante,  $\omega = [P \ Q \ R]$  e  $V_T = [U \ V \ W]$  então:

$$\begin{bmatrix} P & Q & R \end{bmatrix}^T \times \begin{bmatrix} U & V & W \end{bmatrix}^T = \begin{bmatrix} QW - RV \\ RU - PW \\ PV - QU \end{bmatrix}$$

$$F_T = m \begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} + m \begin{bmatrix} QW - RV \\ RU - PW \\ PV - QU \end{bmatrix}$$

$$S_E \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T = \begin{bmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{bmatrix}_B g$$

$$\begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} F_{px} \\ F_{py} \\ F_{pz} \end{bmatrix} + g \begin{bmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{bmatrix} - \begin{bmatrix} QW - RV \\ RU - PW \\ PV - QU \end{bmatrix}$$

$$\dot{U} = \frac{F_{px}}{m} - g \sin \theta - QW + RV$$

$$\dot{V} = \frac{F_{py}}{m} + g \sin \phi \cos \theta - RU + PW$$

$$\dot{W} = \frac{F_{pz}}{m} + g \cos \theta \cos \phi - PV + QU$$

Para a equação dos momentos,  $I$  é uma matriz principal,  $I = \begin{bmatrix} I_{11} & 0 & 0 \\ 0 & I_{22} & 0 \\ 0 & 0 & I_{33} \end{bmatrix}$

$$\omega \times [I\omega] = \begin{bmatrix} (I_{33} - I_{22}) RQ & (I_{11} - I_{33}) QP & (I_{22} - I_{11}) PQ \end{bmatrix}$$

$$M_T = \begin{bmatrix} I_{11} \dot{P} \\ I_{22} \dot{Q} \\ I_{33} \dot{R} \end{bmatrix} + \begin{bmatrix} (I_{33} - I_{22}) RQ \\ (I_{11} - I_{33}) QP \\ (I_{22} - I_{11}) PQ \end{bmatrix}$$

$$\dot{P} = \frac{M_x}{I_{11}} - \frac{(I_{33} - I_{22}) RQ}{I_{11}}$$

$$\dot{Q} = \frac{M_y}{I_{22}} - \frac{(I_{11} - I_{33}) QP}{I_{22}}$$

$$\dot{R} = \frac{M_z}{I_{33}} - \frac{(I_{22} - I_{11}) PQ}{I_{33}}$$

**Cinemática, para a posição:**

$$\vec{r} = X \vec{i} + Y \vec{j} + Z \vec{k}$$

$$\dot{\vec{r}} = \dot{X} \vec{i} + \dot{Y} \vec{j} + \dot{Z} \vec{k}$$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = S_E^T \begin{bmatrix} U \\ V \\ W \end{bmatrix}$$

$$\dot{X} = \cos \psi \cos \theta V + (\cos \psi \sin \theta \sin \phi - \cos \phi \sin \psi) V + (\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi) W$$

$$\dot{Y} = \sin \psi \cos \theta U + (\sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi) V + (\sin \psi \sin \theta \cos \phi - \sin \phi \cos \psi) W$$

$$\dot{Z} = -\sin \theta U + \cos \theta \sin \phi V + \cos \theta \cos \phi W$$

**Cinemática, para os ângulos de Euler:**

$$T = \begin{bmatrix} 1 & \tan \theta \sin \phi & \tan \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sec \theta \sin \phi & \sec \theta \cos \phi \end{bmatrix} \Rightarrow \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T \begin{bmatrix} P \\ Q \\ R \end{bmatrix}$$

$$\begin{aligned} \dot{\phi} &= P + Q \tan \theta \sin \phi + R \tan \theta \cos \phi \\ \dot{\theta} &= Q \cos \phi - R \sin \phi \\ \dot{\psi} &= Q \sec \theta \sin \phi + R \sec \theta \cos \phi \end{aligned}$$

**Cinemática, para quatérniões:**

Partindo dos ângulos de Euler iniciais são obtidos os quatérniões para o início da simulação:

$$q_0 = \pm (\cos(\phi/2) \cos(\theta/2) \cos(\psi/2) + \sin(\phi/2) \sin(\theta/2) \sin(\psi/2))$$

$$q_1 = \pm (\sin(\phi/2) \cos(\theta/2) \cos(\psi/2) - \cos(\phi/2) \sin(\theta/2) \sin(\psi/2))$$

$$q_2 = \pm (\cos(\phi/2) \sin(\theta/2) \cos(\psi/2) + \sin(\phi/2) \cos(\theta/2) \sin(\psi/2))$$

$$q_3 = \pm (\cos(\phi/2) \cos(\theta/2) \sin(\psi/2) - \sin(\phi/2) \sin(\theta/2) \cos(\psi/2))$$

Da equação da cinemática:

$$\begin{cases} P &= 2(q_0 \dot{q}_1 + q_3 \dot{q}_2 - q_2 \dot{q}_3 - q_1 \dot{q}_0) \\ Q &= 2(-q_3 \dot{q}_1 + q_0 \dot{q}_2 + q_1 \dot{q}_3 - q_2 \dot{q}_0) \\ R &= 2(q_2 \dot{q}_1 - q_1 \dot{q}_2 + q_0 \dot{q}_3 - q_3 \dot{q}_0) \end{cases}$$

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = 2 \begin{bmatrix} -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

Do facto da matriz ser ortogonal e quando invertida as equações serem bi-lineares em termos de  $q_i$  e velocidades angulares:

$$\begin{aligned} \dot{q}_0 &= -\frac{1}{2} (Pq_1 + Qq_2 + Rq_3) & \dot{q}_1 &= -\frac{1}{2} (-Pq_0 - Rq_2 + Qq_3) \\ \dot{q}_2 &= -\frac{1}{2} (-Qq_0 + Rq_1 - Pq_3) & \dot{q}_3 &= -\frac{1}{2} (-Rq_0 - Qq_1 + Pq_2) \end{aligned}$$

## Apêndice B

### • Acelerómetro

Um acelerómetro num determinado ponto  $P$  mede a componente de aceleração  $a'_p$  dada por:

$$a'_p = a_p - g \quad (90)$$

Sendo  $r$  o vector que liga o ponto de medição do acelerómetro ao centro de gravidade. A velocidade instantânea no ponto  $P$  para um espaço inercial é dada por:

$$V_P = V_{cg} + \omega \times r$$

e a aceleração inercial é obtida diferenciando esta equação em relação ao referencial inercial:

$$a_P = \frac{d(V_{cg})}{dt} + \frac{d(\omega \times r)}{dt}$$

Esta última equação pode ser expressa em termos do referencial do corpo ABC,  $R_{ABC}$ , para se determinar a saída do acelerómetro. Ao primeiro termo estão associadas as forças envolvidas, enquanto que ao segundo termo se pode usar o teorema de Coriolis para diferenciar  $(\omega \times r)$ . O resultado da soma destes dois termos será a estimativa da aceleração global e pode ser escrito como:

$$a_p = \frac{F_{ABC} + S_E m g}{m} + \dot{\omega}_{ABC} \times r + \omega_{ABC} \times (\omega_{ABC} \times r)$$

### • Câmara embarcada

Seguindo o mesmo princípio para o cálculo da matriz  $S$ , a matriz  $S_{cam}$  é calculada através das seguintes matrizes:

$$S_{cam_x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad S_{cam_y} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad S_{cam_z} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S_{cam} = S_{cam_z} S_{cam_y} S_{cam_x}$$

Atendendo às equações:

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} X_{NED} \\ Y_{NED} \\ Z_{NED} \end{bmatrix} + S^T \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{bmatrix}$$

$$\begin{cases} \phi = -\text{atan2}(S_T(2,3), S_T(3,3)) \\ \theta = -\arcsin\left(\frac{S_T(1,3)}{\det(S_T)}\right) = -\arcsin(S_T(1,3)) \\ \psi = -\text{atan2}(S_T(1,2), S_T(1,1)) \end{cases}$$

A matriz  $M$  pode ser calculada pela multiplicação das matrizes responsáveis pela adição de rotação e translacção aos pontos de referência de modo a se obter a sua projecção no referencial da imagem,  $R_{xyz}$ :

$$M_{per} = \begin{bmatrix} 0 & 0 & 0 & -1/f \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad T_{tr} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ P_x & P_y & P_z & 1 \end{bmatrix}; \quad T_{trcp} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ f - 1 \times 10^{-6} & 0 & 0 & 1 \end{bmatrix}$$

$$S_\phi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi & 0 \\ 0 & \sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad S_\theta = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad S_\psi = \begin{bmatrix} \cos \psi & -\sin \psi & 0 & 0 \\ \sin \psi & \cos \psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M = T_{tr} S_\psi S_\theta S_\phi T_{trcp} M_{per}$$

o que finalmente leva a que a projecção dos pontos seja dada por:

$$Q_P = QM$$

em que  $Q$  corresponde à matriz com as coordenadas dos pontos de referência no referencial fixo,  $R_{NED}$ :

$$Q = \begin{bmatrix} -2 & 1 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ -2 & 0 & 0 & 1 \\ -2 & -1 & 0 & 1 \end{bmatrix}$$

em que a 3ª coluna corresponde ao eixo  $z$  e a 4ª coluna à projecção segundo a unidade focal que é iniciada a 1.

## Apêndice C

---



---

```

%----- inicio.m -----
%RunScript que inicializa as principais constantes utilizadas no sistema

% Parâmetros Físicos :
% g      - aceleração gravítica [m/s^2]
% ro     - densidade do ar [g/m^3]
% ts     - amostragem [s]
% fr     - frequência
% Parâmetros estruturais, ALIV
% m      - massa [kg]
% mm     - Massa motor [Kg]
% xm     - Comprimento do motor segundo x [m]
% ym     - Comprimento do motor segundo y [m]
% zm     - Comprimento do motor segundo z [m]
% lxx    - Momento de inércia total segundo x [Kg.m^2]
% lyy    - Momento de inércia total segundo y [Kg.m^2]
% lzz    - Momento de inércia total segundo z [Kg.m^2]
% I      - Matriz simétrica inercial
% Lc     - Distância do CG ao Rotor (comprimento) [m]
% Ll     - Distância do CG ao Rotor (largura) [m]
% Parâmetros estruturais motor+hélice, ALIV
% Cp     - Coeficiente de pressão
% Ct     - Coeficiente de propulsão
% r_helice - Raio do Hélice [m]
% Vmax   - Voltagem máxima do motor [Volts]
% RPM_max - rotação máxima do motor [rpm]
% rad_max - rotação máxima do motor [rad/s]
% Kf     - F=Kf*(w^2) relação entre Força e rotação aplicada
% Km     - M=Km*(w^3) relação entre momento e rotação aplicada
% KK     - relação entre Kf e Km
% Kv     - relação entre tensão e rotação aplicada
% Vmin   - Tensão mínima ou zona morta [Volts]
% TVOLT  - Volts aplicados a cada motor tal que dw = 0 [Volts]
% rotacao0 - rotação aplicada a cada motor tal que dw = 0 [rad/s]
% Parâmetros iniciais
% Xini   - Condição inicial dos estados
% Xini2  - Condição inicial dos estados no estimador
% Parâmetros dos sensores
% ddx    - Distância do acelerómetro em relação ao CG, segundo x [m]
% ddy    - Distância do acelerómetro em relação ao CG, segundo y [m]
% ddz    - Distância do acelerómetro em relação ao CG, segundo z [m]
% cam    - Posição da câmara, referencial ABC: [x,y,z,phi,teta,psi]
% n_cp   - Distância focal inicial
% L      - Distância entre linhas [m]
% LN     - Distância entre linhas [m]
% f      - Focal da câmara
% pt_1_2 - Ponto de referência, lado direito
% pt_2_1 - Ponto de referência central
% pt_2_2 - Ponto de referência central
% pt_3_2 - Ponto de referência, lado esquerdo
% noise  - Ruído associado às medições do magnetómetro
% n      - número de bits
% ymin   - Valor de saída máximo para o magnetómetro
% ymax   - Valor de saída mínimo para o magnetómetro
% resolucao - Gama de resolução
% Vectors com os estados iniciais
% X_0J   - Vector com os estados iniciais (caso do Joystick)
% X_0    - Vector com os estados iniciais (caso da câmara)
% XU_0   - Vector com a rotacao0 e os estados iniciais de referência
% U_0    - Vector com a rotacao0
% Y_0    - Vector com as saídas dos três sensores
% Y_0J   - Vector com as saídas do acelerómetro e magnetómetro
%-----

clear all
global w_motor0 Kf KK m g I Ll Lc fr rotacao0 ddx ddy ddz cam L LN f
syms TVOLT

%-----
%Parâmetros Físicos

g = 9.81; %Aceleração gravítica [m/s^2]
ro = 1.225; %Densidade do ar [Kg/m^3]
ts = 0.02; %Amostragem [ms]
fr = 16; %Frequência [Hz]
%-----

```



```

%Parâmetros estruturais , ALIV

m = 1.2; %Massa total [Kg]
mm = 0.060; %Massa motor [Kg]
xm = 0.015; %Comprimento do motor segundo x [m]
ym = 0.015; %Comprimento do motor segundo y [m]
zm = 0.032; %Comprimento do motor segundo z [m]
Lc = 0.64; %Distância do CG ao Rotor (comprimento)
Ll = 0.6; %Distância do CG ao Rotor (largura)

lx1 = (1/12)* mm * ((ym^2) + (zm^2)); lx2 = lx1;
lx3 = lx1 + (mm*(Lc^2)); lx4 = lx3;
lxx = lx1 + lx2 + lx3 + lx4; %Momento de inercia total segundo x

ly3 = (1/12)* mm * (xm^2 + zm^2); ly4 = ly3;
ly1 = ly3 + (mm*(Ll^2)); ly2 = ly1;
lyy = ly1 + ly2 + ly3 + ly4; %Momento de inercia total segundo y

lz1 = (1/12)* mm * (xm^2 + zm^2) + mm*(Lc^2); lz2 = lz1;
lz3 = (1/12)* mm * (xm^2 + zm^2) + mm*(Lc^2); lz4 = lz3;
lzz = lz1 + lz2 + lz3 + lz4; %Momento de inercia total segundo z

[I] = [lxx 0 0 ;
       0 lyy 0 ;
       0 0 lzz]; %Matriz simetrica inercial

%
%Parâmetros estruturais motor+hélice , ALIV

Cp = 0.0374; %Coeficiente de Pressão
Ct = 0.1047; %Coeficiente de prepulsão
r_helice = 0.2458/2; %Raio do Hélice [m] = diametro 9''
Vmax = 12; %Voltagem máxima do motor
RPM_max = 6000; %rotação máxima do motor [rpm]

%Valor constante que relaciona Rotação e a Força aplicada ao motor
Kf = double(4*ro*(r_helice^4)*Ct/(pi^2)); %F=Kf*(w^2)

%Valor constante que relaciona Rotação e o Momento aplicado ao motor
Km = double(4*ro*(r_helice^5)*Cp/(pi^3)); %M=Km*(w^3)

%Relaciona Kf e Km
KK = double(r_helice*Cp/(Ct*pi));
rad_max = double(RPM_max*2*pi/60); %RPM máximo do motor em rad/s
Kv = double(2000*2*pi/60); %relação entre tensão e velocidade
Vmin = double(Vmax - rad_max/Kv); %tensão minima ou zona morta

%Volts aplicados a cada motor tal que dw = 0 (assumidos todos iguais)
TVOLT = solve((TVOLT-9)^2 - m*g/(Kf*(Kv^2)*4),TVOLT);
TVOLT = double(TVOLT(1));

%Velocidade de rotação aplicados a cada motor tal que dw = 0
rotacao0 = double(Kv*(TVOLT-Vmin)); %Volts para rad/s
w_motor0 = [rotacao0 rotacao0 rotacao0 rotacao0];

%Aproximação inicial: Relação tensão/rotação nos motores
wtoN = 2*Kf*rotacao0; %Motores idênticos

%
%Parâmetros iniciais

Xini = [0 0 0 0 0 0 0 0 -5 1 0 0 0]; %Condições iniciais para X
Xini2 = [0 0 0 0 0 0 0 0 0 0 0 0]; %Condições iniciais para estimador

%Estados iniciais para a posição de referência
U0 = 0; %Velocidade Linear segundo X,NED
V0 = 0; %Velocidade Linear segundo Y,NED
W0 = 0; %Velocidade Linear segundo Z,NED
P0 = 0; %Velocidade Ângular segundo Ex,ABC
Q0 = 0; %Velocidade Ângular segundo Ey,ABC
R0 = 0; %Velocidade Ângular segundo Ez,ABC
X0_NED = 0; %Posição segundo X, NED
Y0_NED = 0; %Posição segundo Y, NED
Z0_NED = -7; %Posição segundo Z, NED
phi_0 = 0; %Ângulo de Rolamento
teta_0 = 0; %Ângulo de Picada
ppsi_0 = 0; %Ângulo de Guinada
q0_0 = 1; %Quaternião inicial q0
q1_0 = 0; %Quaternião inicial q1
q2_0 = 0; %Quaternião inicial q2
q3_0 = 0; %Quaternião inicial q3
save('constantes '); %Guarda o valor das constantes

```

```

%-----
%Parâmetros dos sensores

    noise = 1e-3;
    n = 10; %número de bits
    ymin = -2*g; %Valor de saída máximo
    ymax = 2*g; %Valor de saída mínimo
    resolucao = (ymax - ymin)/(2^n); %Gama de resolução

%Acelerómetro
    ddx = 0; %Distância em relação ao CG segundo x
    ddy = -0.005; %Distância em relação ao CG segundo y
    ddz = 0.03; %Distância em relação ao CG segundo z

%Câmara embarcada
    w=800/2;
    hh=600/2;
    deg = pi/180;
    n_cp = 1;
    cam = [.01 0 0.01 0 -90*deg 0]'; %Posição da câmara, ABC
    L = 2; %distância entre as linhas esq e dir
    LN = 2; %distância
    f = 2.5; %focal da câmara
    pt_1_2 = [LN L/2 0]; %ponto de referência, direito
    pt_2_1 = [-LN 0 0]; pt_2_2 = [LN 0 0]; %pontos de referência centrais
    pt_3_2 = [LN -L/2 0]; %ponto de referência, esquerdo
    tx = 1.5;
    ty = 1.5;

%-----
%Vectores com os estados iniciais, usados para a construção das matrizes numéricas

%Vector com os estados iniciais (caso do Joystick)
    X_OJ = [P0 Q0 R0 phi_0 teta_0 ppsi_0]';

%Vector com os estados iniciais (caso da câmara)
    X_0 = [U0 V0 W0 P0 Q0 R0 X0_NED Y0_NED Z0_NED phi_0 teta_0 ppsi_0]';

%Vector com os estados iniciais mais a rotação de cada motor inicial
    XU_0 = [w_motor0 U0 V0 W0 P0 Q0 R0 X0_NED Y0_NED Z0_NED phi_0 teta_0 ppsi_0]';

%Vector com a rotação inicial, rad/s
    U_0 = w_motor0;

%Vector com as saídas dos sensores correspondentes aos estados iniciais
    Y_0= sensor(XU_0);
    Y_OJ = zeros(1,6);
    Y_OJ(1:3) = Y_0(1:3); %Construção do vector de saída
    Y_OJ(4:6) = Y_0(12:14); %para o caso do joystick, 6 saídas

%-----
%Construção das matrizes numéricas, matrizes de ganho e de kalman para ambos os casos, módulo com
%joystick e módulo de visualização

%Matrizes numéricas do sistema para posição referência
    [A_num, B_num, C_num, D_num] = numerico (XU_0);

%Matrizes de ganho e de Kalman para construção do controlador
    [K, KJ, A_kd, B_kd, C_kd, D_kd, MM, A_kdJ, B_kdJ, C_kdJ, D_kdJ] = controlo (A_num, B_num, ...
    C_num, D_num);

%-----
%-----
%-----
function [A_num, B_num, C_num, D_num] = numerico (XU_0)
% Função que realiza as operações numéricas para obtenção das matrizes
% necessárias para a utilização no LQR e do Filtro de Kalman usando
% espaço de estados

% Parâmetros de entrada:
% XU_0 - Vector com a rotacao0 e os estados iniciais de referência
% Parâmetro de saída:
% A_num - Matriz A numérica para espaço de estados
% B_num - Matriz B numérica para espaço de estados
% C_num - Matriz C numérica para espaço de estados
% D_num - Matriz D numérica para espaço de estados
% Linearização numérica das matrizes A e B
% J - Matriz Jacobiana
% XU_0 - Vector com a rotacao0 e os estados iniciais de referência
% dxdt_0 - Estados iniciais para valores de referência
% dd - Perturbação
% X_1 - Vector com perturbação numa dada referência

```

```

% dxdt_1 – Vector com stados após perturbação num valor de referência
% Linearização numérica das matrizes C e D
% Z – matriz Jacobiana
% dxdt_0 – Saídas iniciais para valores de referência
% X_1 – Vector com perturbação numa dada referência
% dxdt_1 – Vector com saídas após perturbação num valor de referência
%
%Parâmetros iniciais , de referência

w_motor0 = [XU_0(1) XU_0(2) XU_0(3) XU_0(4)]; %Rotação em cada um dos motores

%Valores de referência
U0 = XU_0(5); %Velocidade Linear segundo X,NED
V0 = XU_0(6); %Velocidade Linear segundo Y,NED
W0 = XU_0(7); %Velocidade Linear segundo Z,NED
P0 = XU_0(8); %Velocidade Angular segundo Ex,ABC
Q0 = XU_0(9); %Velocidade Angular segundo Ey,ABC
R0 = XU_0(10); %Velocidade Angular segundo Ez,ABC
X0_NED = XU_0(11); %Posição segundo X, NED
Y0_NED = XU_0(12); %Posição segundo Y, NED
Z0_NED = XU_0(13); %Posição segundo Z, NED
phi_0 = XU_0(14); %Ângulo de Rolamento
teta_0 = XU_0(15); %Ângulo de Picada
ppsi_0 = XU_0(16); %Ângulo de Guinada

quat = euler2quat([phi_0 teta_0 ppsi_0]); %Passagem para quaterniões
q0_0 = quat(1); %Quaterniões inicial q0
q1_0 = quat(2); %Quaterniões inicial q1
q2_0 = quat(3); %Quaterniões inicial q2
q3_0 = quat(4); %Quaterniões inicial q3

%
%Linearização numérica das matrizes A e B

%Vector com o estado inicial
X_0 = [w_motor0 U0 V0 W0 P0 Q0 R0 X0_NED Y0_NED Z0_NED q0_0 q1_0 q2_0 q3_0]';

dxdt_0 = modelo_sistema(X_0)'; %Estados iniciais
dd=1e-5;J=[]; %Perturbação
for k=1:1:17 %Construção da matriz A e B
X_1=X_0;X_1(k)=X_1(k)+dd; %Estado + perturbação
dxdt_1=modelo_sistema(X_1)'; %Derivada resultante
J=[J (dxdt_1-dxdt_0)/dd]; %Construção das colunas
end %da matriz auxiliar

%linearizacao com os ângulos de euler
for k=1:3
X_1=X_0;eul_1=zeros(3,1);eul_1(k)=dd; %Perturbação quaterniões
q_1=euler2quat(eul_1)'; %equivalentes a euler
X_1(4+9+(1:4))=q_1; %Substituição nos estados
dxdt_1=modelo_sistema(X_1)'; %Derivada resultante
J=[J (dxdt_1-dxdt_0)/dd]; %Adição das novas colunas
end

for k=1:1:16 %Substituição da equação referente aos
%quaterniões

X_0 = [w_motor0 U0 V0 W0 P0 Q0 R0 X0_NED Y0_NED Z0_NED phi_0 teta_0 ppsi_0]';
X_0(k) = X_0(k)+ dd;
[T] = [1 tan(X_0(15))*sin(X_0(14)) tan(X_0(15))*cos(X_0(14)) ;
0 cos(X_0(14)) -sin(X_0(14)) ;
0 (1/cos(X_0(15)))*sin(X_0(14)) (1/cos(X_0(15)))*cos(X_0(14))];

%Equações de Euler
Euler_num(1:3,k) = (T*[X_0(8) X_0(9) X_0(10)]')/dd;
end

%Construção das matrizes numéricas A e B
A_num(1:9,1:9) = J(1:9,5:13); %Utilização das colunas de J
A_num(1:9,10:12) = J(1:9,18:20);
A_num(10:12,1:12) = zeros(3,12);
A_num(10:12,13:16) = Euler_num(1:3,5:16); %Matriz A numérica
B_num(1:9,1:4) = J(1:9,1:4);
B_num(10:12,1:4) = zeros(3,4); %Matriz B numérica

%
%Linearização numérica das matrizes C e D

Z=[];
X_0 = [w_motor0 U0 V0 W0 P0 Q0 R0 X0_NED Y0_NED Z0_NED phi_0 teta_0 ppsi_0];

dxdt_0 = sensor(X_0);

```

```

for k=1:1:16
    X_1=X_0;X_1(k)=X_1(k)+dd;
    dxdt_1=sensor(X_1);
    Z=[Z (dxdt_1'-dxdt_0')/dd];
end
%Construção da matriz C e D
%Estado + perturbação
%Derivada resultante
%Construção das colunas
%da matriz auxiliar

%Construção das matrizes numéricas C e D
C_num(1:14,1:12) = Z(1:14,5:16);
D_num(1:14,1:4) = Z(1:14,1:4);
%Matriz C numérica
%Matriz D numérica

```

---

```

modelo_sistema.m

```

---

```

function [X_ponto] = modelo_sistema (IN)
% Aplica a dinâmica e a cinemática aos estados actuais do Quadrirotor. Simula a dinâmica,
% e serve para construir as matrizes A e B numéricas através da função numerico.m

% Parâmetros de entrada:
% IN - Vector com a rotacao0 e os estados actuais
% Parâmetro de saída:
% X_ponto - Vector com variação de cada um dos estados a ser integrado
% Matriz Transformação
% S - Matriz de transformação, quatérniões
% Forças e Momentos
% F - Força exercida por cada motor [N]
% Fx - Força segundo x [N]
% Fy - Força segundo y [N]
% Fz - Força segundo z [N]
% Mx - Momento segundo x [Nm]
% My - Momento segundo y [Nm]
% Mz - Momento segundo z [Nm]
% Equações da dinâmica
% Vlinear - Vector com a variação das velocidades lineares
% Vrotacao - Vector com a variação das velocidades angulares
% Equações da Cinemática
% Posicao - Vector com a variação da posição NED
% T - Matriz de rotação
% dq - Vector com a variação dos quatérniões
%

```

---

```

global Kf KK m g l Lc Ll
%
%Estados actuais do Quadrirotor

w_motor = [IN(1) IN(2) IN(3) IN(4)];
U = IN(5);
V = IN(6);
W = IN(7);
P = IN(8);
Q = IN(9);
R = IN(10);
X_NED = IN(11);
Y_NED = IN(12);
Z_NED = IN(13);
q0 = IN(14);
q1 = IN(15);
q2 = IN(16);
q3 = IN(17);
%rotação dos motores, rad/s
%Velocidade Linear segundo X,NED
%Velocidade Linear segundo Y,NED
%Velocidade Linear segundo Z,NED
%Velocidade Angular segundo Ex,ABC
%Velocidade Angular segundo Ey,ABC
%Velocidade Angular segundo Ez,ABC
%Posição segundo x, NED
%Posição segundo y, NED
%Posição segundo z, NED
%Quatérnião q0
%Quatérnião q1
%Quatérnião q2
%Quatérnião q3
%
%Matriz Transformação

[S] = [q0^2 + q1^2 - q2^2 - q3^2      2*(q1*q2 + q0*q3)      2*(q1*q3 - q0*q2)      ;
       2*(q1*q2 - q0*q3)      q0^2 - q1^2 + q2^2 - q3^2      2*(q2*q3 + q0*q1)      ;
       2*(q1*q3 + q0*q2)      2*(q2*q3 - q0*q1)      q0^2 - q1^2 - q2^2 + q3^2 ];
%
%Forças e Momentos

for i=1:4
    F(i) = Kf*(w_motor(i)^2);
end

Fx = 0;
Fy = 0;
Fz = -(F(3)+F(1)+F(4)+F(2));
Mx = (F(1)-F(2))*Ll;
My = (F(3)-F(4))*Lc;
Mz = KK*(-F(3)+F(1)-F(4)+F(2));
%Força segundo x desprezavel
%Força segundo y desprezavel
%Força segundo z, negativa
%Momento segundo x
%Momento segundo y
%Momento segundo z
%
%Equações da Dinâmica

```

```

%Segunda lei de Newton
Vlinear = [Fx Fy Fz]'/m + S*[0 0 g]' - ([Q*W-R*V R*U-P*W P*V-Q*U]');

%Aplicação da Lei dos momentos
Vrotacao = inv(I)*([Mx My Mz]') + [(I(2,2)-I(3,3))*R*Q/I(1,1) ...
(I(3,3)-I(1,1))*R*P/I(2,2) (I(1,1)-I(2,2))*P*Q/I(3,3)]';

%-----
%Equações da Cinemática

%Cinemática para a posição
Posicao = S'*[U V W]';
[T] = [ 0 P Q R ;
-P 0 -R Q ;
-Q R 0 -P ;
-R -Q P 0];

%Cinemática para a rotação
lambda = 1 - [q0 q1 q2 q3]*[q0 q1 q2 q3]';
[dq] = (-1/2)*T*[q0 q1 q2 q3]' + lambda * [q0 q1 q2 q3]';

%-----
%Vector de saída com os Estados

X_ponto = [Vlinear' Vrotacao' Posicao' dq'];

----- sensor.m -----

function [Y_sensores] = sensor(X)
% Função que simula os sensores, três acelerómetros, magnetometro, e sensor de posição baseado
% numa câmara embarcada no Quadrirotor

% Parâmetros de entrada:
% X - Vector com a rotacao0 e os estados actuais
% Parâmetro de saída:
% Y_sensores - Vector com as saídas dos Sensores
% Matriz Transformação
% S - Matriz de transformação, Euler
% Forças e Momentos
% F - Força exercida por cada motor [N]
% Fx - Força segundo x [N]
% Fy - Força segundo y [N]
% Fz - Força segundo z [N]
% Mx - Momento segundo x [Nm]
% My - Momento segundo y [Nm]
% Mz - Momento segundo z [Nm]
% Sensor - Acelerómetro
% A_sensor - Vector com as acelerações totais [m/s^2]
% Sensor - Câmara embarcada
% pts - Matriz com os pontos de referência
% NED - Posição actual do Quadrirotor [X Y Z phi teta psi]
% Pos_sensor - Vector com a projecção dos pontos de referência na imagem
% Sensor - Magnetómetro
% Mag_sensor - Vector com a orientação do magnetómetro

global g Kf KK Ll Lc m I ddx ddy ddz cam L LN f

%-----
%Estados actuais do Quadrirotor
w_motor = [X(1) X(2) X(3) X(4)]; %rotação dos motores, rad/s
U = X(5); %Velocidade Linear segundo X,NED
V = X(6); %Velocidade Linear segundo Y,NED
W = X(7); %Velocidade Linear segundo Z,NED
P = X(8); %Velocidade Angular segundo Ex,ABC
Q = X(9); %Velocidade Angular segundo Ey,ABC
R = X(10); %Velocidade Angular segundo Ez,ABC
X_NED = X(11); %Posição segundo x, NED
Y_NED = X(12); %Posição segundo y, NED
Z_NED = X(13); %Posição segundo z, NED
phi = X(14); %Ângulo de rolamento
teta = X(15); %Ângulo de picada
ppsi = X(16); %Ângulo de guinada

%-----
%Matriz Transformação

[S] = [ cos(teta)*cos(ppsi) cos(teta)*sin(ppsi) ...
-sin(teta) ;
(sin(phi)*sin(teta)*cos(ppsi))-(cos(phi)*sin(ppsi)) ...
];

```

```

        (sin(ppsi)*sin(teta)*sin(phi))+cos(ppsi)*cos(phi)   sin(phi)*cos(teta)   ;
        (cos(ppsi)*sin(teta)*cos(phi))+sin(ppsi)*sin(phi)   ...
        (cos(phi)*sin(teta)*sin(ppsi))-sin(phi)*cos(ppsi)   cos(phi)*cos(teta)   ];
%-----
%Forças e Momentos

for i=1:1:4
    F(i) = Kf*(w_motor(i)^2);
end

Fx = 0; %Força segundo x desprezavel
Fy = 0; %Força segundo y desprezavel
Fz = -(F(3)+F(1)+F(4)+F(2)); %Força segundo z, negativa
Mx = (F(1)-F(2))*Ll; %Momento segundo x
My = (F(3)-F(4))*Lc; %Momento segundo y
Mz = Kk*(-F(3)+F(1)-F(4)+F(2)); %Momento segundo z
%-----
%Sensor - Acelerómetro

Vlinear = [Fx Fy Fz]'/m + S*[0 0 g]' - ([Q*W-R*V R*U-P*W P*V-Q*U]');

dU = Vlinear(1); %Aceleração linear segundo x, ABC
dV = Vlinear(2); %Aceleração linear segundo y, ABC
dW = Vlinear(3); %Aceleração linear segundo z, ABC

Vrotacao = inv(I)*([Mx My Mz]') + [(I(2,2)-I(3,3))*R*Q/I(1,1) ...
    (I(3,3)-I(1,1))*R*P/I(2,2) (I(1,1)-I(2,2))*P*Q/I(3,3)]';

dP = Vrotacao(1); %Aceleração angular segundo x, ABC
dQ = Vrotacao(2); %Aceleração angular segundo y, ABC
dR = Vrotacao(3); %Aceleração angular segundo z, ABC

A_sensor = [dU dV dW]' + [(dQ*ddz - dR*ddy) (dR*ddx - dP*ddz) (dP*ddy - dQ*ddx)]' + ...
    [(Q*(P*ddy - Q*ddx)) (R*(Q*ddz - P*ddy)) (P*(R*ddx - P*ddz))] - ...
    [(R*(R*ddx - P*ddz)) (P*(P*ddy - Q*ddx)) (Q*(Q*ddz - P*ddy))] + S*[0 0 g]';
%-----
%Sensor - Câmara embarcada

pt_1_2 = [-LN L/2 0]; %ponto de referência, direito
pt_2_1 = [ LN 0 0]; pt_2_2=[-LN 0 0]; %pontos de referência centrais
pt_3_2 = [-LN -L/2 0]; %ponto de referência, esquerdo
pts=[pt_1_2 1; pt_2_1 1; pt_2_2 1; pt_3_2 1];

NED = [X_NED Y_NED Z_NED phi teta ppsi]; %Posição do Quadrirotor

%Projeção dos pontos de referência na imagem
img_pts = proj_pts(NED, [cam; f], pts);
img_pts(:,2) = -img_pts(:,2); %Correcção de eixo
x1 = img_pts(1,1); x2 = img_pts(2,1); x3 = img_pts(3,1); x4 = img_pts(4,1);
y1 = img_pts(1,2); y2 = img_pts(2,2); y3 = img_pts(3,2); y4 = img_pts(4,2);
Pos_sensor = [x1 y1 x2 y2 x3 y3 x4 y4]; %Posição xy dos pontos de referência na
%imagem
%-----
%Sensor - Magnetómetros

Mag_sensor = S*[1 0 0]'; %Vector com orientação do magnetómetro
%-----
%Vector com as saídas dos Sensores

Y_sensores = [A_sensor' Pos_sensor Mag_sensor'];
%-----
proj_pts.m
%-----
function y = proj_pts(NED,cam,pts)

% Função que calcula a projecção dos pontos de referência no referencial de imagem, Rxyz, com base
% na posição actual do Quadrirotor e das características da câmara

% Parâmetros de entrada:
% NED - Posição actual do Quadrirotor [X Y Z phi teta psi]
% cam - Posição e atitude da câmara em relação ao referencial ABC
% pts - Matriz com os pontos de referência
% Parâmetro de saída:
% y - Vector com a posição x e y da projecção dos pontos
% Matrizes de transformação entre RNED e Rabc
% tff0 - Matriz de transformação para phi entre RNED e RABC
% ttt0 - Matriz de transformação para teta entre RNED e RABC

```

```

%   tty0  - Matriz de transformação para psi entre RNED e RABC
%   S     - Matriz de transformação entre RNED e RABC
%   tffc0 - Matriz de transformação para phi entre RABC e Rxyz
%   tttcam - Matriz de transformação para teta entre RABC e Rxyz
%   tyycam - Matriz de transformação para psi entre RABC e Rxyz
%   Scam  - Matriz de transformação entre RABC e Rxyz
%   ST    - Matriz de transformação entre RNED e Rxyz
% Posição e atitude globais da câmara em relação ao referencial RNED
%   P     - Posição da câmara em relação ao referencial RNED [m]
%   phi   - Ângulo de rolamento [rad]
%   teta  - Ângulo de picada [rad]
%   psi   - Ângulo de guinada [rad]
% Matriz de transformação para a projecção dos pontos no referencial Rxyz
%   M     - Matriz de projecção dos pontos entre RNED e Rxyz
%
%-----
x_cam = cam(1); y_cam = cam(2); z_cam = cam(3);   phi_cam = cam(4); teta_cam = cam(5);
psi_cam = cam(6);
f=cam(7);
X_0 = NED(1); Y_0 = NED(2); Z_0 = NED(3); phi_0 = NED(4); teta_0 = NED(5); psi_0 = NED(6);
%-----
%Matrizes de transformação

tff0 = [1      0      0      ; 0  cos(phi_0) sin(phi_0) ; 0 -sin(phi_0) cos(phi_0)];
ttt0 = [cos(teta_0) 0 -sin(teta_0) ; 0      1      0      ; sin(teta_0) 0 cos(teta_0)];
tty0 = [ cos(psi_0) sin(psi_0) 0      ; -sin(psi_0) cos(psi_0) 0      ; 0      0      1];

S = tty0*ttt0*tff0; %Matriz de transformação entre RNED e RABC

tffc0 = [1      0      0      ; 0  cos(phi_cam) sin(phi_cam) ; 0 -sin(phi_cam) cos(phi_cam)];
tttcam = [cos(teta_cam) 0 -sin(teta_cam); 0      1      0      ; sin(teta_cam) 0 cos(teta_cam)];
tyycam = [ cos(psi_cam) sin(psi_cam) 0      ; -sin(psi_cam) cos(psi_cam) 0      ; 0      0      1];

Scam = tyycam*tttcam*tffc0; %Matriz de transformação entre RABC e Rxyz
ST = Scam*S; %Matriz de transformação entre RNED e Rxyz
%-----
%Posição e atitude globais da câmara em relação ao referencial RNED

P = [X_0;Y_0;Z_0] + S'*[x_cam;y_cam;z_cam]; %Posição da câmara em relação ao referencial RNED [m]
phi = -atan2(ST(2,3),ST(3,3)); %Ângulo de rolamento [rad]
teta = -asin(ST(1,3)/det(ST)); %Ângulo de picada [rad]
psi = -atan2(ST(1,2),ST(1,1)); %Ângulo de guinada [rad]
if teta==0; teta=1e-6; end;
%-----
%Matriz de transformação para a projecção dos pontos no referencial Rxyz

n_cp=1; %unidade focal admitida como 1
x_prp = P(1); y_prp = -P(2); z_prp = P(3);
x_cp=n_cp-1e-6;tetacam=0;

M_per = [0 0 0 -1/n_cp ; 0 1 0 0 ; 0 0 1 0 ; 0 0 0 1 ];
T_tr = [ 1 0 0 0 ; 0 1 0 0 ; 0 0 1 0 ; x_prp y_prp z_prp 1 ];
B_psi = [cos(psi) -sin(psi) 0 0 ; sin(psi) cos(psi) 0 0 ; 0 0 1 0 ; 0 0 0 1];
B_teta = [ cos(teta) 0 sin(teta) 0 ; 0 1 0 0 ; -sin(teta) 0 cos(teta) 0 ; 0 0 0 1];
B_phi = [1 0 0 0 ; 0 cos(phi) -sin(phi) 0 ; 0 sin(phi) cos(phi) 0 ; 0 0 0 1];
B_tetacam = [ cos(tetacam) 0 sin(tetacam) 0 ; 0 1 0 0 ; -sin(tetacam) 0 cos(tetacam) 0 ; 0 0 0 1];
T_trcp = [ 1 0 0 0 ; 0 1 0 0 ; 0 0 1 0 ; x_cp 0 0 1];

%Matriz de projecção dos pontos entre RNED e Rxyz
M_total = T_tr*B_psi*B_teta*B_phi*B_tetacam*T_trcp*M_per;
%-----
%Vector de saída com a posição x e y da projecção dos pontos

pts(:,1) = -pts(:,1); %Correcção da posição x
pts_p = pts*M_total; %Projecção dos pontos
len_pp = length(pts_p(:,1));
for i = 1:len_pp, pts_p(i,:) = pts_p(i,:)/pts_p(i,4);
end

pts_p(:,1:3) = pts_p(:,1:3)*f; %Aplicação da unidade focal correcta
y = pts_p(1:len_pp,2:3); %posição x e y da projecção dos pontos
%-----
controlo.m
%-----
function [K, KJ, A_kd, B_kd, C_kd, D_kd, MM, A_kdJ, B_kdJ, C_kdJ, D_kdJ] = controlo (A_num, ...
B_num, C_num, D_num)
% Função que através das matrizes numéricas calcula os ganhos para o LQR e as matrizes discretas
% a serem usadas no filtro de kalman para ambas as situações, com Joystick(2 sensores), com

```

```

% câmara integrada (Voo autónomo)

% Parâmetro de entrada:
% A_num – Matriz A numérica para espaço de estados
% B_num – Matriz B numérica para espaço de estados
% C_num – Matriz C numérica para espaço de estados
% D_num – Matriz D numérica para espaço de estados

% Parâmetro de saída:
% K – Ganho discreto LQR (com a câmara embarcada)
% KJ – Ganho discreto LQR (Para o caso do Joystick)
% A_kd – Matriz A discreta para F.Kalman em espaço estados (câmara)
% B_kd – Matriz B discreta para F.Kalman em espaço estados (câmara)
% C_kd – Matriz C discreta para F.Kalman em espaço estados (câmara)
% D_kd – Matriz D discreta para F.Kalman em espaço estados (câmara)
% MM – Matriz de interface com Joystick
% A_kdJ – Matriz A discreta para F.Kalman em espaço estados (Joystick)
% B_kdJ – Matriz B discreta para F.Kalman em espaço estados (Joystick)
% C_kdJ – Matriz C discreta para F.Kalman em espaço estados (Joystick)
% D_kdJ – Matriz D discreta para F.Kalman em espaço estados (Joystick)
% Situação autónoma
% Q – Matriz de ponderação Q para aplicação do controlo moderno
% R – Matriz de ponderação R para aplicação do controlo moderno
% K – Matriz ganho, LQR discreto
% G – Matriz de covariância
% Qc – Matriz de covariância
% Rc – Matriz de covariância
% H – Matriz de covariância
% sys2 – Sistema em espaço de estados
% KEST – Sistema em espaço de estados para o estimador
% Situação controlada através do joystick
% A – Nova matriz A, 6 estados, [p q r phi teta psi]
% B – Nova matriz B, 6 estados, [p q r phi teta psi]
% C – Nova matriz C, 6 saídas, [A_sensor Mag_sensor]
% D – Nova matriz D, 6 saídas, [A_sensor Mag_sensor]
% QJ – Matriz de ponderação Q para aplicação do controlo moderno
% RJ – Matriz de ponderação R para aplicação do controlo moderno
% KJ – Matriz ganho, LQR discreto
% C1 – Matriz para saídas [u v w r] para construção da matriz MM
% D1 – Matriz para saídas [u v w r] para construção da matriz MM
% MM – Matriz de interface com Joystick
% GJ – Matriz de covariância
% QcJ – Matriz de covariância
% RcJ – Matriz de covariância
% HJ – Matriz de covariância
% sys2J – Sistema em espaço de estados
% KESTJ – Sistema em espaço de estados para o estimador
%


---


global fr
t_s = 1/fr;
%
%Situação autónoma

%Matriz de ponderação Q para aplicação do controlo moderno LQR
Q = diag([25,25,25,250,250,250,6.25/10,6.25/10,6.25/10,418,418,418]);

%Matriz de ponderação R para aplicação do controlo moderno LQR
R = (1e-1)*diag([1,1,1,1]);

K = lqrd(A_num,B_num,Q,R,t_s); %LQR do sistema, Matriz K

%Filtro de Kalman
dd=(1e-4);
G = eye(12,6);
Qc = diag([dd dd dd dd dd dd]);
Rc = diag([dd dd dd dd dd dd dd dd dd dd dd dd]);
H = 0*eye(14,6);
sys2 = ss(A_num,[B_num G],C_num,[D_num H]); %Passagem espaço estados
[KEST] = kalmd(sys2,Qc,Rc,t_s); %Filtro de Kalman
[A_kd, B_kd, C_kd, D_kd] = ssdata(KEST); %Obtenção das matrizes
%


---


%Situação controlada através do joystick

A(1:3,1:3) = A_num(4:6,4:6);
A(1:3,4:6) = A_num(4:6,10:12);
A(4:6,1:3) = A_num(10:12,4:6);
A(4:6,4:6) = A_num(10:12,10:12); %Nova matriz A, 6 estados
B(1:3,1:4) = B_num(4:6,1:4);
B(4:6,1:4) = B_num(10:12,1:4); %Nova matriz B, 6 estados

```



```

C(1:3,1:3) = C_num(1:3,4:6);
C(1:3,4:6) = C_num(1:3,10:12);
C(4:6,1:3) = C_num(12:14,4:6);
C(4:6,4:6) = C_num(12:14,10:12);           %Nova matriz C, 6 saídas
D(1:3,1:4) = D_num(1:3,1:4);
D(4:6,1:4) = D_num(12:14,1:4);           %Nova matriz D, 6 saídas

%Matriz de ponderação Q para aplicação do controlo moderno LQR
QJ = (1e-2)*diag([364,364,364,364,364,364]);

%Matriz de ponderação R para aplicação do controlo moderno LQR
RJ = (1e0)*diag([10,10,10,10]);

KJ = lqrd(A,B,QJ,RJ,t_s);                 %LQR do sistema , Matriz K

%Filtro de Kalman
ddJ=(1e-6);
GJ = eye(6,6);
QcJ = diag([ddJ ddJ ddJ ddJ ddJ ddJ]);
RcJ = diag([ddJ ddJ ddJ ddJ ddJ ddJ]);
HJ = 0*eye(6,6);
sys2J = ss(A,[B GJ],C,[D HJ]);           %Passagem espaço estados
[KESTJ] = kalmd(sys2J,QcJ,RcJ,t_s);      %Filtro de Kalman
[A_kdJ, B_kdJ, C_kdJ, D_kdJ] = ssdata(KESTJ); %Obtenção das matrizes

C1(1:3,1:3) = A_num(1:3,4:6);
C1(1:3,4:6) = A_num(1:3,10:12);
C1(4,1:3) = A_num(4,4:6);
C1(4,4:6) = A_num(4,10:12);             %Matriz para as 4 saídas [du dv dw dr]
D1(1:3,1:4) = B_num(1:3,1:4);
D1(4,1:4) = B_num(6,1:4);              %Matriz para as 4 saídas [du dv dw dr]
MM = ((-1)*C1*inv(A-B*KJ)*B + D1);     %Interface joystick keyboard

```

## Apêndice D

### • Diagrama de blocos

Até este ponto, os ficheiros usados para a construção das matrizes iniciais, nomeadamente o ficheiro *dinamica.m* e *sensor.m* (Apêndice C) eram, durante a fase de simulação, utilizados posteriormente como *MATLAB Functions* em ambiente SIMULINK. Com a necessidade de construção de um ambiente em tempo real, estas mesmas funções tiveram de ser transformadas em diagramas de blocos. As figuras que se seguem exemplificam o resultado destas mesmas transformações.

A dinâmica do sistema foi construída através de blocos que representassem as variações dos estados, Figura 65.

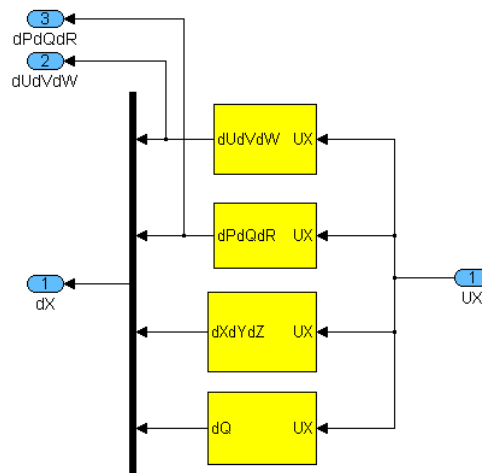


Figura 65: Equações em diagrama de blocos que definem a dinâmica do ALIV

Cada um destes blocos foi construído com base na modelação adiantada na Secção 3.2.2 e apresentada no ficheiro *dinamica.m*. Este bloco recebe os estados actuais e a rotação dos motores, e calcula a variação dos estados globais a serem posteriormente integrados.

Em relação ao ficheiro *sensor.m*, cada um dos sensores do sistema foi desenvolvido num bloco principal, Figura 66. Este bloco recebe as variáveis de entrada dos sensores, e produz as saídas que serão fornecidas ao filtro de Kalman consoante os sensores utilizados.

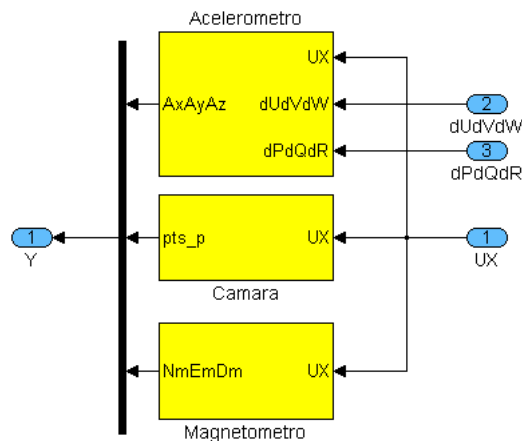


Figura 66: Equações em diagrama de blocos que definem os sensores do ALIV

- Criação do executável

Para a criação do executável é necessário definir os parâmetros de resolução do sistema através do “Solver” no “Configuration Parameters” do SIMULINK, Figura 67.

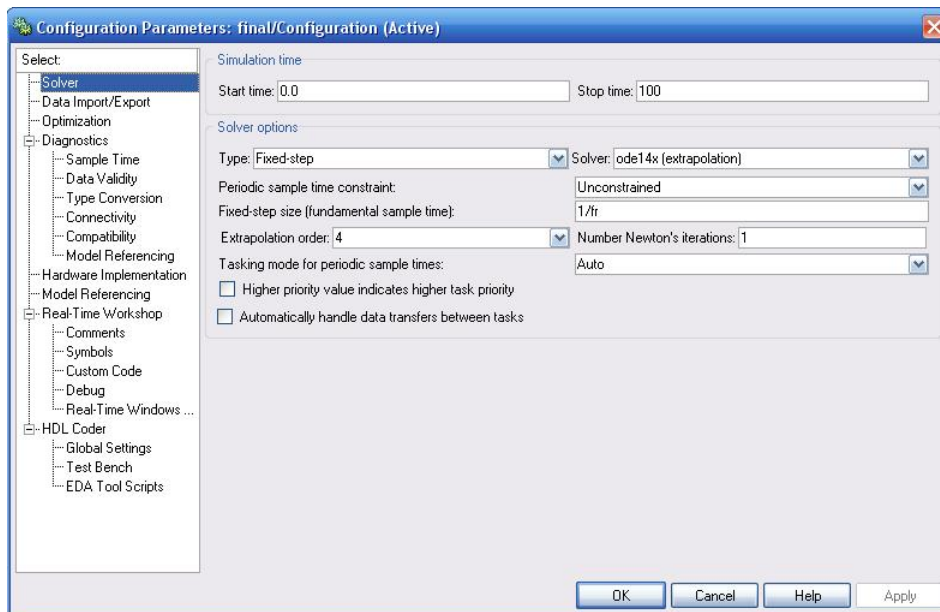


Figura 67: Parâmetros adoptados para a resolução da simulação

Depois é necessário definir a configuração adoptada para a geração do executável a partir do sistema desenvolvido. Para esta situação, foi escolhido o “Generic Real-Time Target” no “Real-Time Workshop”, mais uma vez do “Configuration Parameters” do SIMULINK, Figura 68.

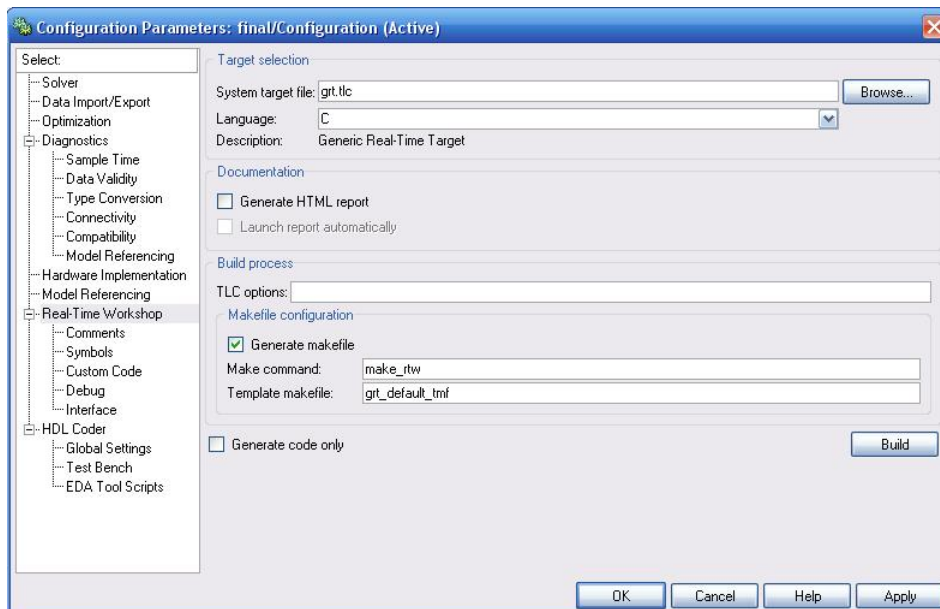


Figura 68: Configuração adoptada para a geração do executável, “Generic Real-Time Target”

Depois, com o “Build” do mesmo indicador (Figura 68), procede-se à construção do ficheiro executável que conterà a resolução em tempo real do sistema desenvolvido em ambiente SIMULINK.

- Real-Time Windows Target

Para realizar a interacção entre o executável e o modelo desenvolvido em SIMULINK, é necessário definir no “*Hardware Implementation*” do “*Configuration Parameters*” as seguinte opções, Figura 69.

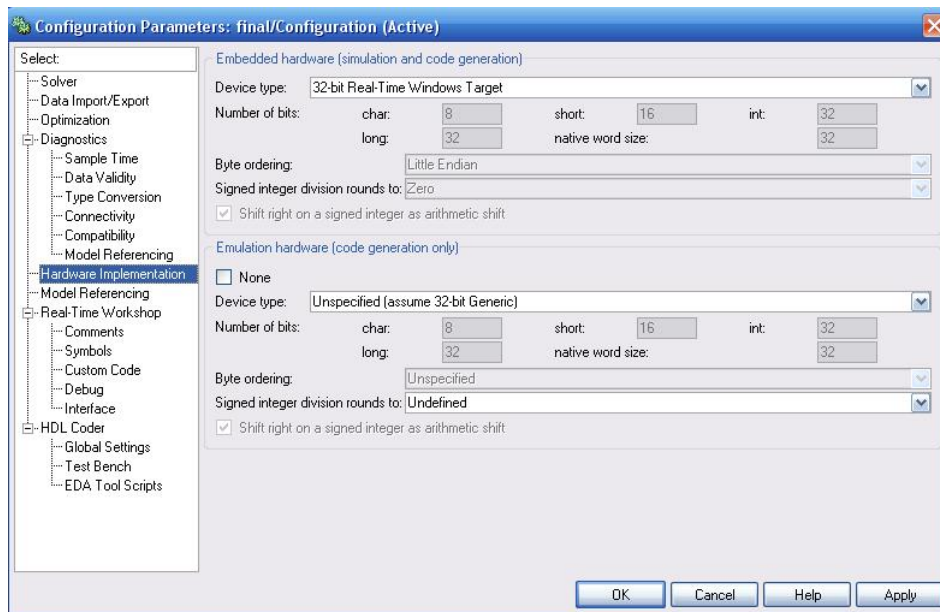


Figura 69: “*Hardware Implementation*” do “*Configuration Parameters*”

e seleccionar desta vez “*Real-Time Windows Target*” no “*Real-Time Workshop*”, Figura 70.

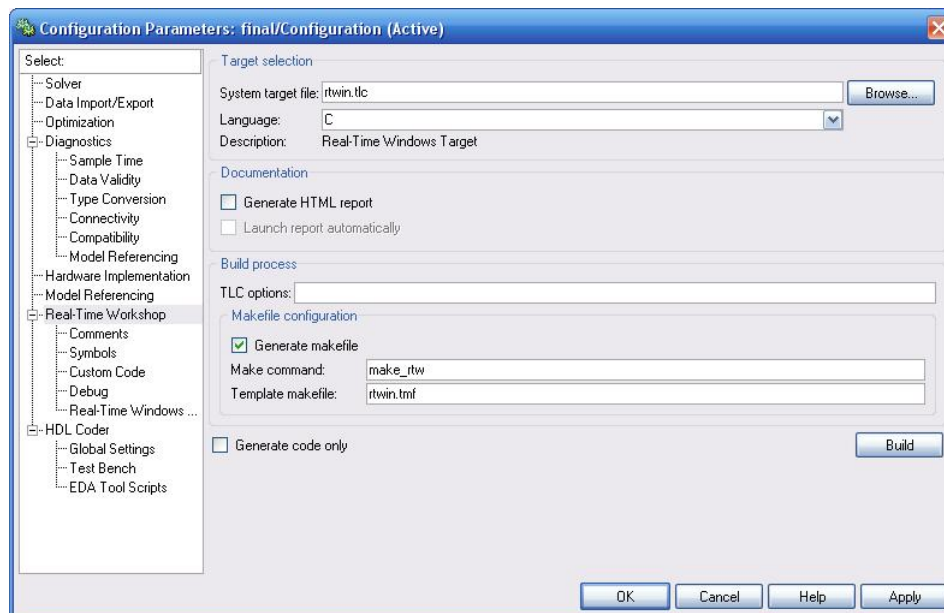


Figura 70: Configuração adoptada para a geração do executável, “*Real-Time Windows Target*”

Por fim, após a compilação, para concluir a interacção do “*Real-Time Windows Target*” com o SIMULINK, é necessário correr o modelo em “*External mode*” através do menu “*Simulation*”, e estabelecer a ligação do mesmo com o modelo criado em tempo real através do “*Connect to Target*” do mesmo menu.

- **FLIGHTGEAR**

O FLIGHTGEAR é um simulador de voo *OpenSource* desenvolvido por elementos independentes que tem como principal objectivo desenvolver uma plataforma acessível a todos os interessados através de serviços gratuitos sem violação dos direitos de programação. Devido ao facto de ser *OpenSource*, o FLIGHTGEAR possibilita que cada utilizador desenvolva e altere as configurações iniciais ou até mesmo porções de código para o simulador reagir conforme as suas necessidades.

- **Incorporação do ALIV no FLIGHTGEAR**

O FLIGHTGEAR tem a capacidade de conseguir descodificar inúmeros ficheiros de modelação 3D, principalmente todos aqueles característicos de programas de modelação 3D *OpenSource*. O SOLIDWORKS não entra nessa categoria, como tal, teve de ser utilizado um “*add-in*” do SOLIDWORKS propicio para este tipo de transformações, o *Dxf Converter for SOLIDWORKS*.

Com isto, após a construção e gravação do modelo no formato apropriado, foi preciso definir o FLIGHTGEAR para visualiza-lo durante a simulação. Para isso, fez-se uso do modelo do FLIGHTGEAR ideal para situações deste género, o UFO. Assim, para que o modelo 3D do ALIV seja o adoptado durante a simulação, basta iniciar o FLIGHTGEAR com o UFO e associar o ficheiro de configuração inicial do mesmo, *ufo.xml*, ao novo modelo pretendido.

Após a inclusão do modelo, é necessário corrigir a sua posição e orientação através de “*offsets*” adicionados ao mesmo ficheiro de configuração inicial do UFO. A Figura 71 ilustra as alterações efectuadas ao ficheiro *ufo.xml* de modo a que o FLIGHTGEAR abra e alinhe correctamente o modelo 3D desenvolvido do ALIV sempre que se iniciar o simulador.

```

<?xml version="1.0" ?>
- <PropertyList>
  <path>aliv.ac</path>
- <offsets>
  <x-m>0</x-m>
  <y-m>1.1</y-m>
  <z-m>1.85</z-m>
  <roll-deg>90</roll-deg>
  <heading-deg>0</heading-deg>
</offsets>

```

Figura 71: “*offsets*” do ficheiro *ufo.xml*

Tipicamente, o FLIGHTGEAR trabalha com ficheiros de modelação 3D *AC3D*, por isso foi utilizado ainda o programa de modelação 3D *OpenSource* BLENDER [22] para fazer a conversão de *DXF* para *AC3D*. Desta forma, foi possível definir no modelo desenvolvido do ALIV diferentes objectos, entre os quais, cada um dos rotores, de forma a que os mesmos pudessem ser animados consoante a velocidade angular atribuída. Este objectivo é conseguido fazendo mais uma vez uso do ficheiro *ufo.xml* e das definições pré-estabelecidas no FLIGHTGEAR. A Figura 72 apresenta como exemplo as alterações necessárias ao ficheiro *ufo.xml* para a obtenção da rotação da hélice número um. Como pode ser constatado na mesma figura, o movimento da hélice é dado em rotações por minuto, e está associado ao motor equivalente.

```

- <animation>
  <type>spin</type>
  <object-name>helice1</object-name>
  <property>/velocities/airspeed-kt</property>
  <factor>-1</factor>
- <center>
  <x-m>-2.4</x-m>
  <y-m>0</y-m>
  <z-m>-4.16</z-m>
</center>
- <axis>
  <x>0</x>
  <y>1</y>
  <z>0</z>
</axis>
</animation>

```

Figura 72: Animação da primeira hélice no FLIGHTGEAR através do ficheiro *ufo.xml*

A partir deste ponto, é necessário que este modelo assuma e reaja de acordo com a dinâmica desenvolvida em ambiente SIMULINK.

- Interface entre o SIMULINK e o FLIGHTGEAR, *Aerospace Toolbox*

A *Aerospace Toolbox* é formada por inúmeros blocos que permitem ao utilizador desenvolver e aperfeiçoar os seus modelos aeronáuticos, fazendo uso desses blocos nos mais variadíssimos campos, como a aerodinâmica, a propulsão e até mesmo na construção de dinâmicas completas de aeronaves. No entanto, como já foi referido, para o caso da presente tese, a importância desta *Toolbox* está associada a outra virtude, a possibilidade da visualização 3D do movimento da aeronave através do simulador de voo FLIGHTGEAR. Para fazer uso desta propriedade, é necessário adicionar os blocos provenientes da *Aerospace Toolbox* para que se possa realizar a interface entre a dinâmica desenvolvida em SIMULINK e o simulador, Figura 73.

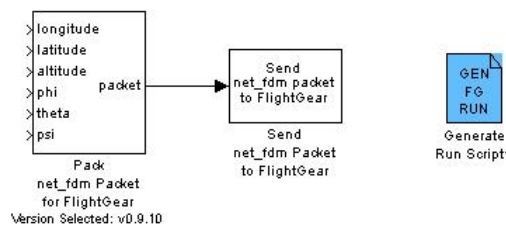


Figura 73: Blocos para interface entre o Simulink e o FLIGHTGEAR

O primeiro bloco é o *Pack net\_fdm Packet for FLIGHTGEAR*, que cria através das entradas da posição e da orientação um pacote compatível com uma determinada versão do FLIGHTGEAR. O segundo bloco é o *Send net\_fdm Packet to FLIGHTGEAR*, que transmite o pacote elaborado anteriormente para um FLIGHTGEAR que esteja a correr no mesmo computador ou num computador em rede. Ambos os blocos são definidos para serem transmitidos à mesma frequência de simulação. O terceiro bloco é o *Generate Run Script*, e é responsável por criar os parâmetros de iniciação para o FLIGHTGEAR. Aqui podem ser definidos entre outros, o tipo de aeronave utilizada (no caso da presente tese o ALIV) e a posição de partida.

Embora sejam estes os blocos principais que possibilitam a geração da imagem no FLIGHTGEAR através do SIMULINK, é necessário proceder algumas adaptações extra para que se mantenha uma

continuidade entre o início dos cálculos e a posição inicial adoptada no FLIGHTGEAR. Nomeadamente à conversão de coordenadas *NED* (adoptadas na simulação) para *LLA* (adoptadas pelo FLIGHTGEAR), através do bloco *Flat Earth to LLA*. Para isso, é necessário indicar a posição e orientação inicial do cálculo de acordo com a posição inicial escolhida no bloco *Generate Run Script*, Figura 74.

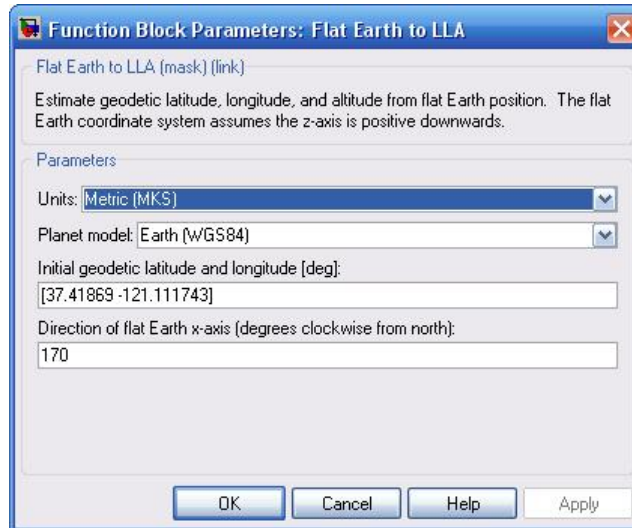


Figura 74: *Flat Earth to LLA* utilizado

Seguindo o mesmo raciocínio, e atribuindo a altitude inicial do cálculo fica totalmente configurado o bloco que envia os dados da dinâmica do ALIV para o FLIGHTGEAR sem que este “salte” da sua configuração inicial para a posição de cálculo, Figura 75.

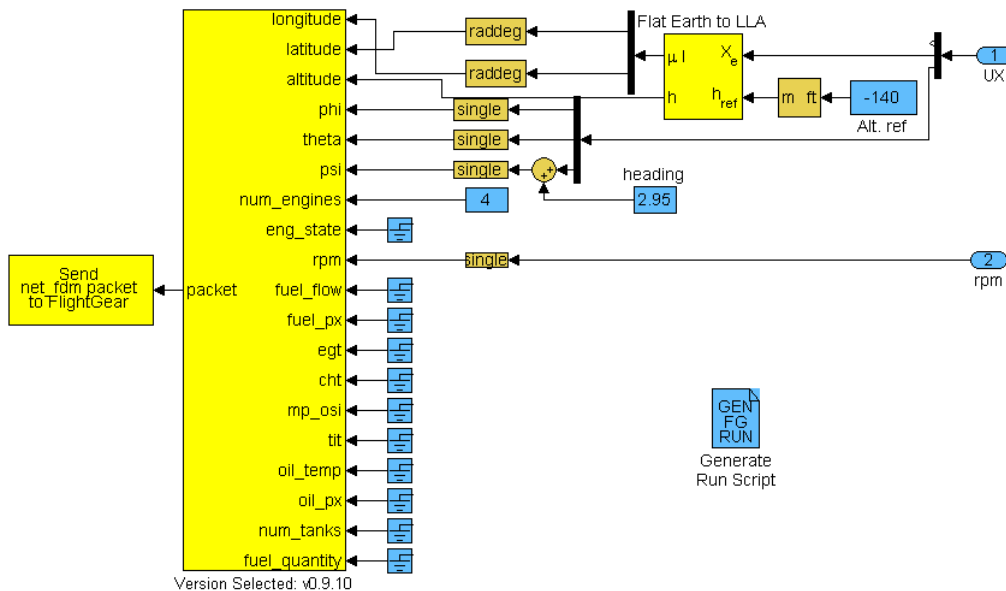


Figura 75: Modelo completo utilizado na interface entre o Simulink e o FLIGHTGEAR

Na imagem anterior é possível observar como foi indicado anteriormente, a passagem dos valores referentes à rotação dos motores para o FLIGHTGEAR, de modo a que depois, estes possam ser associados ao modelo 3D das hélices durante a simulação.