



INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa



PASSAROLA – Dirigível Autónomo para Operações de Salvamento

(Parte Electrónica)

David Lopes Jerónimo

Dissertação para a obtenção de grau de mestre em
Engenharia Electrotécnica e de Computadores

Júri

Presidente: José António Beltran Gerald

Orientador: Prof. Francisco Alegria

Co-Orientador: Prof. Pedro Lima

Vogais: Gonçalo Nuno Gomes Tavares

Outubro 2008

Agradecimentos

Quero agradecer aos Professores Francisco Alegria e Pedro Lima pela orientação e disponibilidade que me proporcionaram no desenvolvimento deste meu trabalho, dissertação de mestrado, bem como ao meu colega Ricardo Alcácer, com quem desenvolvi o trabalho complementarmente mas individualmente em âmbitos distintos, o meu trabalho na componente de electrónica e o trabalho do meu colega na componente de robótica.

Alem do Ricardo, também os colegas Cristina e Manuel foram muito importantes na constituição de um grupo de trabalho muito dedicado e solidário, ao longo dos cinco anos do curso, que nos proporcionou um excelente trabalho de equipa na nossa formação profissional como engenheiros e a maturação das nossas personalidades num curso que escolhemos e para o qual estávamos bem vocacionados e que realizámos com grande entusiasmo.

Também a todos os Professores e colegas, agradeço porque a sua ajuda me proporcionou o estímulo que necessitava para levar a cabo este meu trabalho.

Resumo

O projecto PASSAROLA é a evolução de um trabalho iniciado no Instituto Superior Técnico integrado no Projecto RESCUE, Navegação Cooperativa para Robots de Salvamento. Este projecto visa o desenvolvimento de soluções robóticas para busca e salvamento em situações de catástrofe.

O objectivo deste projecto é a concepção, desenvolvimento e implementação de todo o sistema electrónico que proporcione ao dirigível uma completa automação no seguimento de linhas ou de um robot terrestre – o robot ATRV Jr – em ambiente real.

Para tal, elaboramos um simulador realista que permite testar todo o algoritmo implementado no dirigível, permitindo a sua visualização e conseqüente calibração dos seus comportamentos, minimizando os ajustes necessários na transição do conjunto para ambiente real.

Deste modo, ao dirigível existente foi acoplada uma placa de processamento – DSP – que recebe imagem enviada por uma câmara *onboard* e calcula todo o controlo necessário para comandar os motores durante o decorrer do seu percurso.

Palavras-Chave

Dirigível PASSAROLA

Controladores

PWM

DSP

Processamento de imagem

Seguimento de Linhas e ATRV Jr

VisualDSP++

UsarSim

Abstract

The PASSAROLA project is a continuation of work begun in the IST, and is integrated in the RESCUE Project, Cooperative Navigation for Rescue Robots.

The project aims to develop robotic means for search and rescue in emergency/catastrophe situations.

Thus the objective of the work has concentrated on the design, development and implementation of the electronic system which gives the Blimp complete automation following lines marked on the ground, or controlled by a land based robot - the ATRV Robot Junior, in a real environment situation.

For this, a realistic simulator was assembled that allows the testing of all the algorithms installed in the Blimp, so they can be visualized and their behaviour consequently calibrated to minimize adjustments between the laboratory experiments and the real environment.

In this way, a DSP Processing Board was coupled to the existing Blimp to receive images sent by an on board camera and to calculate all the control required to command the motors during the mission.

Key words

PASSAROLA Blimp

Controllers

PWM

DSP

Image Processing

Line and ATRV Jr Following

VisualDSP++

UsarSim

Índice

Resumo.....	1
Abstract.....	2
Índice.....	3
Índice de Ilustrações.....	5
Índice de Tabelas.....	7
1 INTRODUÇÃO.....	9
2 HARDWARE DO DIRIGÍVEL.....	13
2.1 Descrição e Arquitectura do Dirigível.....	13
2.2 Envelope.....	16
2.3 Cockpit.....	17
2.4 Câmara de Vídeo.....	18
2.5 Motores.....	20
2.6 Controladores.....	24
2.7 Baterias.....	25
2.8 Placa de processamento – DSP.....	26
2.9 Posicionamento por GPS.....	29
2.9.1 Descrição do GPS.....	29
2.9.2 Sistema GPS preparado.....	33
3 SOFTWARE DO DIRIGÍVEL.....	37
3.1 Arquitectura do Software.....	37
3.2 Algoritmos de Processamento de Imagem.....	39
3.2.1 Seguimento de Linhas.....	39
3.2.2 Seguimento do Robot ATRV Jr.....	43
3.3 Geração do Sinal de Áudio para Controlo dos Motores.....	46
3.4 Programação do DSP.....	47
3.4.1 VisualDSP++.....	50
3.4.2 Seguimento de Linhas.....	51
3.4.3 Seguimento ATRV Jr.....	56
3.4.4 Sincronia dos Dois Programas.....	62
4 SISTEMA DE ENSAIO DIRIGÍVEL.....	63
4.1 Constituição.....	63
4.2 Simulador UsarSim.....	68

4.3	Placa de aquisição	69
4.4	Aplicação em Labview	69
4.5	Calibrações, Testes e Experiências.....	71
5	RESULTADOS.....	75
5.1	Resultados em Ambiente de Simulação	75
5.2	Resultados em Ambiente Real	80
6	CONCLUSÕES	83
	Referências Bibliográficas	85
	Referências da Internet	86

Índice de Ilustrações

Ilustração 1 – Robot ATRV Jr	10
Ilustração 2 – Robot PASSAROLA.....	10
Ilustração 3 – Imagem de um dirigível	13
Ilustração 4 – Ilustração do PASSAROLA	14
Ilustração 5 – Fotografia do primeiro dirigível	14
Ilustração 6 – Esquema do sistema original.....	15
Ilustração 7 – Posição das asas traseiras do dirigível	16
Ilustração 8 – <i>Cockpit</i> original.....	17
Ilustração 9 – Novo <i>cockpit</i>	17
Ilustração 10 – Dois ângulos diferentes do dirigível	17
Ilustração 11 – Electrónica do cockpit dianteiro	18
Ilustração 12 – Electrónica do cockpit traseiro.....	18
Ilustração 13 – Imagem da câmara 1	19
Ilustração 14 – Imagem da câmara 2	19
Ilustração 15 – Motores dianteiros e haste de suporte	20
Ilustração 16 – Motor traseiro	20
Ilustração 17 – Teste dos motores utilizando o gerador de funções (em baixo) e visualizando	22
Ilustração 18 – Gráfico do funcionamento do motor	22
Ilustração 19 – Controlador dos motores dianteiros	25
Ilustração 20 – Controlador do motor traseiro	25
Ilustração 21 – Imagem do DSP	28
Ilustração 22 – Esquemático da localização dos conectores, LED's e <i>switches</i>	29
Ilustração 23 – Distribuição das orbitas e satélites pela Terra	30
Ilustração 24 – Esquema das pseudo-distâncias entre receptor e satélites.....	31
Ilustração 25 – Diversos formatos de coordenadas.....	31
Ilustração 26 – Exemplo das reflexões dos sinais transmitidos pelos satélites	32
Ilustração 27 – Esquema do Sistema de GPS Diferencial	32
Ilustração 28 – Antena de recepção GPS.....	33
Ilustração 29 – Módulo de recepção GPS	33
Ilustração 30 – Mapeamento do GPS num trajecto.....	34
Ilustração 31 – Esquemático de alimentação dos dois componentes do sistema GPS	35
Ilustração 32 – Sistema GPS construído	35
Ilustração 33 – Referencial do dirigível.....	38
Ilustração 34 – Eixos (x, y) considerados na imagem.....	39
Ilustração 35 – Esquemático do Algoritmo <i>Sobel</i>	39
Ilustração 36 – Imagem original	40
Ilustração 37 – Imagem resultante do Algoritmo <i>Sobel</i>	40
Ilustração 38 – Imagem após aplicação do Algoritmo <i>Sobel</i> e <i>threshold</i> elevado	40
Ilustração 39 – Exemplo de 3 pontos de uma recta e 6 possíveis rectas que os intersectam.....	41
Ilustração 40 – Gráfico distância vs ângulo resultante dos 3 pontos	41

Ilustração 41 – Resultado do Algoritmo de <i>Hough</i> (à direita) de uma imagem simples (à esquerda).....	42
Ilustração 42 – Resultado do Algoritmo de <i>Hough</i> de uma imagem.....	42
Ilustração 43 – À esquerda encontra-se uma imagem unicamente composta pela componente V da imagem da direita	44
Ilustração 44 – <i>Output</i> do programa MatLab (Em cima - imagem original; à esquerda: imagem original sem os pixels considerados vermelhos pelos <i>thresholds</i> ; à direita: imagem dos pixels considerados vermelhos)	45
Ilustração 45 – Organização dos pixels no formato UYVY.....	45
Ilustração 46 – Quatro períodos dos sinais de controlo enviados para os motores.....	48
Ilustração 47 – Exemplo do método de varrimento entrelaçado.....	49
Ilustração 48 – Exemplo de uma imagem em movimento	49
Ilustração 49 – Imagem do VisualDSP++, versão 4.5	50
Ilustração 50 – Exemplo da segmentação da imagem recebida	51
Ilustração 51 – Imagem recebida do simulador.....	54
Ilustração 52 – Imagem pós-processada com o vector de indicação de direcção.....	54
Ilustração 53 – Esquemático da organização dos algoritmos implementados em cada <i>core</i>	55
Ilustração 54 – Exemplo do centro de massa de dois objectos agregados.....	56
Ilustração 55 – Exemplo do c.m. de um objecto existindo mais que um objecto na imagem	57
Ilustração 56 – Exemplo do aumento da zona de pesquisa de vermelhos	57
Ilustração 57 – Eixo e vector do ângulo de erro	59
Ilustração 58 – Eixos das coordenadas da imagem e respectivos vectores de erro (em x e y)	60
Ilustração 59 – <i>Output</i> do DSP enviado para a base em terra.....	61
Ilustração 60 – Esquemático da organização dos algoritmos implementados em cada <i>core</i>	61
Ilustração 61 – Esquemático das ligações do ambiente de simulação	64
Ilustração 62 – Ligações de vídeo e dos controladores do DSP	65
Ilustração 63 – Bancada de trabalho em laboratório	66
Ilustração 64 – Esquemático das ligações do ambiente real	67
Ilustração 65 – Esquemático das ligações do dirigível	67
Ilustração 66 – Imagens do dirigível simulado	68
Ilustração 67 – Placa de aquisição da <i>National Instruments</i>	69
Ilustração 68 – Interface gráfico do <i>software</i> LabView	70
Ilustração 69 – Velocidade linear do dirigível em X vs <i>dutycycle</i> dos motores dianteiros.....	72
Ilustração 70 – Velocidade angular do dirigível vs <i>dutycycle</i> do motor traseiro.....	72
Ilustração 71 – Trajectória em forma de U, realizada com $V = 0,58\text{m/s}$ e a uma altitude de 6m.....	76
Ilustração 72 – Trajectória de inversão de marcha, realizada com $V = 0,58\text{m/s}$ e uma altitude de 6m.....	77
Ilustração 73 – Trajectória fechada realizada com $V = 0,78\text{m/s}$ e a uma altitude de 6m	78
Ilustração 74 – Trajectória fechada realizada com $V_{\text{med}} \approx 0,73\text{m/s}$ e a uma altitude de 6m.....	79
Ilustração 75 – Sequencia de imagens no decorrer de um percurso rectangular.....	81
Ilustração 76 – Sequencia de imagens no decorrer de uma curva.....	82

Índice de Tabelas

Tabela 1 – <i>Dutycycle</i> do motor traseiro	23
Tabela 2 – <i>Dutycycle</i> dos motores dianteiros	23
Tabela 3 – <i>Dutycycle</i> do motor de rotação da haste dos motores dianteiros	23
Tabela 4 – Variáveis de processamento dos blocos	51
Tabela 5 – Comandos enviados para o simulador	70
Tabela 6 – Resultados dos testes de velocidade	71
Tabela 7 – Resultados dos testes de rotação	71
Tabela 8 – Equações de segundo grau do funcionamento dos motores dianteiros	73
Tabela 9 – Equações de segundo grau do funcionamento dos motores dianteiros	73

1 | INTRODUÇÃO

A Robótica é uma área da Engenharia onde a investigação desperta cada vez mais o interesse de muitas empresas, com vista à optimização de processos. Nos dias de hoje, a automatização de processos é algo quase indispensável em qualquer indústria, onde a intervenção do homem tende a ser cada vez mais reduzida, optimizando processos, reduzindo custos, acelerando e melhorando os serviços, a produção e os resultados desses processos. É este o desafio da Robótica, construir máquinas capazes de interagir entre si e com as pessoas e com isso conseguirem, autonomamente, gerar soluções para problemas e cooperarem entre si, de modo a facilitarem as suas tarefas, reduzindo custos e tempos.

Quando se trata de robótica aplicada à navegação aérea, a situação ainda é mais incipiente. A investigação desta vertente da robótica não se encontra tão desenvolvida, mas promete grande expansão em breve.

Como robótica aérea encontram-se os robots avião, os robots helicóptero e os robots balão/dirigível, onde a principal característica diferenciadora reside na sua constituição e forma de voo e por isso requerem tipos de controlo diferentes. Cada um destes tipos de robots, tendo em comum a necessidade de sustentação no ar, requerem sistemas de controlo diferenciados, no caso dos aviões a necessidade de se atingir uma determinada velocidade para que o efeito aerodinâmico garanta a manutenção do voo. Já nos helicópteros há que conseguir a estabilização num determinado ponto do espaço, i.e. velocidade de deslocamento nula, efeito conseguido pela disposição dos seus motores e hélices. A sustentabilidade dos balões e dirigíveis é conseguida através do composto gasoso que é utilizado para o enchimento dos mesmos mas são os motores que lhe permitem a deslocação horizontal ou vertical.

No Instituto Superior Técnico existe o projecto de busca e salvamento conhecido por **Projecto RESCUE, Navegação Cooperativa para Robots de Salvamento**, desenvolvido no Instituto de Sistemas e Robótica.

Este projecto visa a implementação de um sistema de salvamento autónomo, baseado em robots, que possibilite o socorro a vítimas e o mapeamento do terreno em situações de catástrofe e de difícil acessibilidade. A equipa de robots é composta por um robot terrestre, ATRV Jr (Ilustração 1), e um robot aéreo, um dirigível conhecido por PASSAROLA (Ilustração 2).



Ilustração 1 – Robot ATRV Jr



Ilustração 2 – Robot PASSAROLA

Este projecto pretende integrar um conjunto de agentes autónomos a trabalhar em formação capazes de interagir e cooperar entre si num cenário de catástrofe, terramoto, maremoto ou desastre nuclear, onde as condições são demasiado adversas ou de difícil intervenção humana, e é necessário uma intervenção rápida das equipas de salvamento, de forma a prevenir ou minimizar a ocorrência de vítimas.

Neste projecto, o dirigível terá a missão de efectuar o reconhecimento do terreno, enquanto uma equipa de robots móveis se desloca no solo, mantendo permanente contacto com o dirigível, obtendo informações sobre o terreno e outras, e servindo como transmissor de informações entre robots terrestres e a estação base.

Esta teoria de controlo de formações pode ser alargada para outros sistemas como por exemplo nas missões espaciais, incluindo a exploração de novos planetas. Também à que considerar a vertente militar.

O presente trabalho foi realizado no âmbito desse projecto e dedicou-se totalmente à automação do dirigível PASSAROLA, concebendo, desenvolvendo e implementando todo o sistema electrónico para o seu controlo, de forma a alcançar como objectivo o seguimento de uma linha ou de um robot terrestre em movimento (o robot ATRV Jr).

Tendo este trabalho duas componentes interligadas, a componente electrónica e a componente robótica, foi a primeira desenvolvida por mim e a segunda pelo colega Ricardo Alcácer, em [5], num trabalho, em complementaridade e interligação, com o objectivo comum do funcionamento global do sistema.

Para tal, foi necessário redesenhar o circuito electrónico do dirigível de forma a incluir um computador de bordo responsável pela aquisição de imagem, seu processamento, implementação do algoritmo de controlo do dirigível e condução do mesmo por actuação dos motores, bem como outros componentes necessários para alcançar os fins pretendidos.

Como ponto de partida para este projecto realizou-se uma pesquisa sobre documentos existentes que se relacionassem de alguma forma com o nosso projecto, com o intuito de encontrar conhecimentos e estudos relevantes, de fim de se consolidarem os conhecimentos sobre o projecto, os dirigíveis em geral e o PASSAROLA em particular.

Descobriu-se em [1] que a cinemática e dinâmica do PASSAROLA já tinha sido resolvida, bem como algum estudo e simulação sobre controlo de seguimento de veículos terrestres assumidos como líderes (*Robot ATRV Jr*). Contactado o autor de [1] conseguiu-se o acesso ao simulador, em MatLab, para o dirigível, escolhendo-se este simulador como base de desenvolvimento de trabalho futuro na área do controlador dos motores.

Verificou-se também em [2] que foi realizado algum estudo sobre o PASSAROLA, nomeadamente problemas de controlo devido ao posicionamento dos motores e sua potência, estudo sobre a deslocação óptica em imagens retiradas a partir do dirigível e controlo de navegação onde foram desenvolvidas funções em linguagem C e sua aplicação directa no dirigível, utilizando processamento por computador e comunicação com o *robot* através de um comando de RF.

Embora neste caso o controlo já desenvolvido para o PASSAROLA não seja muito útil para o desenvolvimento do projecto, uma vez que se pretende construir um controlador mais geral, a funcionar num DSP acoplado ao dirigível com comunicação directa com os motores e sensores, tornando o PASSAROLA completamente autónomo; o estudo descrito em [2] enriquece-nos com vivências e problemas para os quais devemos estar preparados durante o trabalho, bem como guia de orientação para a implementação do controlador no sistema real.

Sabendo a princípio que o PASSAROLA terá como sensor uma câmara de vídeo com o objectivo de seguir trajectórias ou objectos presentes no solo e futuramente para aquisição de mapas topográficos, no sentido de fazer o reconhecimento de uma área só acessível por meios aéreos, é também importante o estudo de [3] onde é focado com mais pormenor a dinâmica das imagens numa sequência de vídeo retirada a partir de um dirigível, bem como a calibração da câmara de vídeo e seguimento de objectos numa imagem, sendo neste caso, útil o conhecimento já adquirido na cadeira de Processamento de Imagens e Visão (PIV), onde foi realizado um projecto de seguimento de pequenas imagens (*templates*), conhecidas *a priori*, ao longo de uma sequência de vídeo.

Seguidamente fez-se uma pesquisa exaustiva sobre todos os componentes existentes no dirigível, tais como, controladores, motores, hélices, modulo RF e transmissor de vídeo RF, e de todos os componentes necessários adquirir para completar o protótipo, como a placa de processamento, a câmara *onboard*, baterias e um novo *cockpit*.

Passou-se à fase de teste desses componentes e à selecção dos a adquirir, à sua aquisição e à montagem ao dirigível.

O dirigível existente foi remontado e testado de modo a confirmar toda a sua funcionalidade. Para tal, foi necessário estudar pormenorizadamente todos os componentes, de forma a possibilitar a reconstrução do sistema electrónico do dirigível.

Após esse trabalho todos os componentes do dirigível ficaram em perfeito funcionamento.

No Capítulo 2 encontram-se descritos em pormenor as características e respectivos testes de cada componente de *hardware* do dirigível, inclusive as informações sobre a placa de processamento escolhida para comandar todo o dirigível.

Toda a documentação obtida sobre os componentes do dirigível encontra-se anexada ao relatório através de um CD.

Sobre a componente de *software* desenvolvido para este projecto iniciou-se com a elaboração de um simulador próprio para o dirigível no ambiente Usarsim, explicado no Subcapítulo 4.2, que será uma ferramenta de trabalho de importantíssima utilidade durante o desenvolvimento dos algoritmos de seguimento – Subcapítulo 3.4 –, porque permite verificar a correcta funcionalidade do sistema sem a necessidade de utilizar o dirigível real, facilitando e minimizando tempo no processo de *debug*.

Por fim implementaram-se os algoritmos de controlo, descritos em mais pormenor em [5], e calibrou-se todo o sistema – principalmente o simulador e os controladores – recorrendo a testes no simulador e depois com o dirigível real – Subcapítulo 4.5.

A última fase do projecto consiste no teste de todo o sistema em funcionamento, tanto em simulação como em realidade – Capítulo 5.

2 | HARDWARE DO DIRIGÍVEL

2.1 Descrição e Arquitectura do Dirigível

O que é um dirigível? Um dirigível é um veículo aéreo, composto por um gigantesco balão, uma pequena gôndola e propulsores que o tornam capaz de controlar o seu trajecto – propulsão e orientação. Um exemplo de dirigível é o demonstrado na Ilustração 3.



Ilustração 3 – Imagem de um dirigível

O balão deste veículo, é cheio de um gás pouco denso, como o hélio ou o hidrogénio, mais leve que o ar, oferecendo ao veículo a capacidade de pairar no ar. Esta sua característica torna-o num veículo muito seguro, estável, de baixo custo e pouco poluente, em comparação com outros veículos aéreos. Outra vantagem destes veículos é o baixo nível de ruído, tornando-o um atractivo para a sua utilização em ambientes habitados.

Devido ao facto de ser necessária pouca energia para sustentar o dirigível no ar, a autonomia deste veículo é bastante elevada.

Claro que, como em todos os sistemas, não existem só vantagens. Uma das limitações dos dirigíveis prende-se com o facto do seu elevado volume oferecer grande resistência ao ar não permitindo velocidades elevadas de deslocação, e também ser bastante sensível às variações das condições ambientais, nomeadamente ventos.

Estes veículos são normalmente caracterizados pela forma elipsoidal do envelope, o qual é mantido rígido por uma sobrepressão no interior, conferindo rigidez e aerodinâmica à aeronave.

A título de curiosidade, o nome do dirigível do projecto – PASSAROLA – advém da primeira aeronave conhecida no mundo, um balão de ar quente, o primeiro que se elevou no ar, inventada e construída

pelo padre jesuíta português Bartolomeu de Gusmão, da Universidade de Coimbra, em Portugal, e financiado pelo Rei D. João V. Era uma pessoa com grande capacidade imaginativa e matemática, conseguindo comprovar as suas teorias desenvolvidas acerca da maior leveza do ar quente.

A 8 de Agosto de 1709 Bartolomeu fez voar um pequeno balão de ar quente na Sala dos Embaixadores na Casa da Índia pela primeira vez no mundo, perante uma assistência importante, onde se incluía a família Real, além de elementos do Corpo Diplomático e da Corte Portuguesa.

Mais tarde, Bartolomeu de Gusmão fez a primeira viagem aérea tripulada desde o Castelo de São Jorge ao Terreiro do Paço (cerca de 1Km). Não foram muitas as pessoas que assistiram ao acontecimento mas um cronista da época relatou o feito e ilustrou a sua notícia com um desenho, produto da sua fantasia e imaginação, que designou por "PASSAROLA", desenho este que ainda hoje pode ser visto no Museu do Ar da Força Aérea Portuguesa (Ilustração 4).



Ilustração 4 – Ilustração do PASSAROLA

Pouco mais se sabe acerca deste aparelho pois na época o desconhecimento na área de ciências aeronáuticas era grande e todos os projectos e documentos originais do inventor se perderam com o passar do tempo.

Como curiosidade, a proveniência do nome *Zeppelin* que também é aplicado aos dirigíveis. Este nome advém em memória do Lord Zeppelin, o pioneiro na construção de dirigíveis capaz de se moverem sem o auxílio do vento, pouco antes da primeira guerra mundial. Este dirigível tinha o formato de um charuto e na sua gôndola, existia um conjunto de hélices que eram responsáveis pela manobrabilidade do veículo (Ilustração 5).

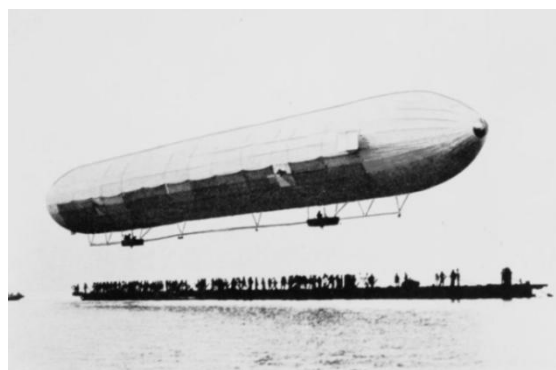


Ilustração 5 – Fotografia do primeiro dirigível

Iremos utilizar como base de desenvolvimento do projecto um dirigível já existente no IST, propriedade do Instituto de Sistemas e Robótica (ISR) e que foi construído no Trabalho Final de Curso da Licenciatura de Engenharia Electrotécnica e de Computadores pelo aluno Thomas Krause. O seu trabalho consistiu no desenvolvimento do estudo sobre a dinâmica do dirigível e controlo do mesmo através de rádio.

O dirigível era comandado através de um telecomando RF em modo manual controlado por um utilizador, ou em modo automático em que um computador enviava comandos através do telecomando RF, utilizando uma ligação RS-232.

Existia a bordo uma câmara que enviava imagens para a *ground station*, ou estação base, que poderia servir para fechar o *loop* de controlo do veículo (Ilustração 6).

O sistema de vídeo – câmara mais sistema de direcção da mesma – foi desmontado em virtude do sistema que existia não cumprir as especificações que pretendemos.

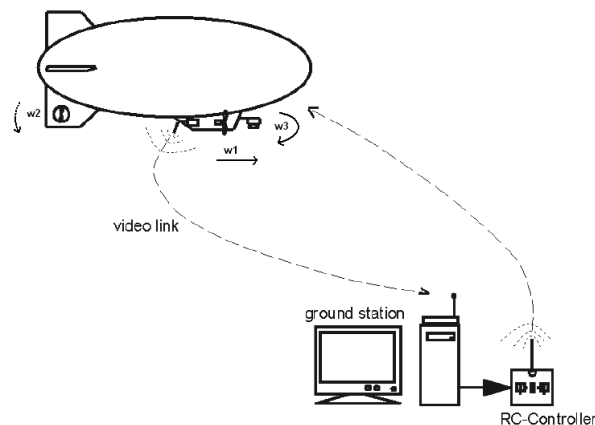


Ilustração 6 – Esquema do sistema original

Este sistema de controlo utilizado no âmbito de [2], foi descontinuado uma vez que se pretende colocar todo o processamento de sensores e controlo do veículo a funcionar harmoniosamente a bordo do dirigível.

O dirigível utilizado para o projecto é composto por diversos componentes como: o envelope, o *cockpit*, os motores de tracção, a electrónica de controlo – controladores dos motores, receptor RF do sinal do comando –, a câmara e respectivo emissor de sinal e as baterias de alimentação dos componentes.

2.2 Envelope

O envelope do dirigível utilizado é feito com material poliuretano, com uma espessura de 3,5mil, i.e., 89 μ m, e tem como dimensões, 4 metros de comprimento e 2 metros de diâmetro maior.

O poliuretano demonstra uma boa flexibilidade e resistência, o que permite aguentar uma maior pressão do ar no seu interior, por outro lado demonstrou ter uma permeabilidade alta significando que se esvazia relativamente rápido, não se mantendo totalmente cheio durante um dia de trabalho e por isso modificando a sua sustentabilidade ao longo do tempo.

Para equilibrar o esvaziamento, é necessário fazer pequenos ajustes no contrapeso utilizado, constituído por pequenos chumbos.

A forma do envelope é a de uma gota de água deitada e tendo na cauda acoplados quatro estabilizadores, dois na vertical e dois na horizontal, que servem para estabilizar o movimento de deslocação do dirigível (Ilustração 7).



Ilustração 7 – Posição das asas traseiras do dirigível

Relativamente ao gás utilizado para encher o balão é o composto químico do elemento hélio (He), escolha que já provinha do trabalho anterior do colega Thomas Krause. Esta escolha deveu-se ao material de que é composto o envelope e ao facto de ser menos perigoso que a outra hipótese de gás a usar, o hidrogénio, que tem como grande desvantagem ser inflamável tornando assim perigoso o seu manuseamento. O hélio por sua vez é um pouco mais caro mas de mais fácil manuseamento e aquisição.

Devido ao preço elevado do hélio puro as empresas que fornecem este tipo de gás misturam-no com outros compostos gasosos mais densos, como o azoto e o hidrogénio, numa proporção de 30%, resultando o gás conhecido por hélio Helibal.

A empresa fornecedora de botijas de hélio utilizada foi a Air Liquide, sediada em Lisboa, Portugal.

Considerámos que por cada metro cúbico de hélio se consegue um poder de elevação de aproximadamente 1Kg em condições normais. Assim, segundo [4], tendo o envelope um volume de 5,5m³, o dirigível terá aproximadamente 5,5Kg de sustentação.

2.3 Cockpit

O *cockpit* apresenta um formato aerodinâmico, construído numa resistente fibra de carbono (PET *Plastic*), oferecendo leveza e ao mesmo tempo robustez (Ilustração 8).

Além do *cockpit* já existente foi necessário proceder à montagem de outro que adquirimos, pois o que existia não tinha as dimensões necessárias para suportar a nova electrónica inserida como por exemplo a placa de processamento (DSP). Este novo *cockpit* foi construído em fibra de vidro transparente, permitindo assim a visualização dos *leds* da placa de processamento para possíveis fins de *debug*. Este material foi escolhido por ser muito resistente e ao mesmo tempo flexível possibilitando fazer diversas aberturas de forma a termos acesso aos botões da placa e permitir a passagem dos diversos cabos de ligação necessários (Ilustração 9).



Ilustração 8 – Cockpit original



Ilustração 9 – Novo cockpit

Como ao acoplar o novo cockpit ao envelope do dirigível se altera o seu centro de gravidade foi necessário efectuar alguns testes de forma a se encontrar o ponto de acoplamento que oferecesse o melhor equilíbrio. Esse ponto situa-se na metade inferior do dirigível, ligeiramente mais atrás que o meio do dirigível, como se pode visualizar na Ilustração 10.



Ilustração 10 – Dois ângulos diferentes do dirigível

A função destes dois componentes é suportar toda a electrónica de controlo como, o DSP, os controladores dos motores, o *encoder*, a haste de suporte dos motores dianteiros, a câmara de vídeo usada para navegação, o transmissor de vídeo RF e ainda as baterias de alimentação de todos os componentes electrónicos do veículo (Ilustrações 11 e 12). Permite futuramente a instalação de um GPS para navegação no exterior

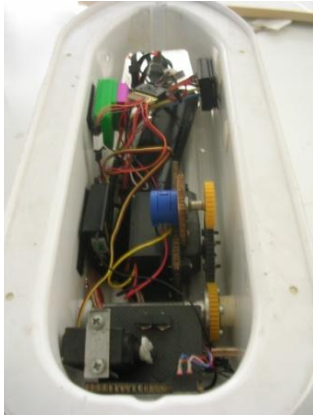


Ilustração 11 – Electrónica do cockpit dianteiro



Ilustração 12 – Electrónica do cockpit traseiro

A haste que se encontra ligada ao cockpit dianteiro é feita em alumínio, garantindo um suporte rígido de forma a poder suportar os dois motores dianteiros.

2.4 Câmara de Vídeo

Como a placa de processamento utilizada tem como entrada de vídeo uma ligação de vídeo composto, foi necessário a aquisição de uma câmara com o mesmo tipo de saída.

Assim, começou-se por efectuar um estudo sobre câmaras, as suas especificações e características, de modo a escolher a câmara que melhor se adaptaria às necessidades do projecto.

Após se definirem as especificações desejadas e devido do facto de esta ser acoplada ao dirigível, procurou-se uma câmara que oferecesse a melhor escolha. Assim, a câmara tem de ser leve para minimizar o peso que o dirigível suporta, deve ser resistente para suportar pequenos acidentes que possam ocorrer durante a utilização do dirigível, ser de dimensões reduzidas, ser pouco sensível à luz pois o local de testes sofre grandes alterações de luminosidade durante o dia e por último possuir um ângulo óptico alargado, grande angular. Esta última especificação deriva principalmente do facto do local de testes ser no interior de um pavilhão gimnodesportivo, o que implica testes em voo de baixa altitude. A baixa altitude combinada com uma câmara de ângulo óptico pequeno dificultaria o seguimento de um objecto por parte do dirigível, pois pequenos movimentos do objecto em terra traduzem-se em grandes deslocações na imagem recebida pela câmara.

Através da visualização das Ilustrações 13 e 14 pode-se verificar a diferença entre duas câmaras com diferentes ângulos ópticos e diferentes sensibilidades à iluminação.



Ilustração 13 – Imagem da câmara 1



Ilustração 14 – Imagem da câmara 2

Como se pode verificar, a imagem da câmara 1 possui um ângulo óptico menor que a grande angular da câmara 2 (82°), o que se traduz na abertura da imagem recebida. Em contrapartida a câmara 2 distorce ligeiramente a imagem ao longo das suas margens.

Também se visualiza facilmente a diferença de sensibilidade à luz entre as câmaras, traduzindo-se numa imagem mais escura para a câmara com maior sensibilidade à luz (câmara 1 – 10 Lux) e analogamente, uma imagem mais clara para a câmara com menor sensibilidade à luz (câmara 2 – 0,01 Lux). A baixa sensibilidade à luz permite trabalhar em ambientes mais escuros, sem que a imagem escureça em demasia, dificultando a distinção das cores e prejudicando a detecção da cor vermelha como pretendido.

No momento do teste efectuado e demonstrado nas Ilustrações 13 e 14, as câmaras encontravam-se fixadas no mesmo local e em instantes de tempo simultâneos, de forma a se poder comparar as diferenças de imagem entre ambas.

Desta forma, foi elaborada uma pesquisa das câmaras existentes no mercado que se encontrem dentro dos pré-requisitos pretendidos, e que oferecessem a melhor relação qualidade/preço. Assim, optou-se por adquirir a câmara da marca Sony HQ1 Helmet Camera, com as especificações:

- CCD Chip: 1/3" ExviewHAD+SS-HQ1
- 82° de ângulo óptico;
- *Resolution*: PAL – 795x596 pixels;
- *Power Supply*: 12VDC;
- *Power Consumption*: 150mA or less;
- *Weight*: 65g;
- *Low light sensitivity of 0,01Lux.*



As especificações mais detalhadas da câmara encontram-se na pasta Anexos -> Datasheets, que se encontra no CD acoplado a esta tese.

2.5 Motores

O **sistema de propulsão do dirigível** é composto por três motores que são os responsáveis pela locomoção do mesmo.

Estes motores estão dispostos da seguinte forma:

Dois motores acoplados a uma haste que se encontra ligada ao cockpit dianteiro, montada sobre servomecanismos (motor + *encoder*) responsáveis pela rotação da haste permitindo aos motores girarem 180°, e tornando assim possível a deslocação do dirigível em altitude. São estes os motores responsáveis pelo movimento do dirigível no eixo X e Z, tanto no semi-eixo positivo como no negativo, pois os motores possuem reversão (Ilustração 15). Os dois motores partilham o mesmo controlador, o que obriga a funcionarem em simultâneo. Estes motores são retratados ao longo do relatório como os **motores dianteiros do dirigível**.

Um motor montado transversalmente no estabilizador traseiro inferior responsável pela rotação em torno do seu centro de massa (em torno do eixo Z) conferindo assim a orientação desejada ao dirigível (Ilustração 16). Este motor será retratado como o **motor traseiro do dirigível**.

Os eixos considerados no decorrer deste relatório encontram-se explicados no Subcapítulo 3.1.



Ilustração 15 – Motores dianteiros e haste de suporte



Ilustração 16 – Motor traseiro

Estes motores são alimentados por corrente contínua e comandados através de controladores que irão ser explicados mais pormenorizadamente no Subcapítulo 2.6, e que recebem sinais PWM (*Pulse Width Modulation*, ou seja Modulação por Largura de Impulsos).

Os **motores de corrente contínua** são motores que recebem à entrada uma tensão variável que controla a sua velocidade de rotação. Estes motores são compostos por um estator fixo, com campos magnéticos fixos criados por ímanes permanentes ou electroímãs, e um rotor móvel ligado ao eixo de transmissão de movimento, composto por bobinas que, quando percorridas por correntes

eléctricas, geram campos magnéticos. Em função da polaridade dos campos magnéticos do rotor, estes submetem-no a forças de atracção e repulsão com o estator produzindo o seu movimento giratório, que poderá ser invertido através da inversão da polaridade da tensão eléctrica aplicada, permitindo assim a inversão de marcha no movimento do dirigível.

Os motores utilizados no dirigível são:

- Dianteiros (2):

- Marca – Graupner;
- Modelo – *Speed 400*;
- Tensão – 7,2 Volt;
- Peso – 73 gramas.



- Traseiro (1):

- Marca – Multiplex;
- Modelo – Permax 480;
- Tensão – 7,2 Volt;
- Peso – 92 gramas;



- Rotação da barra (1):

- Tensão – 5 Volt;
- Inclui *encoder* para medição.



Aos motores dianteiros encontra-se acoplado um desmultiplicador (*gearbox*), com uma proporção de 1:4,1. O motor traseiro está ligado directamente à hélice.

Como se pode verificar através da visualização das ilustrações acima, as **hélices** utilizadas são de duas pás sendo as dianteiras com a característica de 279x120mm e a traseira de 180x80mm. O primeiro valor corresponde à dimensão do diâmetro das hélices e o segundo é o factor de propulsão que corresponde ao valor que em cada revolução da hélice esta avança no sentido perpendicular à sua revolução sem escorregamento.

Os motores foram testados de forma a confirmar o seu bom funcionamento, com o auxílio de um gerador de funções da Tektronix, modelo AFG 3022, capaz de reproduzir ondas PWM com as características desejadas.

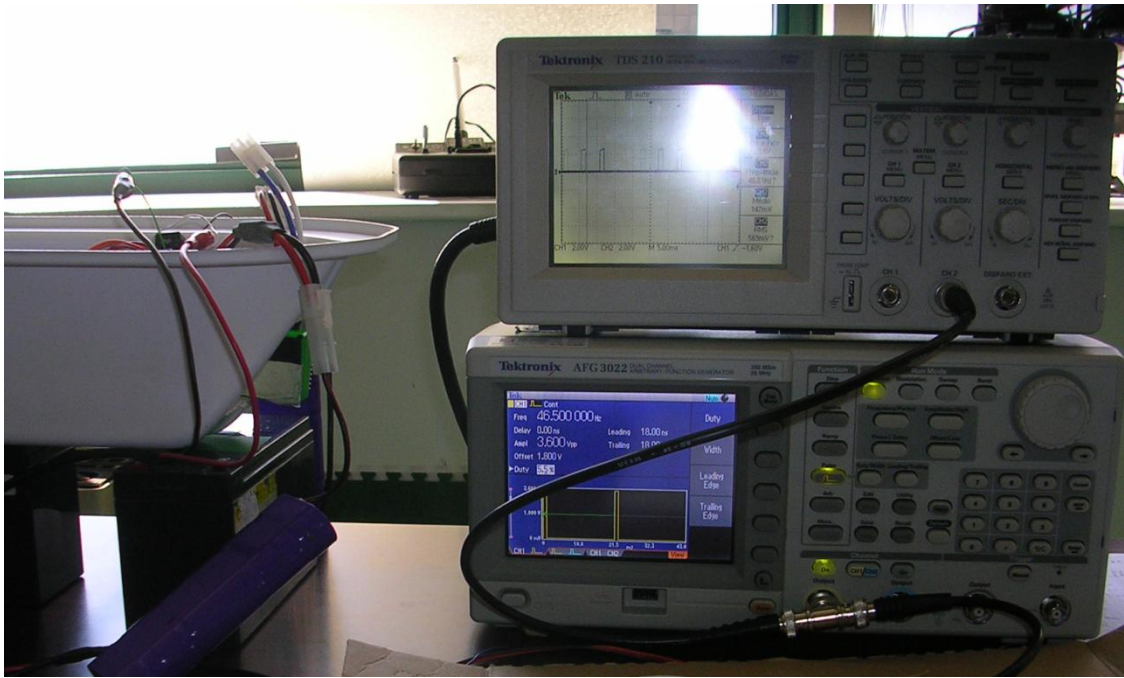


Ilustração 17 – Teste dos motores utilizando o gerador de funções (em baixo) e visualizando o sinal recebido com auxílio de um osciloscópio (em cima)

Durante os testes foi possível verificar, com o auxílio de um osciloscópio da marca Tektronix modelo TDS 210, que existe uma “zona morta” (Ilustração 18) na inversão da rotação da hélice, problema que foi resolvido através da implementação de um pequeno algoritmo no código do DSP, o qual impõe um *offset* de um valor ao sinal enviado para o controlador do motor traseiro e dois valores para o controlador dos motores dianteiros, de forma a este não coincidir com a zona de transição. Durante os testes verificaram-se os valores de *dutycycle* de funcionamento dos motores criando-se as tabelas dos valores mínimos, máximos e da “zona morta” de cada um, apresentados nas Tabelas 1, 2 e 3.

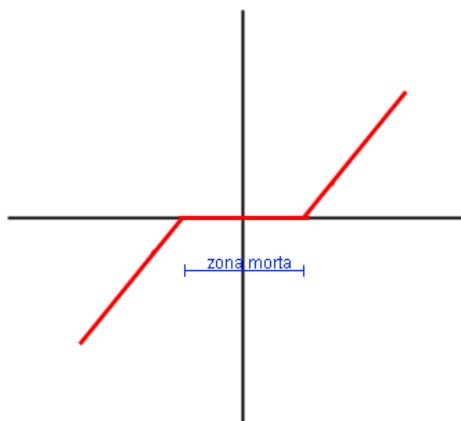


Ilustração 18 – Gráfico do funcionamento do motor

Tabela 1 – *Dutycycle* do motor traseiro

<i>Dutycycle</i> (%)	Movimento do motor
5,0	Roda ao máximo para a esquerda
7,2	Parado – zona “morta”
7,4	
11,2	Roda ao máximo para a direita

Tabela 2 – *Dutycycle* dos motores dianteiros

<i>Dutycycle</i> (%)	Movimento dos motores
5,0	Roda ao máximo para a esquerda
6,8	Parado – zona “morta”
7,2	
10,0	Roda ao máximo para a direita

Tabela 3 – *Dutycycle* do motor de rotação da haste dos motores dianteiros

<i>Dutycycle</i> (%)	Rotação da haste
5,2	+90°
6,0	0°
7,4	-90°

Para especificações mais detalhadas dos motores, ver a pasta de Anexos -> Datasheets, que se encontra no CD acoplado a esta tese.

2.6 Controladores

Os **controladores** são os responsáveis pela conversão entre o sinal de controlo (PWM) – sinal de entrada – e o sinal recebido pelos motores de corrente contínua – sinal de saída –, controlando assim a velocidade de rotação do motor em função da tensão que se aplica aos seus terminais.

O formato do sinal conhecido como **Pulse Width Modulation (PWM)** ou Modulação de Largura de Impulso, tem por base manter a amplitude e frequência do Impulso variando a sua largura de acordo com a amplitude do sinal modulador. Assim, a largura ou duração – **dutycycle** – do Impulso irá permitir variar a velocidade de rotação dos motores.

Estes controladores eram originalmente comandados através de um módulo receptor de rádio frequência (RF) de sinais do telecomando. Um dos objectivos deste projecto foi a anulação desse receptor passando a ser a placa de processamento *onboard* a responsável pelo envio dos sinais de comando do dirigível para os controladores. Desta forma todo o controlo é feito no dirigível, tornando-o um veículo aéreo autónomo.

No dirigível em estudo existem três controladores, um para os motores dianteiros (são motores não independentes), outro para o motor traseiro e outro para o controlo da rotação da haste que suporta os motores dianteiros. Todos eles encontram-se instalados dentro do cockpit dianteiro e são alimentados por duas baterias, uma para o controlador dianteiro, e a outra é partilhada pelos controladores traseiro e de rotação de forma a otimizar o peso do dirigível. Estes controladores são também responsáveis pela alimentação dos seus respectivos motores.

Derivado do facto do módulo RF ser desactivado do circuito electrónico do dirigível, foi necessário alterar o processo de alimentação do **sistema de rotação da haste** que suporta os motores dianteiros, controlador + motor + *encoder*, pois era alimentado através desse mesmo módulo, que fornecia a tensão de 5 Volts necessária para o seu funcionamento. Desta forma, foi escolhida e implementada uma solução que utiliza a tensão que o controlador do motor traseiro fornece à saída, pois desta forma não é necessário adicionar uma bateria extra de 5 Volts, o que aumentaria significativamente o peso do dirigível, visto que a tensão à saída do controlador é suficiente para alimentar todo o sistema de rotação da haste.

Ao sinal de controlo do sistema, devido às especificações do controlador, foi necessário impor um sinal de *offset* de 3 Volts, através de duas pilhas AA de 1,5 Volts, de forma a conseguirmos comandar o sistema através dos comandos enviados pela placa de processamento *onboard*, pois as saídas de áudio cortavam a componente continua do sinal.

Este *offset* não foi possível ser executado através de código na placa de processamento, porque a placa reajusta o sinal médio de saída a zero não permitindo criar o sinal pretendido à saída da mesma.

Após o estudo detalhado do controlador do motor traseiro verificou-se que o mesmo possuía um sistema de protecção no arranque, conhecido por *Power On Protection* (POR), que só permite o início de funcionamento do motor após o sinal de controlo do mesmo ter atravessado a posição de repouso, isto é, o motor só começará a receber comandos vindos do seu controlador após este ter recebido o sinal de controlo referente à rotação nula do motor. Este sistema de protecção é utilizado em alguns controladores de forma a evitarem o descuido do utilizador ao ligar os motores com a indicação de comando no máximo, que poderia causar situações indesejadas.

Foi elaborado um algoritmo que inicializa o controlador do motor traseiro através do envio de um sinal PWM com um *dutycycle* abaixo do valor de repouso, e seguidamente o envio de um sinal com *dutycycle* acima. Desta forma, o controlador do motor traseiro fica inicializado e preparado para receber comandos para enviar para o motor.

É aconselhável, no caso de se adquirirem motores mais potentes para o dirigível, adquirirem-se também controladores com sistema POR.



Ilustração 19 – Controlador dos motores dianteiros



Ilustração 20 – Controlador do motor traseiro

As especificações mais detalhadas dos controladores dos motores encontram-se na pasta de Anexos -> Datasheets, que se encontra no CD acoplado a esta tese.

2.7 Baterias

Foi elaborada uma pesquisa sobre as baterias existentes no mercado de forma a se encontrar a solução que melhor serve o sistema utilizado. Foram tidos em conta certos aspectos importantes como:

- Tensão fornecida (Volt);
- Intensidade de carga (miliAmpere hora);
- Peso (gramas);
- Dimensões (milímetros).

Devido à necessidade de capacidades de energia e tensão elevadas para o funcionamento dos componentes electrónicos, optou-se por adquirir baterias mais potentes de forma a aumentar o seu tempo de duração e possibilitando a partilha de uma mesma bateria por mais do que um componente electrónico, apesar do inconveniente de estas baterias serem mais pesadas.

Desta forma, foram adquiridas as seguintes baterias:

- *Reedy Black Label 2*, Ni-MH, 7,2V, 3700mAh, 407g, 102x31x20mm – 1;
- *Reedy Black Label 2*, Ni-MH, 7,2V, 3300mAh, 399g, 102x31x20mm – 1;
- *Flight Power EVO 25*, Li-Po, 11,1V, 1500mAh, 124g, 102x31x20mm – 2.

No *cockpit* dianteiro efectuaram-se as ligações necessárias para que os três controladores sejam alimentados através de duas baterias, uma para o sistema dos motores dianteiros, e outra para o sistema de rotação da haste dos motores dianteiros e para o sistema do motor traseiro.

Esta alteração foi possível graças à elevada capacidade da bateria de 7,2 Volts que adquirimos.

A esta bateria está também ligado um interruptor cujo funcionamento será explicado no Capítulo 4.

No *cockpit* traseiro, existe uma bateria de 7,2 Volts para alimentação da placa de processamento e uma bateria de 11,1 Volts para alimentação da câmara *onboard* e do emissor RF de vídeo, utilizado unicamente para fins de *debug* e visualização, em terra, das imagens que a câmara do dirigível está a captar.

Para mais especificações sobre baterias ver a pasta de Anexos -> Datasheets, que se encontra no CD acochado a esta tese.

2.8 Placa de processamento – DSP

Digital Signal Processor (DSP) é um microprocessador programável, com uma arquitectura especializada capaz de operar em tempo real no processamento digital de sinais com velocidades muito superiores aos microprocessadores para aplicações genéricas.

Este *chip* é o cérebro da placa de processamento utilizada, que é composta para além do microprocessador DSP por diversos outros componentes necessários para o seu funcionamento, como memórias internas, *cache* de dados e instruções, unidades de DMA (*Direct Memory Access*), *timers*, portas externas (porta serie, portas RCA, portas para módulos de extensão), entre outros.

O processamento de um sinal digital é feito com recurso a operações simples (adição, multiplicação e transferência de memória), sendo necessário no caso de sinal analógico (saída de transdutores, p.ex.: microfone) a sua conversão para sinal digital com recurso a um ADC (*Analog-to-Digital Converter*), para posterior tratamento pelo DSP, e seguidamente a reposição do sinal de saída em sinal analógico, com o auxílio de um DAC (*Digital-to-Analog Converter*). Nestes processadores a arquitectura é especialmente desenhada para que a realização de tais operações seja efectuada com a maior rapidez possível, processando centenas de milhões de amostras por segundo de forma a reflectir o mais próximo possível um desempenho tempo-real, isto é, a capacidade de amostrar na saída um sinal “live” processado no mesmo instante em que foi amostrado. O recurso às técnicas do DSP permite que hoje em dia se realizem praticamente todo o tipo de tratamento de sinais de forma digital superando as técnicas analógicas anteriormente usadas.

A utilização destes processadores encontra-se disseminada nas diversas áreas de investigação e consumo, existindo DSP's de alta performance (digitalização de imagens), de baixo consumo de energia (produtos portáteis) e controlo digital optimizado (controlo de motores eléctricos).

Cada empresa que projecta DSP's cria um ambiente de desenvolvimento (IDE) próprio para a sua gama de chips, incorporados nas placas, facilitando a manipulação e programação do microprocessador.

Como para o controlo autónomo do dirigível através de uma câmara era necessário incorporar uma placa de processamento *onboard* que se responsabilizasse pela aquisição de imagem, seu processamento e decisão da trajectória a seguir, produzindo e enviando sinais de controlo para os motores do dirigível, adquiriu-se uma placa de processamento conhecida como DSP, especializada em processamento de imagem, pois é fundamental a rapidez de processamento das imagens captadas pela câmara de forma a ser o mais próximo do tempo-real o tempo entre a captação de imagem e a transmissão dos comandos para os motores.

Assim foi escolhido o módulo ADSP-BF561 *Blackfin*, do fabricante *Analog Devices, Inc.* Esta placa é composta por um *chip* composto por dois poderosos e eficientes microprocessadores de baixo consumo, permitindo repartir as tarefas a executar simultaneamente por ambos os *cores*. Optou-se por desenvolver a tarefa de aquisição de imagem e seu processamento no *core A*, enquanto o *core B* é responsável pela modulação do sinal de controlo enviado para os motores.

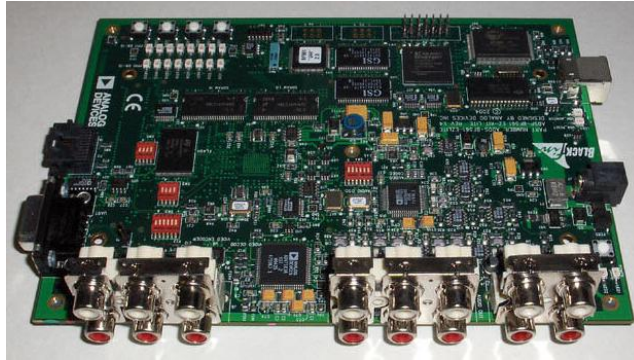


Ilustração 21 – Imagem do DSP

Características do ADSP-BF561:

- *Max. Clock:* 600 MHz;
- Número de processadores (*Cores*): 2;
- 64k *words* de L1 SRAM por *core*;
- 128k *words* de L2 SRAM;
- 3 Conectores de expansão (Memory Interface, Peripheral Interface, e Host Port Interface)
- Alimentação de 7,2V;
- Programação por ligação USB.

Para a programação da placa de processamento foi utilizado o ambiente de desenvolvimento (IDE) fornecido pela *Analogic Devices* com o nome VisualDSP++, na versão 4.5.

Foi necessário configurar os *switches* do DSP de acordo com o manual de instruções do programa *VideoInEdgeDetection*, de forma a preparar a placa para a aquisição de imagem, configurar o modo de *boot*, activar os *push buttons* e o *clock* interno. Desta forma, os *switches* devem encontrar-se nas seguintes posições:

- SW2: 1-OFF 2-OFF 3-OFF 4-OFF 5-OFF 6-ON;
- SW3: 1-OFF 2-ON 3-ON 4-OFF;
- SW4: 1-ON 2-ON 3-ON 4-ON 5-OFF 6-OFF;
- SW5: 1-OFF 2-ON 3-ON 4-ON.

O DSP adquirido para o projecto tem como portas de entrada e saída, conectores RCA de áudio estéreo e de vídeo composto, para as ligações de áudio e vídeo à placa, respectivamente. O modo de ligação dos conectores aos componentes do dirigível encontra-se descrito detalhadamente no Capítulo 4.

Na Ilustração 22, encontra-se assinalado a localização na placa DSP dos conectores de ligação de áudio, vídeo, USB e alimentação da placa, os *push buttons*, *switches* e LED's. Em [6] encontra-se o manual do DSP, com as especificações do DSP em maior pormenor.

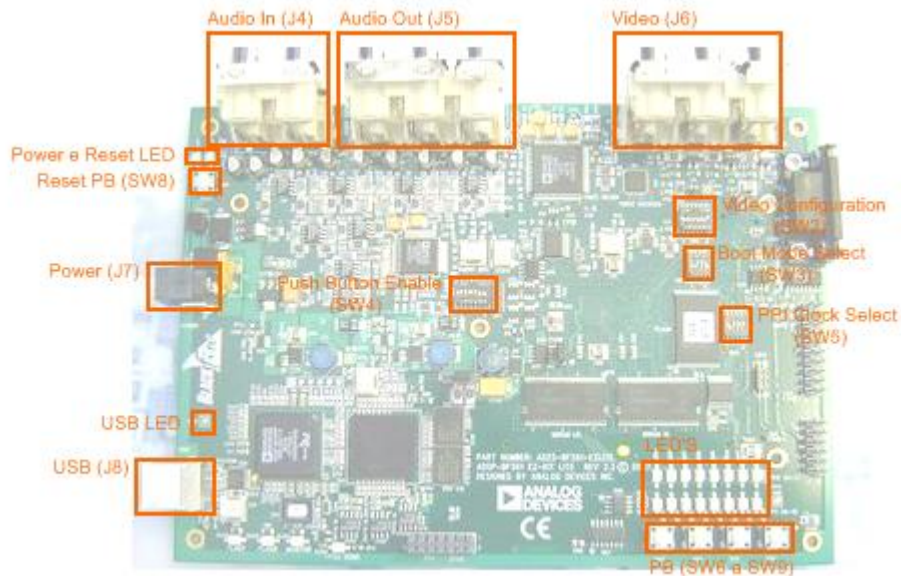


Ilustração 22 – Esquemático da localização dos conectores, LED's e *switches*

2.9 Posicionamento por GPS

2.9.1 Descrição do GPS

Global Positioning System, normalmente denominado por GPS, é um sistema de localização indispensável para a navegação moderna, fornecendo a posição tridimensional do ponto medido, em tempo real e com grande precisão, tanto em terra, como no mar ou no ar.

Este sistema foi concebido pelo Departamento de Defesa dos Estados Unidos da América, sendo inicialmente operado apenas por militares norte americanos. Actualmente o Departamento continua a financiar e controlar todo o sistema GPS, mas este pode ser utilizado gratuitamente por qualquer pessoa, numa versão menos exacta, desde que possua um receptor que capte os sinais emitidos pelas duas dúzias de satélites em órbita.

Para se conseguir adquirir a localização específica de um ponto no Globo Terrestre, através do sistema GPS, é necessário o conjunto de três sub-sistemas:

1. O segmento espacial;
2. O segmento de controlo;
3. O segmento do utilizador.

O segmento espacial (*space segment*) é composto pela constelação de satélites existente em órbita à volta do planeta Terra, com o objectivo de enviar sinais de informação referente à sua órbita (*ephemeris*), via rádio, especialmente codificados e que permitem o cálculo da posição, velocidade e tempo pelos receptores de GPS. Estes sinais transmitidos são divididos em duas portadoras, emitidas em microondas, de diferentes frequências, a L1 que transporta a mensagem de navegação (à frequência 1575,42MHz) e a L2 utilizada para medir o atraso ionosférico pelos receptores (1227,60MHz).

A constelação de satélites, composta por 24 satélites, está organizada em seis planos orbitais, constituídos por quatro satélites cada, igualmente espaçados de 60°, e inclinados cerca de 55° relativamente ao plano equatorial (Ilustração 23), a uma altitude de 20200 km, de forma a assegurar que em cada ponto da Terra são visíveis entre 5 e 8 satélites, número suficiente para calcular a localização do receptor com eficácia.



Ilustração 23 – Distribuição das órbitas e satélites pela Terra

O segmento de controlo (*Control Segment*) é formada por um sistema de estações terrestres de rastreio, dispersas pelo mundo, ao longo da Zona Equatorial, responsáveis pela monitorização das órbitas, sincronização dos relógios atómicos de bordo e actualização dos dados de almanaque transmitidos pelos satélites.

Por fim, o segmento do utilizador (*GPS User Segment*), que inclui os receptores de rádio que captam os sinais emitidos pelos satélites, com uma unidade de processamento capaz de descodificar em tempo real a informação recebida, e os utilizadores do sistema. Um receptor GPS (*GPSR – GPS Receiver*) descodifica as transmissões do sinal de código e fase de múltiplos satélites, e calcula a sua posição com base na interceptação das pseudo-distâncias entre os satélites e o receptor, num cálculo semelhante a uma triangulação, denominada, trilateração (Ilustração 24). Destes cálculos, resulta a

posição do utilizador, que pode ser lida em diversos formatos, como latitude, longitude e altitude, ou coordenadas [X,Y,Z] reportadas num referencial que se encontra no centro da Terra, ou em coordenadas UTM (*Universal Transverse Mercator*), cujos referenciais cartesianos são o Equador e o meridiano central de cada zona (Ilustração 25). A partir da variação da posição em ordem ao tempo, ou da frequência de *Doppler*, é possível calcular a velocidade.

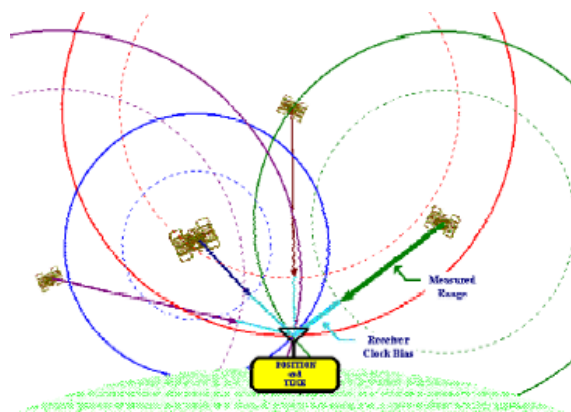
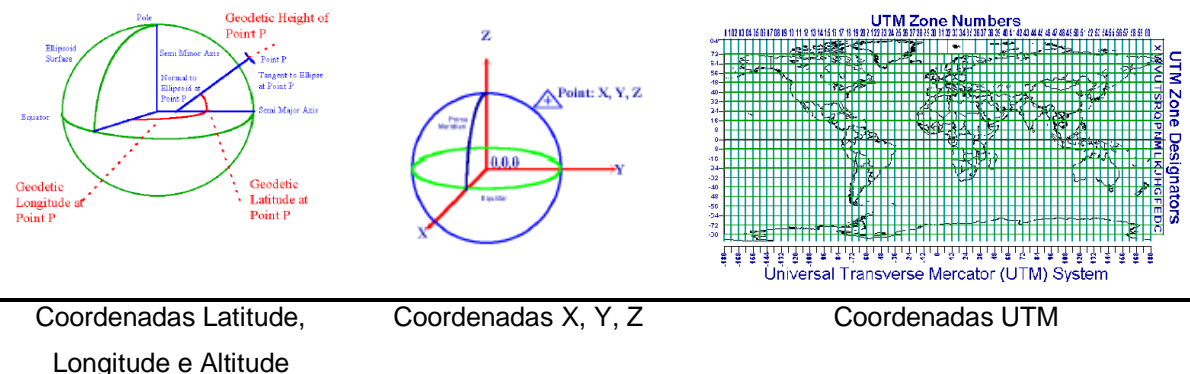


Ilustração 24 – Esquema das pseudo-distâncias entre receptor e satélites



Coordenadas Latitude,
Longitude e Altitude

Coordenadas X, Y, Z

Coordenadas UTM

Ilustração 25 – Diversos formatos de coordenadas

Para uma optimização dos cálculos, é necessário ter em atenção diversos aspectos. Os valores da posição calculados são naturalmente susceptíveis a erros, devido a ruídos externos, à utilização da leitura de informação de satélites “próximos” uns dos outros, a desvios dos dados, como por exemplo do relógio interno, dos dados transmitidos, da ionosfera ou devido a multi-trajectos, isto é, reflexão em obstáculos, edifícios, relevos naturais ou no solo (Ilustração 26).

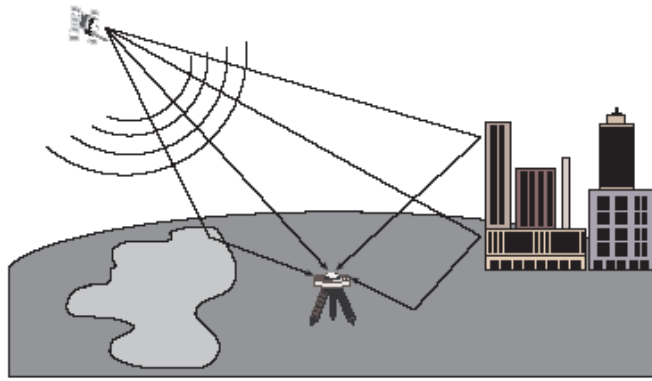


Ilustração 26 – Exemplo das reflexões dos sinais transmitidos pelos satélites

Assim, para cálculos mais exactos e precisos, como trabalhos de cartografia, pode-se utilizar um GPS com função de modo diferencial, *Diferencial GPS* (DGPS). Um DGPS recebe os sinais dos satélites e processa os dados cruzados com os sinais enviados por estações fixas em locais de referência, que fornecem correcções e informações da posição relativa aos receptores de sinal GPS, eliminando erros sistemáticos e desvios, otimizando a ordem de precisão do GPS da ordem de um metro. Esta solução só é fiável quando é necessária a máxima precisão e exactidão dos cálculos, pois os receptores deste tipo são dispendiosos e seria necessário conceber inúmeras estações de correcção, visto ser necessária a existência de uma estação num raio inferior a 100 km com o DGPS para se obter bons resultados (Ilustração 27).

Uma outra solução seria a utilização do máximo número de satélites possível para efectuar o cálculo da posição, minimizando assim ao máximo os erros.

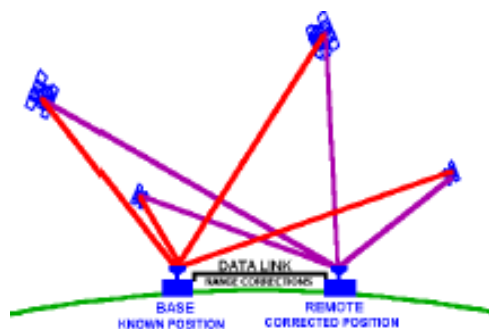


Ilustração 27 – Esquema do Sistema de GPS Diferencial

No mercado actual estão disponíveis não só sistemas de GPS, portáteis ou fixos, para cálculo da localização, mas também sistemas de GPS com diversas funcionalidades acopladas, como por exemplo a possibilidade de se obterem rumos, calcular desvios, marcar pontos, traçar rotas, entre outras. Estes sistemas podem ainda ser interligados e combinados com outros aparelhos de medição, como sondas, pilotos automáticos, radares, etc., facilitando e tornando mais eficazes e rápidas múltiplas funções do quotidiano.

Futuramente, a Europa irá lançar um sistema semelhante ao GPS, denominado GALILEO, que se encontra em fase de implementação. Pretende-se que este sistema seja controlado pela sociedade civil de modo a não estar dependente de interesses políticos ou de objectivos militares. Este sistema será composto por 27 satélites operacionais e mais 3 de reserva, em órbita a 23616km de altitude. Uma inovação do sistema fase ao existente é a possibilidade de emissão de resposta a sinais, como por exemplo, pedidos de socorro, podendo assim informar os náufragos que o seu pedido de socorro foi captado. O início de funcionamento do sistema GALILEO ainda está por definir, devido a complexidades no seu desenvolvimento.

2.9.2 Sistema GPS preparado

O sistema GPS preparado para ser implementado no dirigível é um sistema adquirido à empresa NovAtel, e é composto por uma antena de recepção de sinal e um módulo de tratamento e processamento do sinal recebido.

O modelo da antena é GPS-701 (Ilustração 28), desenhado para operar com a frequência GPS L1 (1575.42MHz) e o módulo de recepção GPS é o modelo *FlexPak* G2L-3151W, OEM4 Family (Ilustração 29). A interligação entre os dois módulos é feita através de um cabo coaxial com adaptadores para conectores TNC.



Ilustração 28 – Antena de recepção GPS



Ilustração 29 – Módulo de recepção GPS

Como primeira fase do desenvolvimento do sistema GPS para o dirigível, foi necessário compreender o funcionamento do presente GPS. Para tal, experimentou-se o sistema com recurso ao interface gráfico GPSolution 4, fornecido pela *NovAtel*, para teste das diversas funções (*logs* e comandos) do aparelho, de modo a concluir qual seria o *log* e qual o seu formato de saída (ASCII, Abreviated ASCII, ou Binário) que melhor se adaptaria ao projecto.

Pretendendo-se que o programa adquira informações sobre a localização do receptor em metros, onde por agora, apenas interessam as duas coordenadas (X,Y), e considerando a aproximação da superfície do globo, no ponto a um plano, optou-se pelo pedido do *log*: «LOG COM1 BESTUTMA ONTIME 2» ao receptor, e que devolve as suas coordenadas actuais no sistema UTM (em metros). Resumidamente este *log* pede que o receptor envie para a sua porta

COM1 a BESTUTM conseguida, em formato ASCII, em períodos de 2 segundos. Este receptor GPS possui dezenas de *logs* e comandos que permitem obter não só a posição actual já processada em vários formatos de coordenadas como o relógio com informações sobre o ano, mês, semana, dia e horas, e também o almanaque, com informação de todos os satélites, velocidade linear, informação sobre as estações, correcções, ajustes e até mesmo mudança de sistema de localização (SBAS, L-BAND, DGPS, etc.).

Durante os testes efectuados, concluiu-se que a performance do GPS é suficientemente precisa, cerca de 1,8 metros segundo o fabricante.

A ocorrência de *multipath* na antena do receptor, que retorna neste caso uma incerteza do resultado, foi também verificada experimentalmente, nos casos de ocorrência de reflexões (p.ex.: utilização da antena junto a um edifício de vidro) ou de interacção de obstáculos (p.ex.: edifícios). Nestas situações o receptor GPS perde a comunicação e há que aguardar uma melhor recepção de sinais (Ilustração 30), verificando-se por isso a necessidade de operar num espaço amplo, de modo a ser “visível” o número suficiente de satélites, com sinais directos, para o cálculo da posição do receptor.

Na Ilustração 30 demonstra-se o resultado de um teste efectuado ao sistema GPS, no trajecto Instituto Superior Técnico – Cidade Universitária, onde se podem verificar algumas descontinuidades no trajecto devido a problemas de transmissão entre os satélites e o receptor de sinal, correspondente a obstáculos presentes no caminho, neste caso os edifícios da Avenida de Roma, que estreitaram a banda de “visão”, impossibilitando a recepção de dados de quatro satélites. Em cima à esquerda, pode-se visualizar o efeito de uma reflexão causada por um obstáculo.

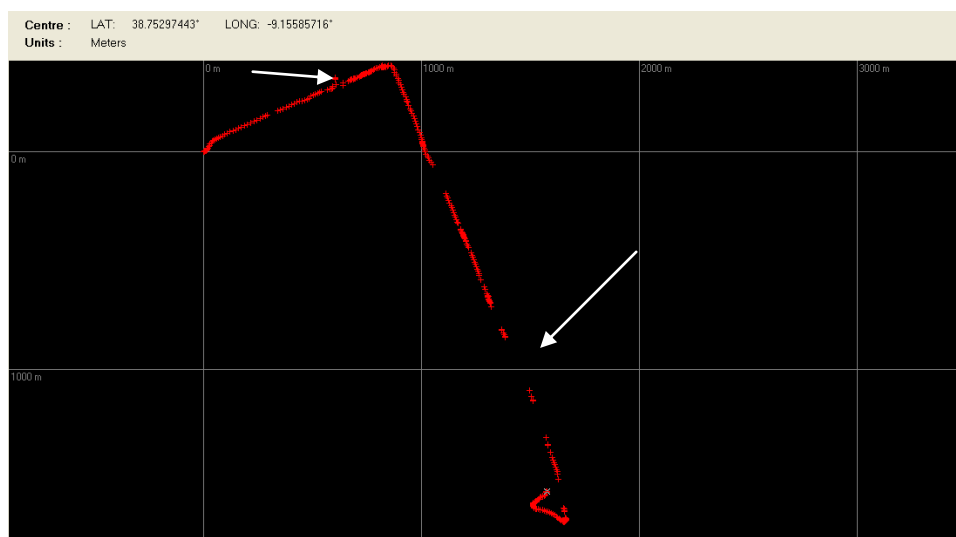


Ilustração 30 – Mapeamento do GPS num trajecto

De modo a tornar o sistema autónomo, foi necessário encontrar uma solução para a alimentação dos componentes, optando-se por uma pequena bateria de 12 Volts. Assim o módulo de recepção de sinal é alimentado pela bateria, sendo a antena alimentada através do módulo, pelo cabo de transmissão de dados (Ilustração 31).

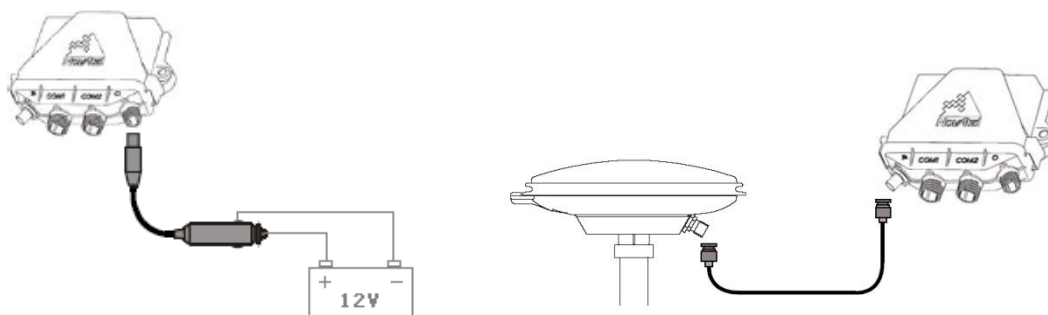


Ilustração 31 – Esquemático de alimentação dos dois componentes do sistema GPS

Método de funcionamento

De forma a se acoplar o DSP e o GPS em estreita ligação, projectou-se um sistema electrónico simples de orientação e localização, resolvendo a problemática dos protocolos de comunicação, avaliando simultaneamente as capacidades do DSP. Assim projectou-se um algoritmo no DSP que pedisse informações ao GPS através de sequências específicas de código, como por exemplo,

«LOG COM1 BESTUTMA ONTIME 2»,

ao qual responde automaticamente a sequencia de código para o DSP,

```
«#BESTUTMA,COM1,0,78.5, FINESTEERING,1382,302312.000,00000000,ef8c,2177;
SOL_COMPUTED,SINGLE,29,S,4287648.8 386,487942.7961,120.2957,53.7810,
WGS84,3.1845,2.2964,4.5378,"",0.000,0.000,7,7,0,0,0,0,0* d6 44f2d9»,
```

tendo este a responsabilidade do tratamento desses dados para funções de controlo.

O sistema construído baseia-se na ligação do sistema GPS (receptor + antena + bateria) ao DSP (acoplado com a placa de extensão RS-232) por um cabo RS-232. Este cabo liga a COM1 do receptor com o canal 1 da UART *daughter card* do DSP. Para alimentação do DSP é usada uma bateria de 5V, ficando assim o sistema completamente independente (Ilustração 32).



Ilustração 32 – Sistema GPS construído

3

SOFTWARE DO DIRIGÍVEL

3.1 Arquitectura do Software

Em simultâneo com o estudo e desenvolvimento da componente de *hardware* fomos recolhendo informação e desenvolvendo estratégias para a abordagem da componente de *software* do nosso trabalho.

Esta componente de *software* debruçou-se na programação dos algoritmos de controlo das trajectórias do dirigível na placa de processamento DSP,

Iniciámos pelo estudo da placa DSP, através da leitura dos manuais, pesquisa em fóruns de informações e dicas, e a experimentação dos exemplos fornecidos pelo *software* da placa.

Após alguns testes, iniciou-se o estudo detalhado do método de aquisição de imagem de uma câmara, através do DSP, com o objectivo do futuro processamento dessa mesma imagem e ajuda na navegação do dirigível. Esta aquisição de imagem foi efectuada através do exemplo fornecido apelidado de `VideoInEdgeDetection`.

Seguidamente estudou-se em pormenor o processamento de imagem no DSP. Por estudo em pormenor entende-se o estudo do código do exemplo testado, a pesquisa em fóruns de exemplos e informações, e de seguida a elaboração de pequenos exemplos, como o negativo da imagem, a execução do Algoritmo de *Sobel* para detecção de contornos e a detecção de linhas através de algoritmos eficazes para o efeito.

Após o domínio do DSP no que se refere à componente de vídeo, estudou-se a área referente ao áudio da placa, de forma a possibilitar a ligação dos motores do dirigível às saídas de som da placa, com o intuito de se controlar a navegação do veículo através do DSP.

Após a análise e conhecimento de funcionamento do DSP, elaboraram-se dois programas independentes, um de seguimento de linhas e outro de seguimento do robot ATRV Jr, descritos nos Subcapítulos 3.4.2 e 3.4.3.

A componente do trabalho referente ao controlador dos motores, em ambos os programas foi realizada em colaboração com o colega Ricardo Alcácer e integrada na sua tese, em [5], de onde se traduziram os algoritmos efectuados em MatLab para linguagem C utilizada para programar o DSP.

Em simultâneo com o desenvolvimento dos programas de seguimento, foi elaborado o simulador do dirigível, explicado em mais pormenor no Subcapítulo 4.2.

Para finalizar o projecto, foram efectuados diversos testes do dirigível no simulador UsarSim, de forma a se calibrarem e afinarem todas as variáveis dos controladores e comprovar o perfeito funcionamento dos algoritmos realizados antes de transitar, como passo final, para o mundo real, onde se efectuaram experiências com o dirigível real.

Eixos e Coordenadas Utilizadas

De seguida é realizada uma abordagem sobre os eixos de referência do dirigível que serão referidos ao longo deste relatório.

A descrição do veículo ao longo do relatório é feita com base no referencial que se encontra fixo ao veículo, sendo a sua origem coincidente com o centro de gravidade (C.G.) do dirigível.

Desta forma o eixo X é o eixo que aponta na direcção frontal do veículo, o eixo Z aponta em direcção ao solo, e o eixo Y é colocado de forma a tornar o sistema ortogonal. Cada eixo do referencial fornece mais um grau de liberdade ao movimento do dirigível referente à rotação sobre cada um deles, perfazendo assim um total de seis coordenadas independentes necessárias para definir a posição e orientação do veículo univocamente.

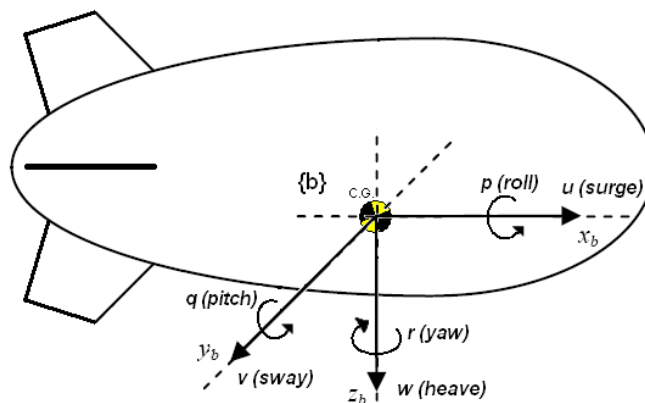


Ilustração 33 – Referencial do dirigível

O sistema de eixos da imagem recebida/processada utilizados para descrever pormenores no decorrer deste relatório encontram-se representados na Ilustração 34.

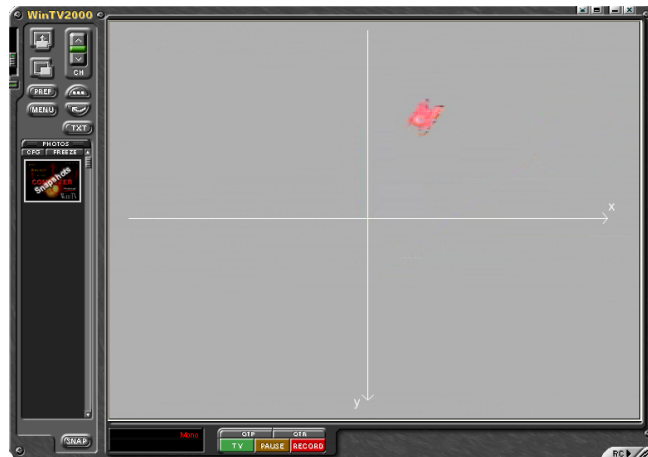


Ilustração 34 – Eixos (x, y) considerados na imagem

3.2 Algoritmos de Processamento de Imagem

3.2.1 Seguimento de Linhas

Após a pesquisa dos conteúdos fornecidos pelo *software* da placa de processamento optou-se por utilizar o exemplo de vídeo chamado *VideoInEdgeDetection*. Este exemplo aplica o algoritmo *Sobel*, em tempo real, à imagem de entrada produzindo à saída uma imagem a preto e branco, onde o pixel com a cor branca significa uma mudança brusca de intensidade de cor produzida pelas fronteiras ou intersecções entre objectos com diferentes intensidades de cor.

O **Algoritmo Sobel** baseia-se em convoluir a imagem original com duas matrizes (horizontal e verticalmente), produzindo duas novas imagens que após se calcular a raiz quadrada da soma dos quadrados das mesmas, se obtém a imagem final para detecção de fronteiras. A esta imagem, ao se aplicar um *threshold* que define o valor máximo da intensidade dos pixéis que se considera como pontos de contorno – Ilustração 35 –, obtém-se como resultado uma imagem a preto e branco, de onde se salientam a branco, as fronteiras dos objectos da imagem original, como se pode visualizar nas Ilustrações 36 e 37.

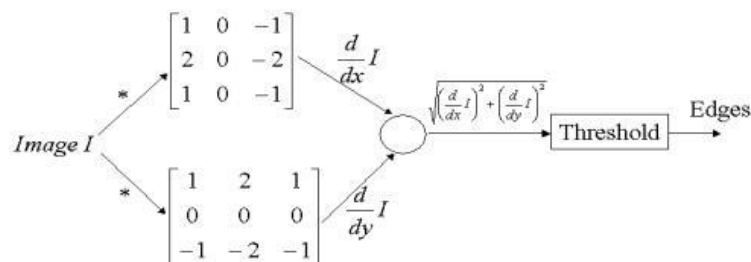


Ilustração 35 – Esquemático do Algoritmo Sobel



Ilustração 36 – Imagem original

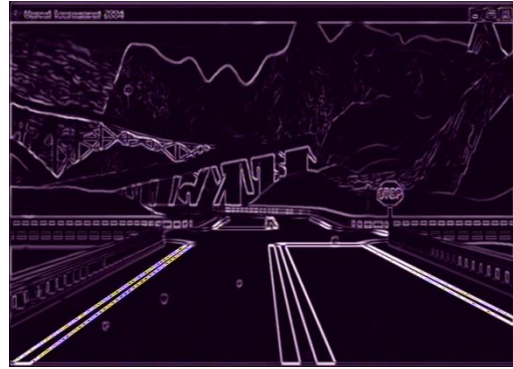


Ilustração 37 – Imagem resultante do Algoritmo Sobel

Como se pode verificar através da visualização da Ilustração 37, as fronteiras que mais se destacam são os contornos das linhas da estrada, pois são os que detêm o maior contraste entre as cores das suas fronteiras (preto-branco e preto-amarelo).

Concluimos assim que este algoritmo seria eficaz no **seguimento de linhas** pois ao implementar um *threshold* elevado obtém-se uma imagem pré-processada muito simples constituída exclusivamente pelas linhas da estrada (Ilustração 38), otimizando o desempenho do algoritmo de seguimento.

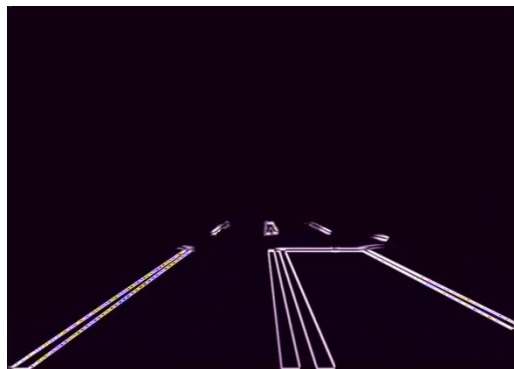


Ilustração 38 – Imagem após aplicação do Algoritmo Sobel e *threshold* elevado

O passo seguinte para o seguimento das linhas será encontrar rectas na imagem e calcular o ângulo dessas mesmas rectas de forma a se escolher o seguimento da recta mais vertical da imagem, podendo utilizar-se o ângulo da recta para controlo do motor traseiro, responsável pela rotação do dirigível sobre o eixo Z (rotação *yaw*).

Após uma pesquisa de algoritmos para esse efeito escolheu-se o **Algoritmo da Transformada de Hough**, sendo este o algoritmo que melhor se adapta às necessidades do algoritmo a implementar.

Este algoritmo tem como princípio, em cada ponto da imagem, traçar todas as rectas possíveis que intersectem esse ponto, descrevendo uma sinusóide referente ao ponto em análise, num gráfico de distância vs ângulo. Essa distância representa a distância mais curta entre a origem pré-estabelecida da imagem e a recta que intersecta o ponto em estudo (a recta da distância cruza a recta em estudo com um ângulo de 90 graus). O ângulo representa o ângulo do segmento de recta da distância mais curta ao eixo horizontal que passa na origem da imagem.

As Ilustrações 39 e 40 demonstram um pequeno exemplo do funcionamento do algoritmo.

Este algoritmo baseia-se na equação $d = x \cdot \cos \theta + y \cdot \sin \theta$ que descreve cada recta que intersecta o ponto (x,y) , onde d e θ correspondem à distância e ao ângulo, respectivamente, acima mencionados.

Assim, ao efectuar este algoritmo em todos os pontos da imagem que se considerou como sendo pontos fronteira e sobrepondo os gráficos resultantes obtêm-se um gráfico de distância vs ângulo onde os pontos mais acentuados correspondem às características das rectas existentes na imagem, como se pode visualizar no exemplo da Ilustração 41.

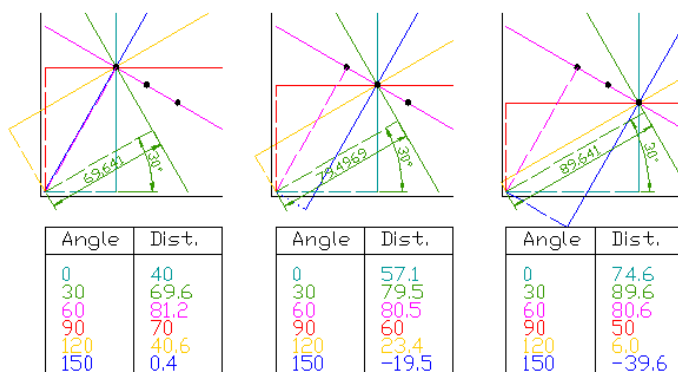


Ilustração 39 – Exemplo de 3 pontos de uma recta e 6 possíveis rectas que os intersectam

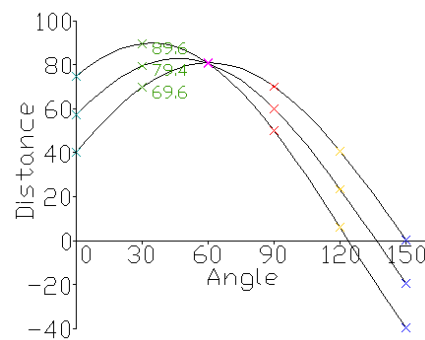


Ilustração 40 – Gráfico distância vs ângulo resultante dos 3 pontos

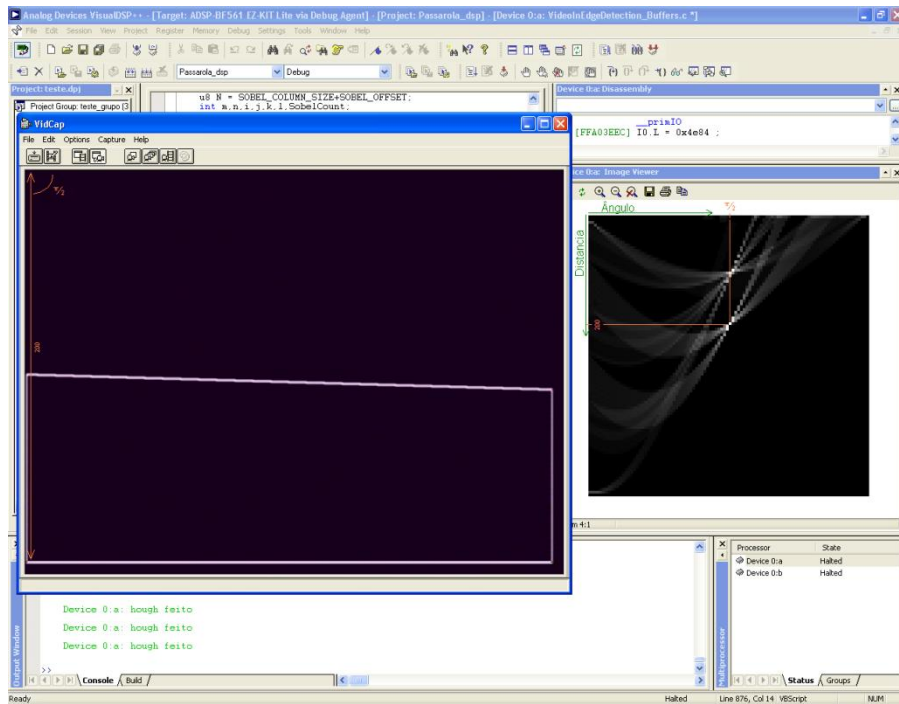


Ilustração 41 – Resultado do Algoritmo de *Hough* (à direita) de uma imagem simples (à esquerda)

Como para o seguimento de linhas com o dirigível é benéfico conhecer-se a distância das rectas da imagem em relação ao centro da mesma de forma a calcular-se o erro de direcção do dirigível da forma mais simples, alterou-se o algoritmo de *Hough* já implementado no que diz respeito à deslocação da origem da imagem para o centro da mesma e o cálculo do ângulo da recta da distância mais curta até 2π (Ilustração 42).

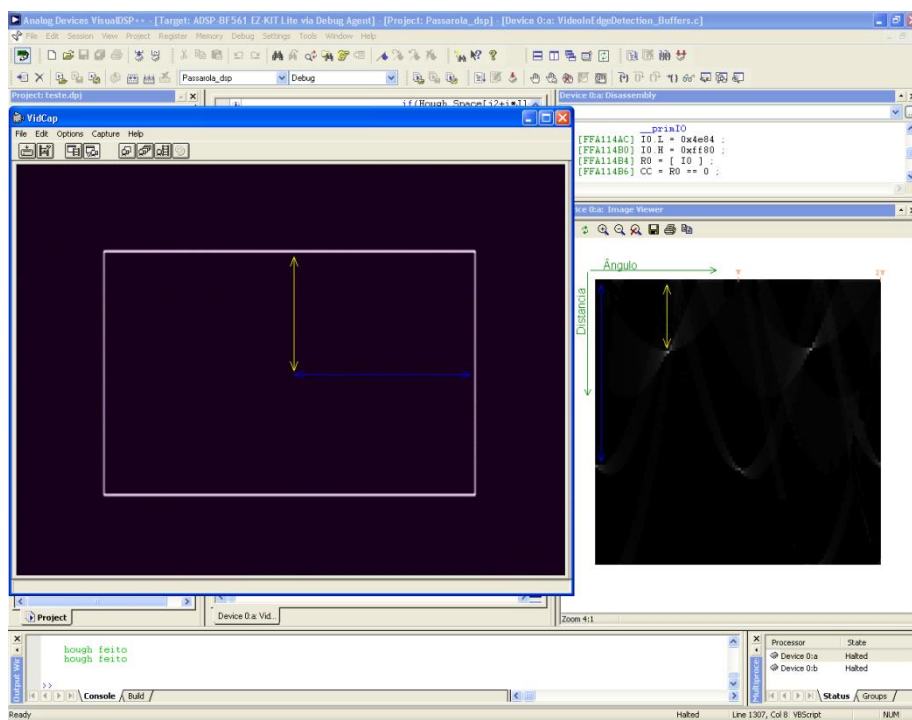


Ilustração 42 – Resultado do Algoritmo de *Hough* de uma imagem

Devido do facto de este algoritmo obrigar a inúmeros cálculos por parte do DSP, verificou-se que após a sua implementação, o DSP começou a oferecer lentidão durante o processamento das imagens. Desta forma, foi necessário encontrar uma solução que optimizasse a velocidade de processamento sem comprometer as qualidades no processamento. Esta solução passou por escalar os valores de distância (Máximo = 50) e ângulo do gráfico resultante de *Hough* (Máximo = 100), visto que este não exige elevada precisão.

Desta forma, o gráfico de *Hough* compacto, fornece muito maior rapidez tanto nos cálculos de execução do mesmo como no seu processamento.

3.2.2 Seguimento do Robot ATRV Jr

Foi elaborada uma pesquisa exaustiva ao ficheiro de exemplo apelidado de *VideoInEdgeDetection*, de forma a encontrar a variável de memória onde se aloja a imagem de entrada e saída do DSP, de modo a ser possível fazer o processamento da mesma entre a passagem da imagem de entrada para a saída. Essas variáveis são: *pVideoInBuf* a de entrada, e *pVideoOutBuf* a de saída.

Como para o seguimento do robot ATRV Jr, não é necessário executar o algoritmo *Sobel*, este foi desactivado de forma a optimizar o programa no que respeita ao tempo de processamento.

Como o robot a ser seguido é um robot em que a cor vermelha é predominante, procurou-se encontrar a solução mais eficaz para a sua detecção. A hipótese mais simples passaria por encontrar os pixels vermelhos da imagem recebida pelo DSP, o que se reflectiu numa pesquisa sobre o tipo de formato com o qual a imagem de entrada é guardada na memória da placa de processamento.

O **formato da imagem** recebida pelo DSP é conhecido como **YUV ou YCbCr, na variação 4:2:2**. Este formato define uma cor através das suas características de luminância (Y) e crominância azul (U) e vermelha (V), e a variação do formato corresponde ao método de armazenamento em memória da imagem. Foram elaborados alguns testes à imagem, de modo a se encontrar a melhor solução para a detecção dos vermelhos. Optou-se inicialmente por anular as componentes da imagem Y e U, deixando unicamente activa a componente da crominância vermelha (V).

Aplicando um *threshold* à componente V da imagem, define-se a gama de crominância vermelha pretendida, podendo restringir a gama de cores que se irá encontrar. Esta solução oferece um resultado razoável na detecção de vermelhos na imagem, como se pode verificar na Ilustração 43.



Ilustração 43 – À esquerda encontra-se uma imagem unicamente composta pela componente V da imagem da direita

Foi elaborada outra alternativa para a detecção de vermelhos que se baseava na transformação do formato YUV da imagem em RGB (Red, Green and Blue). Esta solução não se mostrou favorável devido ao número de cálculos necessários para efectuar a transformação, tornando o algoritmo muito pesado.

Desta forma, optou-se por manter a solução primeiramente descrita que após a realização de alguns testes utilizando a câmara do dirigível, se concluiu necessitar de ser otimizada ao nível da robustez às diferenças de iluminação (responsável pelas variações na componente Y) do terreno, de modo a captar toda a gama de vermelhos.

De forma a resolver esta deficiência elaboraram-se diversos testes a várias gamas de vermelhos, com diferentes exposições de luminosidade. De seguida processaram-se os resultados através de um pequeno programa elaborado em MatLab, para verificar em que gama de valores se encontra cada componente da cor (YUV) para os diferentes pixéis vermelhos, através da visualização dos gráficos resultantes, de modo a criar-se *thresholds* específicos para cada uma das componentes, melhorando a pesquisa de vermelhos (Ilustração 44).

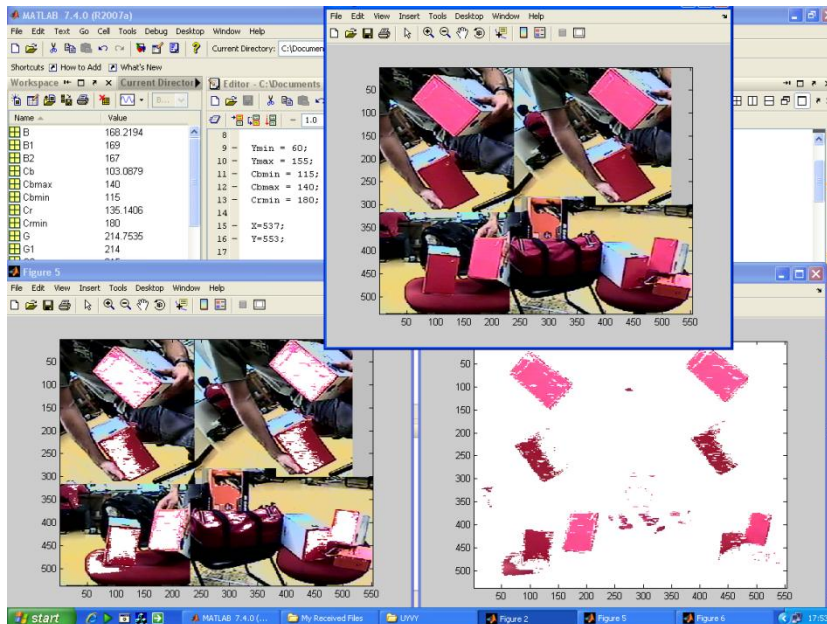


Ilustração 44 – *Output* do programa MatLab (Em cima - imagem original; à esquerda: imagem original sem os pixels considerados vermelhos pelos *thresholds*; à direita: imagem dos pixels considerados vermelhos)

Como se pode visualizar na Ilustração 44 as gamas escolhidas para os *threshold* das componentes são suficientes para se conseguir captar um vasto leque de vermelhos. Desta forma, conclui-se ser esta uma boa solução para restringir a procura de pixels vermelhos.

O formato de imagem YUV pode ser armazenado em memória de diversos modos.

No caso do DSP adquirido, este armazena a imagem utilizando o **formato UYVY**, sendo este o formato mais popular da variação 4:2:2 do formato YUV.

Este formato é o mais comum na codificação de imagem devido à relação qualidade de imagem resultante vs compactação de informação dos pixels. Isto é, se cada componente YUV de cada pixel fosse guardado em memória, eram necessários 24 bits pois para guardar a informação de cada componente são necessários 8 bits, mas no formato UYVY só é necessário guardar 16 bits de informação de cada pixel, pois este formato faz uma junção de informações de cada dois pixels consecutivos.

Assim, o formato UYVY guarda as informações de cada pixel segundo o esquema apresentado na Ilustração 45 poupando espaço em memória e tempo de processamento, sem comprometer a qualidade da imagem.

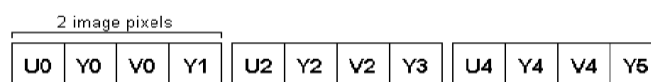


Ilustração 45 – Organização dos pixels no formato UYVY

3.3 Geração do Sinal de Áudio para Controlo dos Motores

Neste Capítulo, apresenta-se o estudo efectuado sobre o sistema de saída de áudio da placa de processamento com o intuito de usufruir dessa mesma saída para controlar os motores.

Assim, iniciou-se o estudo por um exemplo de áudio fornecido pelo fabricante da placa com o nome de `Audio_Talkthrough`. Neste caso dispõe-se como única função a de enviar o sinal de áudio recebido directamente para a saída, sem qualquer processamento intermédio.

Após se ter testado o funcionamento deste exemplo, efectuaram-se pequenas experiências com a finalidade de se aumentar o conhecimento sobre o recebimento e envio de sinais de áudio utilizando o DSP, entre elas a modificação do sinal de entrada, a alteração das portas de saída do sinal e a sua divisão.

Seguidamente foi instalado o DSP entre o módulo receptor de sinal de comando (módulo RF) existente no interior do dirigível e os controladores dos motores do dirigível. Esta instalação possibilita o teste à capacidade do DSP de reproduzir os sinais necessários para o controlo dos motores.

Após a confirmação das capacidades do DSP, iniciou-se a programação de sinais, que permitam comandar os motores do dirigível. De modo a prevenir todo o sistema electrónico do dirigível de erros de programação e picos de tensão testou-se previamente todo o conjunto com a saída do DSP ligada a um osciloscópio de forma a confirmar o perfeito desempenho do DSP, comparando os sinais por ele emitidos com os sinais enviados directamente do receptor RF do dirigível para os controladores dos motores.

Como foi referido no Subcapítulo 2.6 os controladores recebem, à entrada, sinais PWM. Desta forma, foi necessário implementar um algoritmo que produza um sinal PWM, e que ofereça, facilmente, a alteração do *dutycycle* do sinal. Assim, elaborou-se um ciclo *while* com partição por mil, correspondente a um período do sinal, onde o *dutycycle* pode ser definido com um valor entre 0 e 1000, referente à permilagem de tempo que se deseja que o sinal esteja activo.

Decidiu-se representar um período pelo valor 1000, derivado do facto dos valores de *dutycycle* pretendidos para o controlo dos motores serem de pequena ordem.

Após a confirmação do funcionamento do algoritmo elaborado, ligou-se o DSP directamente aos controladores dos motores e produziu-se um sinal PWM com *dutycycle* constante, de forma a enviar um valor de rotação constante para o motor, iniciando assim o controlo dos motores através do DSP.

3.4 Programação do DSP

Optou-se por elaborar dois programas distintos para o controlo do dirigível em virtude dos objectivos de cada serem distintos, o que se traduziu na utilização de métodos de raciocínio diferentes.

O primeiro programa descrito neste relatório tem como objectivo comandar o dirigível a partir do seguimento de linhas brancas existentes no solo.

O segundo controla o movimento do dirigível através do seguimento do pequeno robot conhecido como ATRV Jr, descrito no Capítulo 1.

Neste Capítulo retratam-se os aspectos comuns em ambos os projectos e seguidamente apresentam-se, nos Subcapítulos 3.4.2 e 3.4.3 as decisões tomadas referentes a cada projecto.

Foi implementado um **timer** no *core A* através da função `InitTimers` para verificação de tempos de processamento e possível utilização em cálculos referentes ao controlo dos motores.

Foram activados os **LED's** do DSP, para possível utilização de *debug*, o qual só foi utilizado para verificação do funcionamento dos *timers* elaborados.

Os **push buttons** do DSP encontram-se activos e em funcionamento durante o desenrolar do programa. Estes *push buttons* têm as seguintes funções:

- *Push Buttons 1 (SW6)*: Pausa/Retoma o processamento de vídeo. Este botão bloqueia a imagem actual, até se voltar a pressionar o botão. Poderá servir para verificação de uma imagem, tornando-a fixa para facilitar o seu estudo;
- *Push Buttons 2 (SW7)*: Alterna entre o envio do *Input* directamente para o *Output* e o processamento entre a recepção de imagem e o envio da mesma para a saída, tornando possível verificar o que é captado pela câmara do dirigível, para verificação da imagem recebida;
- *Push Buttons 3 (SW8)*: Sincroniza e actualiza os parâmetros de captura e envio de vídeo do DSP, permitindo reajustar o formato de imagem e recuperação de erros referentes à manipulação de imagem;
- *Push Buttons 4 (SW9)*: Termina o programa que se encontra a correr no DSP.

Para **debug** do código elaborado, utilizou-se a saída de vídeo, colorindo os pixéis de acordo com a intenção pretendida, e a função `printf`, `Expressions` e `Image Viewer` existentes no programa `VisualDSP`, que possibilitam a verificação dos valores das variáveis do código.

No caso de se utilizar a função `printf`, esta apresenta os valores das variáveis respectivas durante a execução do código, mas proporcionando um atraso durante o correr do mesmo.

A função `Expressions` encontra-se no menu `View -> Debug Windows`, e possibilita a leitura do valor de qualquer variável do código após se fazer `Halt` de ambos os processadores, facilitando a procura de eventuais erros.

Por ultimo, a função `Image Viewer` possibilita verificar qual a imagem que se encontra numa das variáveis de memória do DSP após o `Halt` dos processadores.

Depois do código de comando dos motores ter sido testado directamente ligado aos motores, verificou-se existirem interferências nos sinais enviados para os mesmos.

Estas interferências são visíveis através da visualização dos seus comportamentos irregulares durante os testes. Um dos comportamentos que demonstrou a existência de interferências foi o facto de quando todos os motores se encontram em movimento constante, valor de `dutycycle` enviado para cada um deles constante, e se alterar o valor de `dutycycle` de um deles, verifica-se que todos os outros sofrem uma alteração no sinal recebido, reagindo com uma sobrecarga ou uma quebra no seu movimento, durante o período de transição do valor de `dutycycle` alterado.

Ao verificar-se esta situação ligou-se a saída do DSP a um osciloscópio que confirmou a existência de interferências entre os três sinais de controlo.

Para resolver estas interferências implementou-se um desfasamento no envio dos sinais para os três motores, isto é, em vez de se enviar o sinal para todos eles em simultâneo, foram espaçados igualmente durante o período de envio, desfasados 333% do tempo do período, como se encontra demonstrado na Ilustração 46, permitindo assim que a influência de cada um sobre os outros seja suficientemente pequena de forma a não influenciar a rotação dos motores.

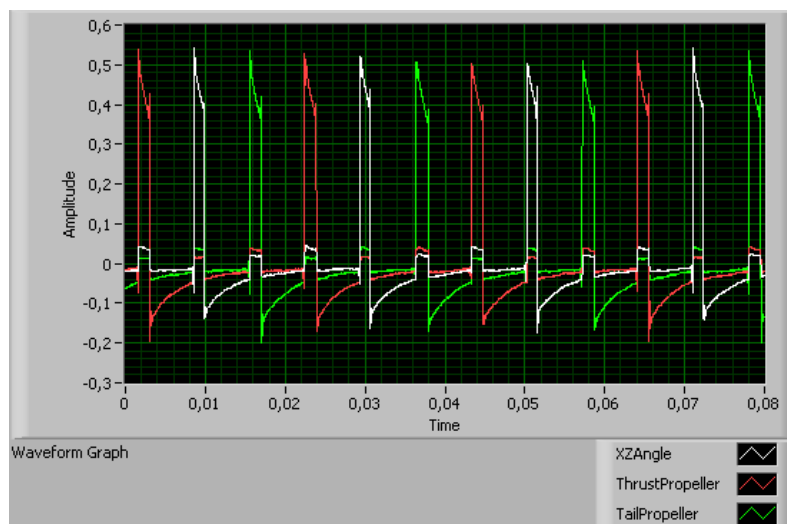


Ilustração 46 – Quatro períodos dos sinais de controlo enviados para os motores

Em relação ao método de recebimento de imagem do DSP, este adquire-a através do formato que é utilizado mais usualmente nos dias de hoje, conhecido como o **método de varrimento entrelaçado de imagens**.

Este método capta uma imagem e grava-a num vector com dois campos: o campo das linhas ímpares e o campo das linhas pares. Cada um destes campos é preenchido em cada varrimento da imagem de forma que, após finalizar o preenchimento de um dos campos, irá recommençar o varrimento da imagem para o preenchimento do campo seguinte, o que irá implicar que a imagem adquirida para o segundo campo seja da *frame* seguinte.

Resumindo, para adquirir uma imagem completa é necessário adquirir as linhas ímpares da *frame* actual e as linhas pares da frame seguinte – Ilustração 47.

Este método provoca um pequeno arrastamento nas imagens em movimento derivado do intervalo de tempo entre a captação de duas *frames*, como se pode visualizar na Ilustração 48; em contrapartida, permite maior rapidez na aquisição de imagens e futuro reenvio das mesmas, pois a placa de processamento também utiliza este método para enviar a imagem para o *output* de vídeo.

Outra vantagem deste sistema assenta nos movimentos contínuos nas imagens ao invés de se visualizar o movimento aos soluços.

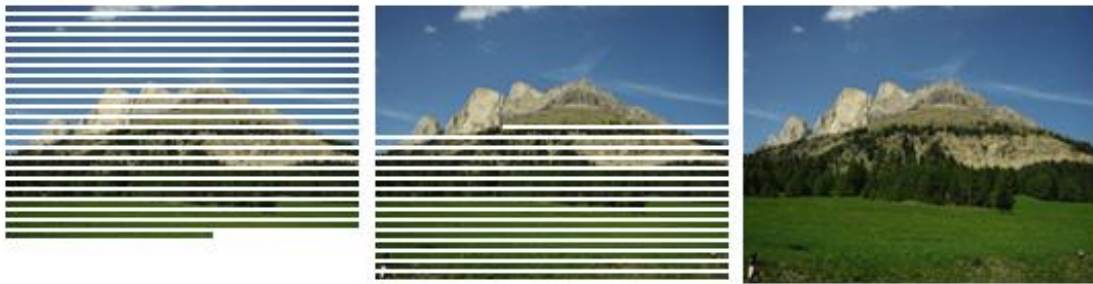


Ilustração 47 – Exemplo do método de varrimento entrelaçado

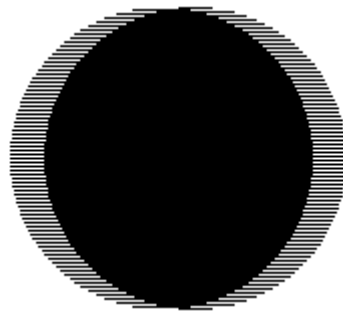


Ilustração 48 – Exemplo de uma imagem em movimento

O formato de imagem utilizado foi o **sistema PAL**, pois a câmara adquirida para incorporação no dirigível utiliza esse sistema.

3.4.1 VisualDSP++

Como foi referido, os fornecedores de placas de processamento DSP disponibilizam uma plataforma própria de trabalho para o desenvolvimento, execução, simulação e depuração do código do projecto que se pretende concretizar. Como o DSP adquirido para o projecto foi desenvolvido pela *Analog Devices*, o *software* de desenvolvimento fornecido foi o VisualDSP++, na versão 4.5.

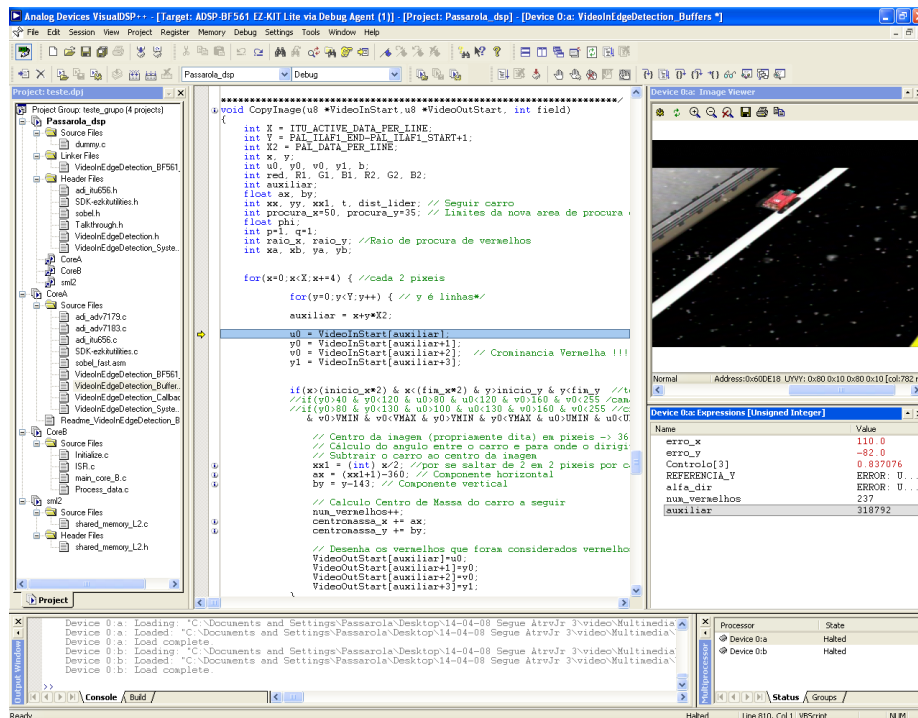


Ilustração 49 – Imagem do VisualDSP++, versão 4.5

Este *software* oferece diversas ferramentas úteis durante o desenvolvimento do código implementado na placa tais como, entre outras:

- Um *debugger* para verificação de erros de compilação;
- Uma ferramenta de visualização das imagens guardadas em memória no DSP (*Image Viewer*) – em cima à direita na Ilustração 49;
- Uma ferramenta de verificação dos valores das variáveis utilizadas (*Expressions*) – a meio à direita na mesma imagem;
- Uma ferramenta de verificação da organização dos espaços de memória utilizados;

As ferramentas acima referidas só actualizam após o *stop* do decorrer do programa.

Este programa permitiu a programação do DSP em linguagem C e *Assembly*.

3.4.2 Seguimento de Linhas

Os algoritmos mencionados neste Subcapítulo encontram-se explicados no Subcapítulo 3.2.1.

O processamento de imagem que elabora o algoritmo *Sobel* utiliza uma segmentação da imagem adquirida, seguidamente explicado, de forma a otimizar o tempo de processamento do algoritmo, pois a dimensão da imagem processada pelo algoritmo *Sobel* é mais reduzida que o tamanho da imagem completa.

A imagem recebida é processada em blocos de 120 pixels horizontais e 72 verticais, perfazendo um conjunto de 6 blocos horizontais por 4 verticais, como representado na Ilustração 50. A variável `SobelOutBlockCount` contém o número do bloco que se encontra a ser processado no momento.



Ilustração 50 – Exemplo da segmentação da imagem recebida

De forma a evitar aguardar entre a finalização de processamento de um bloco e o início do seguinte, pelo preenchimento da variável com o bloco seguinte, o algoritmo desenvolvido utiliza duas variáveis para o preenchimento dos blocos, uma para os blocos ímpares, outra para os pares (Tabela 4). Desta forma, enquanto se processa a variável referente a um bloco, a outra variável está a ser preenchida com o bloco seguinte e que fica pronto para ser processado antes de o primeiro ser finalizado.

Tabela 4 – Variáveis de processamento dos blocos

Sobel_In_Buf0	0/23	3/27				
Sobel_In_Buf1	0/24	4/28	8/32	12/36		...
Sobel_In_Buf0	1/25	...				21/45
Sobel_In_Buf1	2/26					22/46

Como foi explicado no Subcapítulo 3.4, a imagem é adquirida pelo DSP por dois vectores, referentes às linhas ímpares e às linhas pares da imagem, o que traduz na numeração dupla de cada segmento da imagem, como se pode visualizar na Tabela 4.

De forma a minimizar o tempo de processamento de cada *frame* optou-se por **processar apenas alguns dos blocos da imagem recebida**. Esta decisão deve-se ao facto de o algoritmo de seguir linhas implicar um elevado número de cálculos fazendo a execução do algoritmo arrastar-se durante o processamento de todos os blocos, e provocando atrasos consideráveis.

Assim, foram trabalhadas outras soluções como por exemplo a de se processarem os blocos em xadrez, pois como o objectivo é seguir linhas com este sistema de blocos processados será possível encontrar todas as rectas da imagem. Esta solução não foi no entanto levada avante pois continuaríamos a ter inúmeros blocos a calcular, e como só nos interessa procurar linhas a seguir à volta do centro da imagem e só no hemisfério superior, optou-se por analisar unicamente dois blocos, os blocos acima do centro da imagem à esquerda e à direita – **blocos 8/32 e 12/36**, respectivamente. Esta solução permitiu elaborar todo o processamento necessário para seguir linhas em tempo quase real.

Denota-se um pequeno atraso, mas dentro do aceitável, no processamento das frames que contêm um elevado número de píxeis de rectas dentro dos dois blocos processados, situações que são pontuais e não afectam em demasia o decorrer do programa.

Verificou-se também que a solução escolhida se enquadra dentro das especificações requeridas pois o facto de o dirigível se movimentar a uma altitude considerável faz com que as rectas que se desejam seguir se encontrem dentro da área desejada.

Após a aplicação do algoritmo *Sobel* à imagem completa aplica-se o algoritmo de *Hough* aos dois blocos definidos para controlo do dirigível. Este algoritmo, como explicado no Subcapítulo 3.2.1, retorna as posições – distância ao centro da imagem e ângulo dessa distância – das rectas existentes nos dois blocos processados.

Utilizando o gráfico resultante do algoritmo de *Hough* efectua-se uma **pesquisa dos quatro píxeis** mais pronunciados que correspondem aos quatro picos do gráfico, que por sua vez correspondem a quatro possíveis rectas encontradas nos dois blocos da imagem. Nesta pesquisa foi tido em conta o facto de uma recta ser representada por uma pequena mancha de pontos com valores elevados no gráfico de *Hough*, o que obrigou a elaborar-se um sistema que pesquise todos os pontos vizinhos de cada pixel a fim de se encontrar o ponto máximo, e só esse ser considerado.

Com o objectivo de eliminar possíveis ambiguidades ao obter-se o ponto máximo da mancha de pontos analisada, todos os pontos vizinhos a ele que se encontrem dentro de um raio pré-definido são anulados, de forma a não interferirem nos cálculos seguintes, o que vai permitir oferecer rapidez nos cálculos, pois esses pontos do gráfico não serão analisados futuramente.

Após a obtenção dos quatro pontos mais acentuados do gráfico de *Hough* são **calculados os parâmetros, declive (m) e ordenada na origem (b)**, de cada um dos quatro pontos correspondentes a quatro supostas rectas da imagem (Equação (1)). Supostas rectas, pois os pontos encontrados no gráfico de *Hough* poderão corresponder a pontos isolados na imagem em vez de rectas, pormenor que também foi considerado no decorrer do desenvolvimento do projecto através da verificação da intensidade dos pontos vizinhos do ponto analisado na imagem resultante do algoritmo *Sobel*.

$$\begin{cases} m = \cot g(\theta) = \frac{\cos(\theta)}{\text{sen}(\theta)} \\ b = \frac{s}{\text{sen}(\theta)} \end{cases} \quad (1)$$

Com os parâmetros das rectas calculados, **calculam-se os dois pontos de intersecção de cada recta com a circunferência de raio pré-definido através da constante D** e centrada no centro da imagem, através das equações:

$$\begin{aligned} \text{eq. da recta} \rightarrow & \begin{cases} i = mj + b \\ D^2 = i^2 + j^2 \end{cases} \Leftrightarrow \begin{cases} i = mj + b \\ j = \frac{-(2mb) \pm \sqrt{(2mb)^2 - 4(1+m^2)(b^2 - D^2)}}{2 * (1+m^2)} \end{cases} \end{aligned} \quad (2)$$

Seguidamente são calculados os ângulos dos segmentos de recta que unem cada ponto ao centro da imagem, sendo que o valor de referência do ângulo – 0 radianos – se encontra sobre o eixo horizontal positivo.

Posteriormente é escolhido o ângulo a seguir pelo dirigível, através do critério do ângulo que se encontra mais vertical, isto é, mais perto do valor de referência $\pi/2$, pois será esse o ângulo referente à recta que se encontra a ser seguida.

Antes de enviar, por fim, o valor do ângulo através da variável de memória partilhada `Controlo[3]` para os controladores dos motores, é feita a verificação de continuidade do movimento do dirigível, significando isso, que o valor do ângulo final é submetido ao teste de, obrigatoriamente, se encontrar dentro do intervalo de limites $-\pi/3$ e $\pi/3$ e centrado no valor do ângulo anterior.

Desta forma conseguiu-se oferecer um controlo de seguimento básico ao algoritmo protegendo-o de eventuais ruídos na imagem que adulterassem os resultados pretendidos enviando momentaneamente para os motores um valor disparatado que iria influenciar a suavidade do seu deslocamento.

As Ilustrações 51 e 52 são imagens retiradas à entrada e saída do DSP. Através destas imagens, pode-se verificar que a primeira é uma imagem enviada directamente pelo simulador *UsarSim*, que ainda não sofreu qualquer processamento.

A segunda imagem é uma imagem pós-processada pelo DSP, tendo como base a primeira. Esta imagem como se pode visualizar contém a imagem resultante do algoritmo *Sobel*, a semicircunferência utilizada para cálculo do vector de direcção do dirigível e o próprio vector de

direcção que indica qual é o erro de ângulo do dirigível em relação à vertical, com o qual é efectuado o controlo de rotação do mesmo de forma a minimizar esse erro.

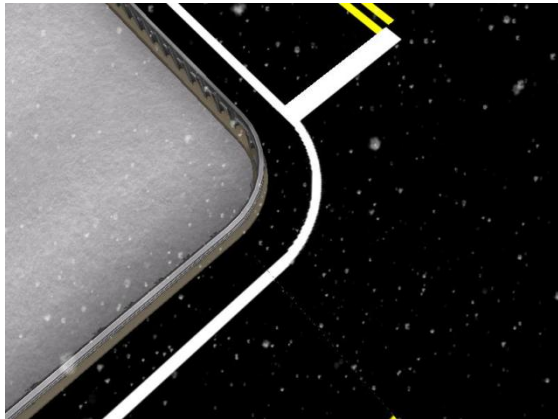


Ilustração 51 – Imagem recebida do simulador

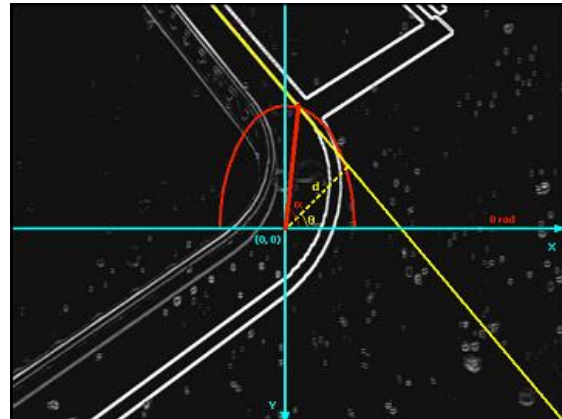


Ilustração 52 – Imagem pós-processada com o vector de indicação de direcção

Para que o projecto final corresse em tempo o mais aproximado do real obrigou a algumas optimizações de cálculos durante o seu desenvolvimento.

Um exemplo dessa optimização encontra-se no algoritmo *Sobel* onde todo o seu cálculo é efectuado numa linguagem de baixo nível, neste caso utilizou-se a linguagem *Assembly*, oferecendo grande rapidez nos cálculos efectuados em cada pixel.

Outra optimização elaborada constituiu a criação de uma tabela de valores de seno e outra de coseno, permitindo assim otimizar o seu cálculo. Esta decisão deveu-se ao facto de a placa de processamento não conseguir efectuar estes cálculos em tempo real, quando são necessários em largo número. Assim, permitiu-se a utilização destes cálculos em larga escala sem prejudicar o desempenho do algoritmo.

As tabelas são compostas por 100 valores, correspondentes aos ângulos de 0 a 2π , o que para a sua utilização no algoritmo de *Hough* onde o resultado final está escalado em 100 unidades para os ângulos comprovou-se ser um valor satisfatório, pois não se obtêm erros de aproximação dos resultados relevantes.

Como foi referido no Subcapítulo 3.4, utilizou-se a ferramenta *Image Viewer* do programa *VisualDSP*, para verificar o correcto funcionamento da aquisição de imagem, da imagem pós-processada pelo algoritmo *Sobel* e da imagem pós-processada pelo algoritmo de *Hough*.

Para a visualização das imagens é necessário correr o projecto e posteriormente fazer *Halt* a ambos os processadores na altura desejada. De seguida, executa-se a aplicação e escolhe-se a variável de memória que se pretende visualizar. Escolhe-se depois o formato em que a mesma se encontra elaborada e o tamanho da imagem pretendida (que corresponde ao tamanho do vector da variável escolhida).

No caso de se desejar visualizar a imagem resultante do algoritmo *Sobel*, deve-se escolher a variável *Image_In* ou *Image_Out*, com o formato 8bits e o tamanho da imagem de 122 por 72 pixels.

Para o caso de se desejar visualizar o gráfico resultante do algoritmo de *Hough*, deve-se escolher a variável `Hough_Space`, o formato 8bits e a dimensão da imagem de 50 pixéis verticais – Distância – por 100 horizontais – Ângulo.

Após o algoritmo de seguimento de linhas se encontrar em funcionamento, com a velocidade do dirigível no eixo X constante, deparou-se-nos um problema de seguimento da linha nas situações de curva, principalmente nas curvas mais apertadas devido à velocidade excessiva do robot.

Desta forma, optou-se por elaborar um controlador simples para os motores dianteiros, de forma a que na aproximação às curvas se reduza a velocidade, de modo a permitir o seguimento correcto da linha, sem erros.

Os controladores dos motores dianteiros e traseiro encontram-se descritos com maior detalhe em [5].

De seguida ilustra-se um diagrama de blocos com a descrição da ordem dos processos do algoritmo de Seguimento de Linhas, onde se visualiza os dois ciclos de trabalho independentes referentes a cada *core* do DSP.

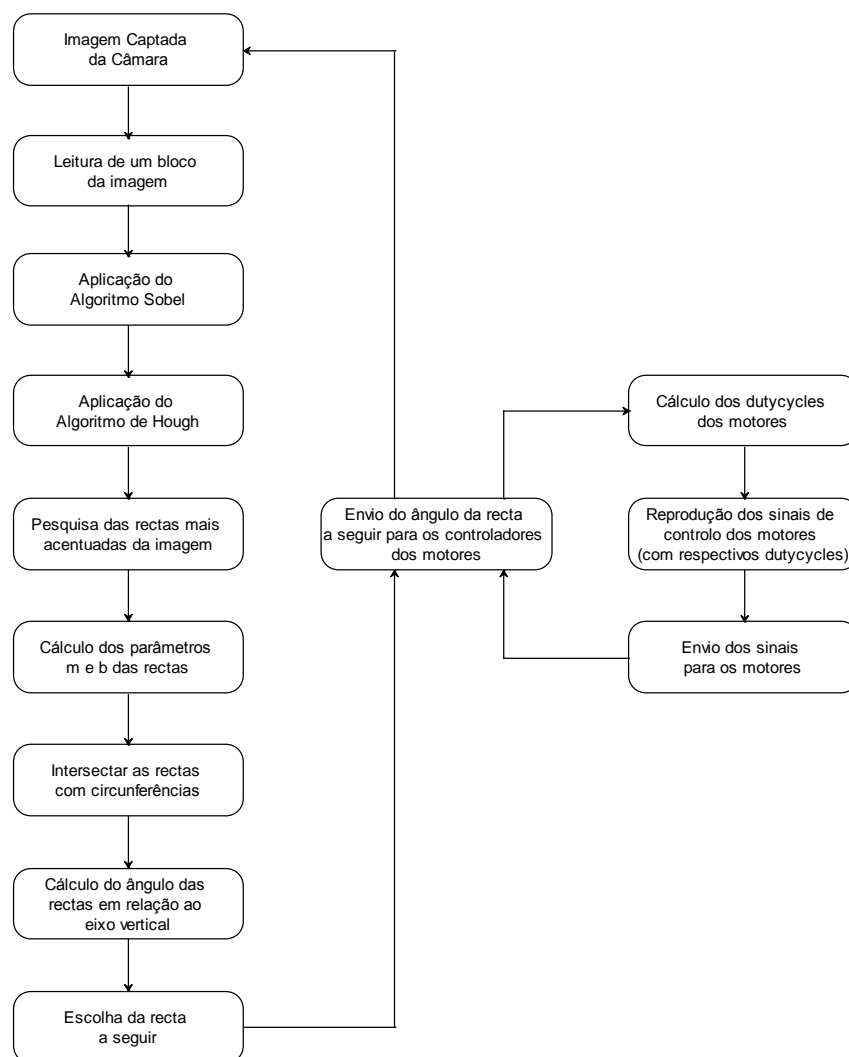


Ilustração 53 – Esquemático da organização dos algoritmos implementados em cada *core*

3.4.3 Seguimento ATRV Jr

Como explicado no Subcapítulo 3.2.2, o seguimento do robot ATRV Jr é efectuado através do **seguimento dos pixéis vermelhos da imagem**.

Da imagem resultante da aplicação dos algoritmos referidos no Subapítulo 3.2.2, optou-se por colorir toda a imagem em tom de cinzento (Y=128, U=128 e V=128), excepto os pontos considerados vermelhos dentro da área predefinida de pesquisa, de forma a facilitar a visualização da imagem e respectiva compreensão e dar a possibilidade de inserir cinco pequenos gráficos na imagem com a variação dos erros das variáveis de controlo e *duty cycles* ao longo de um período de tempo de 13 segundos, como irá ser explicado adiante.

Após a pesquisa dos pixéis vermelhos de uma *frame*, calcula-se o **centro de massa** desses pixéis através do somatório de cada uma das suas componentes e divide-se no final pelo número de pixéis considerados vermelhos (Equação (3)).

$$\begin{cases} \text{centro de massa}_x = \frac{\sum \text{pixel}_x}{\text{número total de pixeis vermelhos}} \\ \text{centro de massa}_y = \frac{\sum \text{pixel}_y}{\text{número total de pixeis vermelhos}} \end{cases} \quad (3)$$

Com este algoritmo implementado, deparamo-nos com um problema no caso de existirem dois conjuntos distintos de pixéis vermelhos na imagem correspondentes a dois objectos vermelhos. Neste caso, o algoritmo não retorna o centro de massa de um dos objectos mas sim o centro de massa dos dois objectos agregados, resultando um vector que não indicaria a posição correcta a seguir.

Um exemplo desta situação é o demonstrado na Ilustração 54.

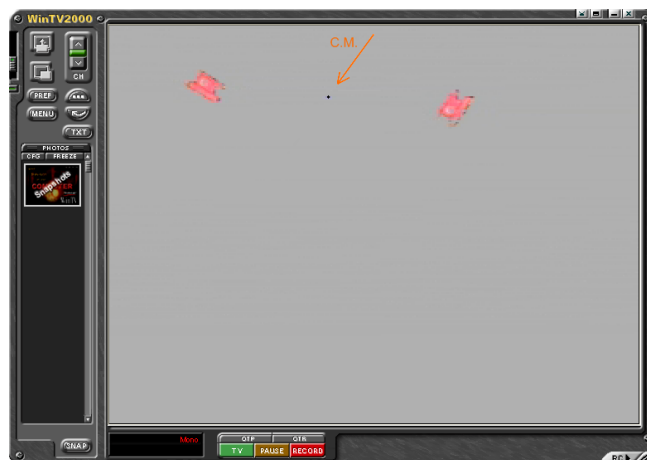


Ilustração 54 – Exemplo do centro de massa de dois objectos agregados

Assim, elaborou-se uma pequena restrição aquando da verificação da cor do pixel processado. Esta restrição consiste na pesquisa de pixels vermelhos unicamente na zona circundante do último centro de massa calculado, otimizando em termos de rapidez a execução do algoritmo, Ilustração 55.



Ilustração 55 – Exemplo do c.m. de um objecto existindo mais que um objecto na imagem

Caso a zona procurada seja demasiado pobre em pixels vermelhos devido ao movimento repentino do objecto vermelho a seguir na imagem, esta zona é alargada, de forma linear até ao tamanho da imagem, tendo como ponto de paragem um ponto mínimo de pixels vermelhos predefinido, como se pode verificar através da visualização do exemplo demonstrado na Ilustração 56. Este valor pode ser alterado de acordo com o objectivo final mas aconselha-se, caso o dirigível trabalhe a uma altitude elevada, a diminuir o número de vermelhos encontrados pois o robot ATRV Jr visível pela câmara será constituído por poucos pontos vermelhos devido à distância entre os dois.



Ilustração 56 – Exemplo do aumento da zona de pesquisa de vermelhos

Desta forma após o programa fazer o primeiro varrimento da imagem e detectar um corpo vermelho, este será seguido sem interferência de outros corpos na imagem, excepto no caso de se cruzarem dois corpos de tal modo próximos, que a zona de procura de pixéis vermelhos inclua os dois corpos. Caso esta situação ocorra o algoritmo só alterará o alvo a seguir no caso do numero de pixéis vermelhos encontrados no objecto a seguir seja menor que o numero de pixéis vermelhos do objecto que o cruzou. Estas situações poderão ser minimizadas através da adaptação do raio da zona de pesquisa de vermelhos de acordo com a altitude de trabalho do dirigível e da dimensão do objecto a seguir. Desta forma, quanto maior a altitude de trabalho e/ou menor for o objecto a seguir, menor deverá ser o raio da zona de pesquisa.

Após o cálculo do centro de massa do objecto a ser seguido, são efectuados os cálculos **das variáveis de controlo do dirigível: ângulo de rotação do dirigível, erro vertical e horizontal em relação ao ponto pretendido da localização do centro de massa do objecto a ser seguido pelo dirigível.**

As coordenadas do centro de massa do objecto alvo de seguimento situam-se sob o ponto do centro da imagem (X=360 e Y=143).

A variável de **controlo de rotação do dirigível (`alfadir`)**, é o ângulo medido desde o ponto do centro de massa do objecto vermelho a seguir e a semi-recta vertical que intersecta o centro da imagem, com origem no ponto situado 40 pixéis abaixo do ponto do centro da imagem.

Esta implementação deve-se ao facto de melhorar a reacção do dirigível quando este ultrapassa, por curta distância, o ponto final desejado (o centro da imagem), o que, caso o ângulo de rotação seja calculado com origem nesse ponto, ele não conseguiria estabilizar o objecto a seguir nesse ponto, pois sempre que o ultrapassasse, por pouco que fosse, iria rodar 180° sobre si para tentar reposicionar-se no ponto de repouso.

Esta situação suceder-se-ia no caso do dirigível se encontrar a seguir o robot terrestre em recta e este parasse repentinamente durante o seu trajecto.

Caso não se tivesse tomado a decisão de calcular o ângulo com uma pequena distância do ponto de repouso este iria, devido à sua dinâmica, ultrapassar o ponto pretendido obrigando-o a girar sobre si mesmo de forma a minimizar o erro do ângulo.

Com a solução implementada o dirigível irá ultrapassar o ponto pretendido, mas só irá rodar sobre si caso tenha ultrapassado em grande distância o ponto. Desta forma caso o robot terrestre não varie de velocidade bruscamente, o dirigível irá abrandar e seguidamente recuar após ter passado o ponto, até este se encontrar no local desejado.

O vector de erro do ângulo (`alfadir`), é calculado através da função `atan2`, retornando valores entre $-\pi$ e π , onde o valor de 0 radianos se encontra coincidente com o eixo horizontal positivo. O valor de `alfadir` é guardado na posição 4 da variável partilhada entre os dois processadores, `Controlo[3]`, e este é posteriormente alterado no ficheiro `Process_data.c`, onde é feita uma conversão dos valores retornados para valores entre $-\pi/2$ e $3\pi/2$, como representado na

Ilustração 57, de forma a facilitar o envio do sinal de controlo do motor traseiro, pois basta subtrair ao **valor de referencia do ângulo ($\pi/2$)** o valor da variável actual, para se obter o valor de erro do ângulo entre $-\pi$ e π , correspondendo ao erro positivo a rotação para a direita do dirigível (o objecto a seguir encontra-se à direita da recta vertical que passa pelo centro da imagem) e ao erro negativo a rotação contrária (o objecto a seguir encontra-se do lado esquerdo da imagem).

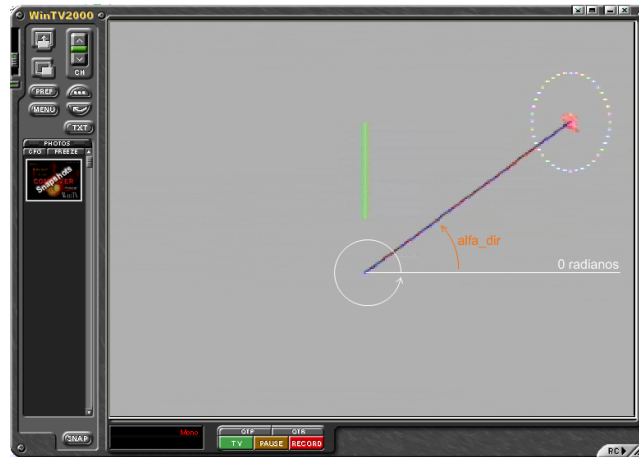


Ilustração 57 – Eixo e vector do ângulo de erro

A variável a que se chamou **controlo lateral do dirigível ($erro_x$)**, é a componente horizontal do vector de erro entre o centro da imagem e o centro de massa do objecto a ser seguido. Este erro é enviado para o processador responsável pelo envio de sinais para os motores através da variável partilhada `Controlo[0]`, que é utilizada no cálculo do controlador do motor traseiro do dirigível (ficheiro `Process_data.c`).

Os valores retornados por esta variável são positivos quando o dirigível se encontra à esquerda do robot terrestre, o que implica um erro lateral no sentido do eixo horizontal positivo (direita). Analogamente, quando o dirigível se encontra à direita do robot a seguir, o erro lateral encontra-se sobre o eixo horizontal negativo (esquerda), e portanto terá um valor negativo.

O valor máximo do erro será igual ao valor de metade dos pixéis horizontais da imagem - 360 para o caso do valor ser negativo, e + 360 para o caso contrário, sendo o **valor de referência o ponto 0**, de forma a localizar o robot terrestre no centro da imagem recebida pela câmara do dirigível.

Na Ilustração 58, é possível visualizar um exemplo do vector de erro da componente x.

Reciprocamente, a variável de **controlo vertical do dirigível ($erro_y$)** é a componente vertical do vector de erro entre o ponto do centro da imagem e o ponto do centro de massa do objecto. Devido ao facto de os pixéis da imagem crescerem no sentido descendente, o $erro_y$ tem valores negativos quando o ponto do centro de massa se encontra acima do ponto do centro de imagem, i.e., o dirigível encontra-se atrasado em relação ao robot terrestre, e valores positivos quando se encontra abaixo, i.e., o dirigível ultrapassou o ponto de referência do robot terrestre.

Este valor de erro é enviado para o processador *core B* através da variável da memória partilhada `Controlo[1]`, ao qual é feito o simétrico antes de ser utilizado no cálculo do controlador dos motores dianteiros no ficheiro `Process_data.c`.

Os valores máximos obtidos por este erro são equivalentes à metade dos pixels verticais de uma imagem completa -143 no eixo negativo de Y e +143 para o eixo positivo – e o **ponto de referência situa-se em 0**.

Optou-se por desenhar este vector com cor, como se encontra demonstrado na Ilustração 58, de modo a facilitar a análise, durante o desenrolar dos testes, do funcionamento do algoritmo.

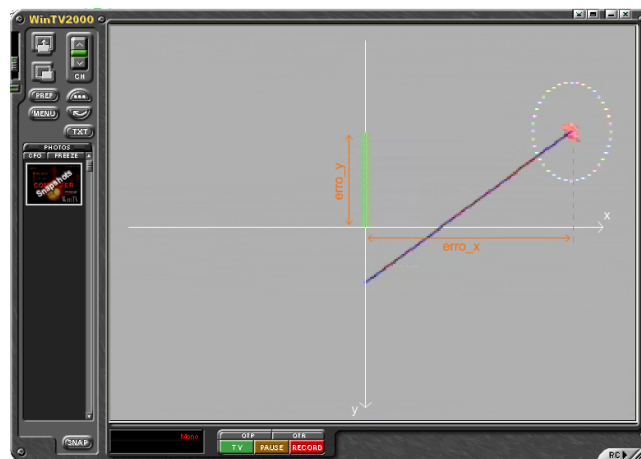


Ilustração 58 – Eixos das coordenadas da imagem e respectivos vectores de erro (em x e y)

Como foi referido, foram introduzidos cinco gráficos das diversas variáveis de controlo em função do tempo, na imagem de *output* (Ilustração 59), os quais são actualizados em tempo real. Estes gráficos dão-nos um *feedback* do comportamento do dirigível e do seu desempenho ao longo do tempo, permitindo assim confirmar a funcionalidade dos controladores dos motores na obtenção da posição de referência. Os gráficos incluídos na imagem de *output* são, em cima, os das variáveis de *dutycycle* enviadas para os motores, o motor traseiro à esquerda e os dianteiros à direita; em baixo, os dos valores de erro de posição, no eixo Y e X, e de rotação, respectivamente.

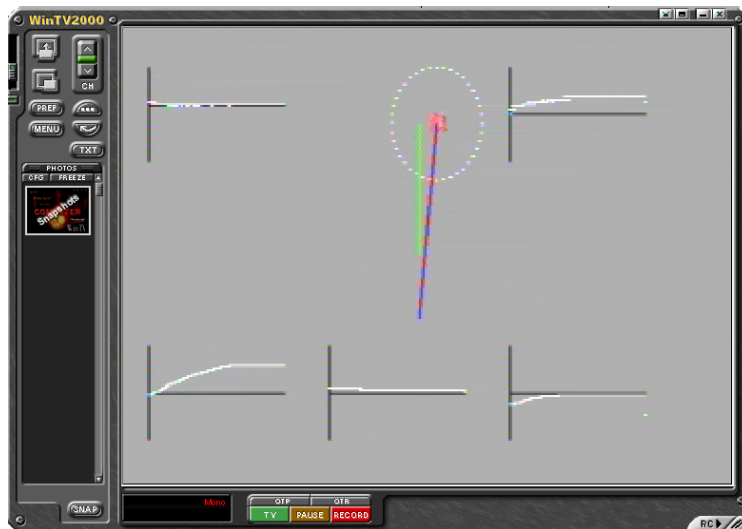


Ilustração 59 – Output do DSP enviado para a base em terra

Optou-se por manter a última instrução enviada para os motores no caso do robot terrestre sair da imagem recebida pela câmara. Esta opção deve-se ao facto do dirigível ser mais lento a arrancar que o robot terrestre, e no caso de este arrancar com velocidade elevada e sair da imagem, o dirigível irá acelerar ao máximo com o objectivo de o alcançar o mais brevemente possível. Devido à dinâmica do dirigível, após o seu corpo se encontrar em movimento, este irá conseguir facilmente acompanhar o robot ATRV Jr nos casos em que este se movimenta a velocidades elevadas.

Os controladores dos motores dianteiros e traseiro encontram-se descritos com maior detalhe em [5].

Analogamente ao Seguimento de Linhas, encontra-se o diagrama com a mesma divisão entre cores.

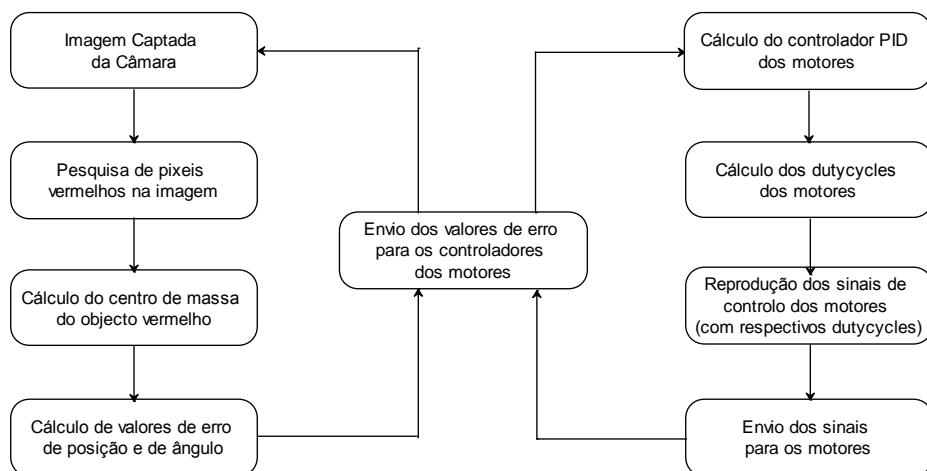


Ilustração 60 – Esquemático da organização dos algoritmos implementados em cada core

3.4.4 Sincronia dos Dois Programas

Nesta fase pretende-se juntar num único projecto os dois programas de processamento, o de vídeo e o de controlo dos motores, que se encontram a funcionar separadamente, e ambos no processador *core A*.

Neste Subcapítulo aborda-se o modo utilizado que permitiu a junção entre os dois programas de forma a se conseguir ter todo o controlo do dirigível a funcionar em sintonia.

Foi efectuado um estudo detalhado sobre os ficheiros *linker* (.ldf) de cada um dos programas, vídeo e áudio, responsáveis pela inicialização das memórias, processadores, bibliotecas e alocação de memória para vectores.

Como o DSP está equipado com dois processadores, decidiu-se fazer uma separação de funções por *cores*, de forma a não os sobrecarregar. Assim sendo, o *core A* ficou responsável pelo processamento de imagem e o *core B* pelo controlo dos motores e respectivos controladores.

Como em ambos os projectos os processadores utilizados eram unicamente o *core A*, foi necessário fazer uma pesquisa de como activar o segundo processador, memória e periféricos.

Esta divisão de funções entre os processadores foi estruturada de forma a otimizar a informação partilhada e a troca de valores entre processadores.

A junção dos dois ficheiros .ldf foi elaborada em três passos e testados consecutivamente:

1. Iniciou-se definindo certos ficheiros como bibliotecas de forma a se poder chamar nos ficheiros .c as suas funções e alocaram-se as bibliotecas dos dois projectos na memória do L2 SRAM;
2. Transferiu-se o conteúdo do ficheiro .ldf do programa de áudio, referente ao processador *core A*, para o local referente ao processador *core B* existente no ficheiro .ldf do programa de vídeo. Foi necessário ter atenção às variáveis com nomes repetidos e às diferenças de nomes das variáveis ao longo do código conjunto;
3. De seguida, juntaram-se ao projecto de vídeo, já contendo o nome ficheiro .ldf, os ficheiros do programa de áudio, pré-processados para funcionar no *core B*.

Após o passo 3 concluído obteve-se um projecto que engloba o processamento de imagem da câmara e o controlo dos motores através das saídas de áudio, em perfeita sincronia de funcionamento.

Prosseguindo-se para a fase de processamento das trajectórias a seguir por parte do dirigível.

4.1 Constituição

Para a preparação de todo o sistema de forma a ser ensaiado, foi necessário preparar dois computadores, para efectuarem tarefas distintas.

O **Computador de Trabalho** é o computador onde se efectua todo o processo de programação dos algoritmos utilizados para constituir todo o sistema. Para tal, foi necessário instalar uma placa de TV e um programa de captação de imagem que permite a aquisição da imagem proveniente do DSP, possibilitando efectuar facilmente testes e *debug* aos algoritmos elaborados.

Foi instalada também a placa de processamento adquirida, Subcapítulo 2.8, bem como o respectivo *software* de programação VisualDSP++, que se encontra descrito na Subcapítulo 3.4.1.

Depois foi instalada a placa de aquisição de sinais, Subcapítulo 4.3, e um programa de tratamento dos sinais recebidos, programa LabView descrito na Subcapítulo 4.4.

O segundo computador utilizado, é o **Computador de Simulação**, i.e., é o computador responsável por correr unicamente o simulador UsarSim, Subcapítulo 4.2, utilizado para testar todo o código implementado no DSP, antes de se transitar para o dirigível real. Este simulador encontra-se instalado num computador à parte, devido à necessidade de elevado processamento, de forma a conseguirmos uma simulação em tempo real.

Tendo-se criado um simulador, que corre num computador independente, possibilitou-se o teste de todo o algoritmo desenvolvido e implementado no DSP antes de se utilizar o dirigível real.

Desta forma e de modo a simular o ambiente de trabalho o mais próximo do real, foi elaborado um sistema de ligações que permite conectar o DSP directamente ao simulador através de uma placa de aquisição de sinais (ADC), explicada no Subcapítulo 4.3.

Este sistema permite utilizar a mesma placa de processamento que se irá implementar *onboard* no dirigível, como o cérebro do sistema simulado oferecendo assim a máxima proximidade com o sistema real utilizado, minimizando os erros e problemas existentes na transição do sistema simulado para o real.

Desta forma foi necessário modificar e criar os diversos tipos de conexões necessários para permitir as ligações entre os sistemas electrónicos do dirigível e preparou-se o *hardware* e *software* dos computadores de trabalho utilizados de modo a permitir utilizar todo o sistema em sincronia.

Preparou-se toda a montagem do sistema de forma a otimizar o número de ligações e facilitar a transição das conexões do ambiente de simulação para o ambiente real.

Na Ilustração 61 encontra-se o **esquemático das ligações necessárias de forma a simular o desempenho do dirigível no simulador UsarSim.**

A placa de aquisição referida encontra-se representada por um bloco identificado com as siglas ADC (*Analog-to-Digital Converter*).

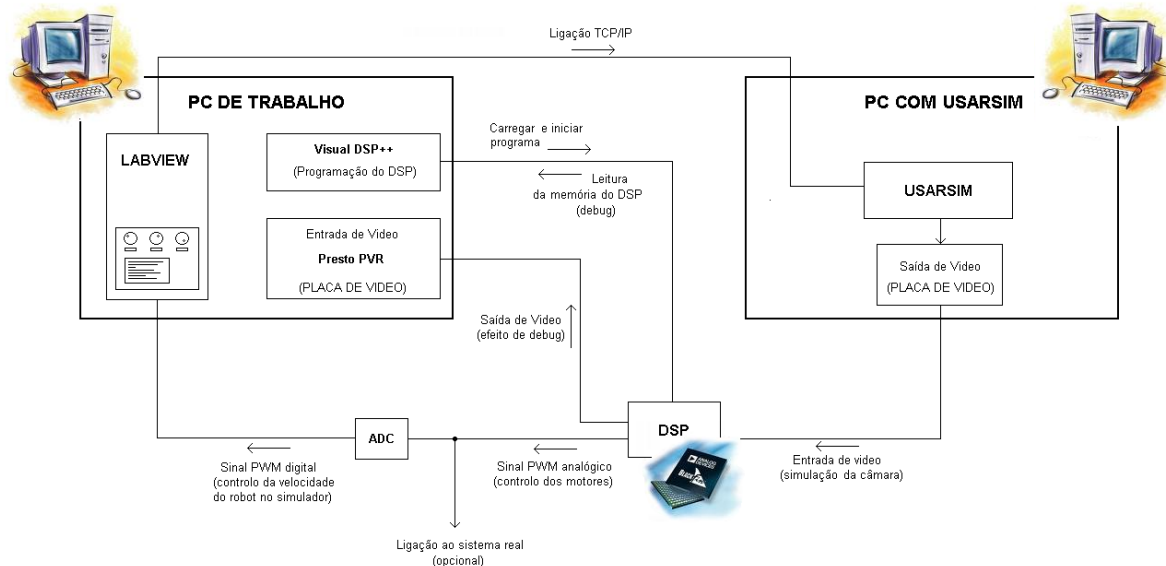


Ilustração 61 – Esquemático das ligações do ambiente de simulação

Este esquema de montagem permite a possibilidade de ligar a saída dos comandos do DSP aos motores do dirigível real permitindo visualizar os movimentos reais dos motores em simultâneo com os movimentos simulados.

Começando pela **placa de processamento (DSP)**, esta encontra-se ligada com um cabo USB ao computador de trabalho de forma a possibilitar o carregamento do programa elaborado e a execução de *debug* das variáveis de memória, utilizando o programa VisualDSP++ referido no Subcapítulo 3.4.1.

O DSP também se encontra ligado ao mesmo computador, mas a uma placa de vídeo através de um cabo de vídeo composto de forma a permitir a visualização do *output* de vídeo do DSP, utilizando um programa para a visualização do mesmo no computador – Presto PVR.

Esta ligação encontra-se exemplificada na Ilustração 62 pela designação de *Vídeo Out*.

Também como saídas do DSP, encontram-se as ligações aos três controladores dos motores do dirigível.

De forma a permitir, como referido acima, a ligação simultânea dos *outputs* dos motores para o sistema simulado e para o sistema real, as saídas de *output* do DSP foram duplicadas na própria placa, como se pode verificar na Ilustração 62, pois esta possuía saídas de áudio suficientes que permitiram a duplicação.

A ligação para a placa de aquisição é feita por cabos de vídeo composto (RCA-RCA) e convertidos em BNC na ligação à placa propriamente dita.

De forma a facilitar a montagem do sistema foi atribuída uma cor a cada saída da placa DSP para os controladores (Ilustração 62), e correspondente entrada de cada controlador para a placa de aquisição (Ilustração 67).

Como ligação de entrada do DSP encontra-se a ligação de *Vídeo In*, correspondente à ligação da câmara de bordo.

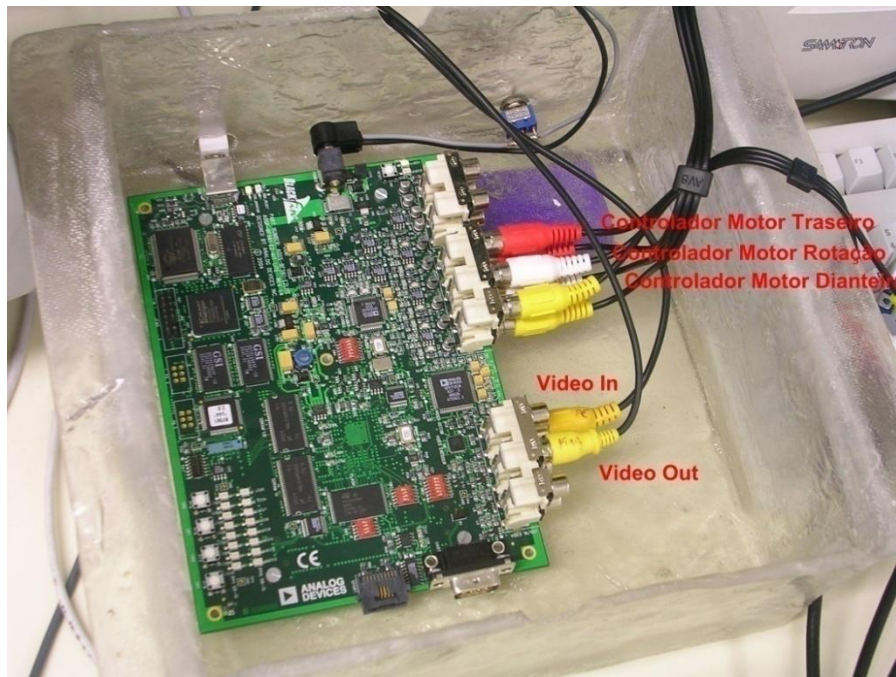


Ilustração 62 – Ligações de vídeo e dos controladores do DSP

Em relação ao **Computador de Trabalho** (Pentium IV 3,0 GHz, com 1GHz de RAM e WinXP), necessitou de uma placa de aquisição de imagem (placa de TV), cuja função é receber a imagem de *output* do DSP, permitindo a sua visualização para fins de *debug* ou simplesmente para se visualizar o funcionamento do algoritmo implementado e correspondente movimento do dirigível.

Neste computador foi também instalado o programa LabView descrito no Subcapítulo 4.4 que interliga a placa de aquisição do Subcapítulo 4.3 (com conexão USB) ao simulador UsarSim que se encontra instalado no computador de simulação (ligado através de uma conexão TCP/IP entre os dois computadores).

Por fim, foi também instalado neste computador o programa VisualDSP++ cuja função se encontra descrita no Subcapítulo 3.4.1 e que conecta directamente a placa DSP através de um cabo USB.

Por último, o **Computador de Simulação com UsarSim** (Pentium IV 2,4GHz, com 1GHz de RAM, placa gráfica ATI de 512MHz de RAM e WinXP) recebe como entrada a ligação TCP/IP vinda do Computador de Trabalho, e como saída o envio da imagem simulada da câmara do dirigível para a placa DSP através de uma placa de TV que foi instalada.

Tal como a designação do computador indica é neste computador que se encontra instalado o simulador utilizado, UsarSim, que se encontra explicado no Subcapítulo 4.2.

Na Ilustração 63 é possível visualizar-se a montagem em laboratório do ambiente de trabalho onde, da esquerda para a direita, se encontram:

- O Portátil de Trabalho, para programar o DSP em ambiente de teste real;
- O Computador *desktop* de Trabalho, utilizado para programar o DSP em ambiente de simulação;
- O *cockpit* traseiro com o DSP incorporado;
- O Computador extra usado para controlar o ATRV Jr em ambiente de simulação;
- O Computador de Simulação com UsarSim que corre o simulador para simulação do conjunto;
- Em cima da bancada é visível o *cockpit* dianteiro do dirigível, onde se encontra acoplado a haste de rotação que suporta os motores dianteiros. Por detrás deste *cockpit* encontra-se a asa do dirigível que suporta o motor traseiro.

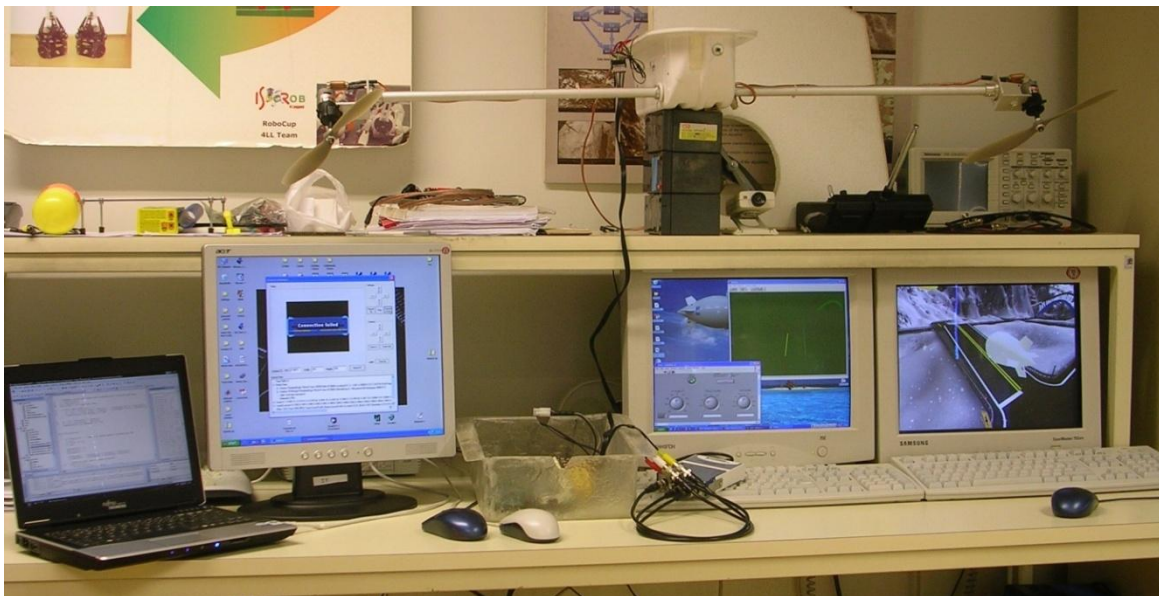


Ilustração 63 – Bancada de trabalho em laboratório

De modo a aumentar a portabilidade de todo o sistema aquando da transição para o real preparou-se um **Portátil de Trabalho** com o *software* de programação do DSP, o VisualDSP++, e adquiriu-se uma placa de TV externa possibilitando a visualização da imagem de saída do DSP que é transmitida através de um sistema de transmissão de rádio frequência.

O esquemático das ligações necessárias para o uso do dirigível em ambiente real encontra-se na Ilustração 64.



Ilustração 64 – Esquemático das ligações do ambiente real

Como se pode verificar através do esquema só é necessário um portátil de trabalho e o dirigível real, para se testarem os algoritmos implementados no DSP.

No que diz respeito ao dirigível este sofreu algumas alterações ao nível das ligações e sistemas electrónicos *onboard*, como foi referido ao longo deste relatório.

O **esquemático das ligações efectuadas no dirigível** encontra-se na Ilustração 65, na qual se demonstra a separação da electrónica pelos dois cockpits acoplados ao dirigível e respectiva ligação entre eles.

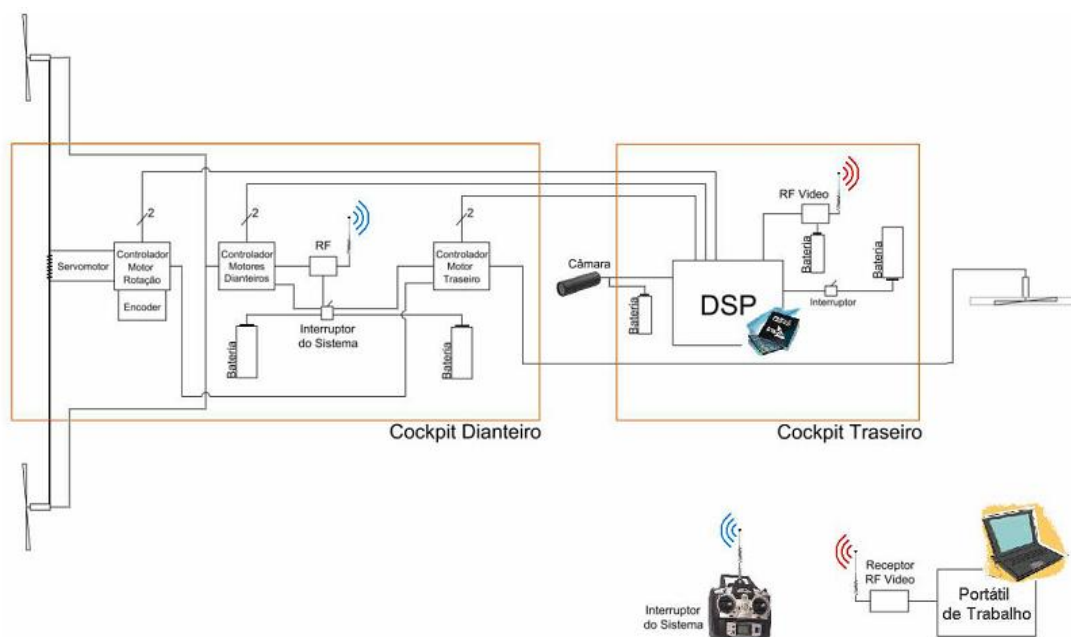


Ilustração 65 – Esquemático das ligações do dirigível

Como o dirigível irá funcionar a uma altitude de aproximadamente 5 metros no ambiente de testes e de modo a prevenir acidentes derivados de falhas de baterias ou de programação do algoritmo do DSP durante a execução de testes foi implementado um “Interruptor do Sistema” que se encontra identificado na Ilustração 65 e que corta a alimentação das baterias que alimentam os controladores dos motores, caso seja accionado através de um interruptor existente no antigo comando RF do dirigível, e que foi desactivado devido ao facto do dirigível passar a ser autónomo.

Desta forma, o utilizador poderá desligar todos os motores do dirigível, em caso de necessidade, sem recorrer directamente ao dirigível, garantindo mais segurança em caso de falha.

4.2 Simulador USARSim

No desenvolvimento de aplicações robóticas, uma das fases determinantes para o sucesso da aplicação é sem dúvida a criação de um simulador realista que teste os seus comportamentos em ambientes virtuais oferecendo ao investigador a facilidade de teste dos códigos desenvolvidos para o controlo dos robots, permitindo fazerem-se de modo simples as calibrações e ajustes necessários para o óptimo funcionamento de todo o conjunto retornando assim a transição para o robot real de forma mais directa e quase sem alterações.

Desta forma, foi elaborado um simulador próprio para o robot PASSAROLA, tendo sido escolhido para o efeito o simulador UsarSim – *Urban Search and Rescue Simulator*. Este simulador é baseado no motor de jogo *Unreal Tournament 2004* da *EpicGames* e é o simulador mais utilizado por todo o Mundo para simular novos robots.

Este simulador mostrou-se essencial no desenrolar deste projecto permitindo minimizar os tempos e custos de testes, e ofereceu um ambiente de trabalho importantíssimo para futuros projectos baseados em dirigíveis.

Em [5] encontram-se todos os estudos efectuados para a criação deste simulador, bem como a explicação de todos os detalhes que se considerou de modo a implementar um dirigível simulado o mais próximo da realidade.

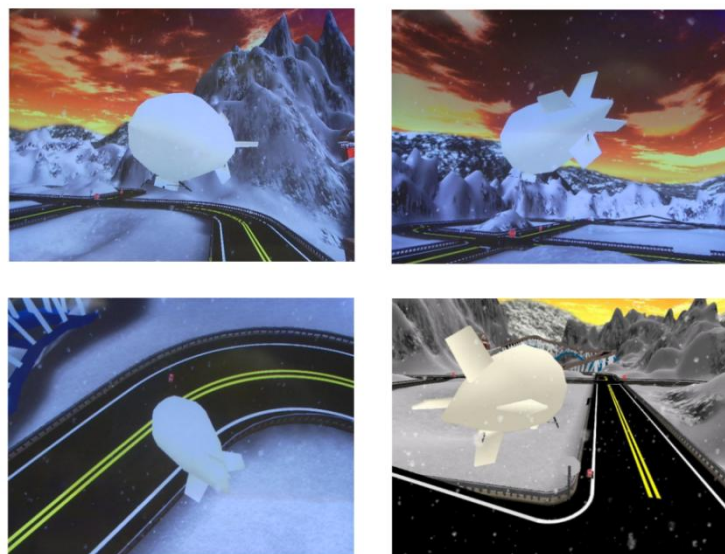


Ilustração 66 – Imagens do dirigível simulado

4.3 Placa de aquisição

De forma a conectar a placa de processamento do dirigível ao simulador elaborado, foi necessário adquirir e utilizar uma placa de aquisição da marca *National Instruments*, modelo NI USB-9233, cuja função será de seguida explicada,

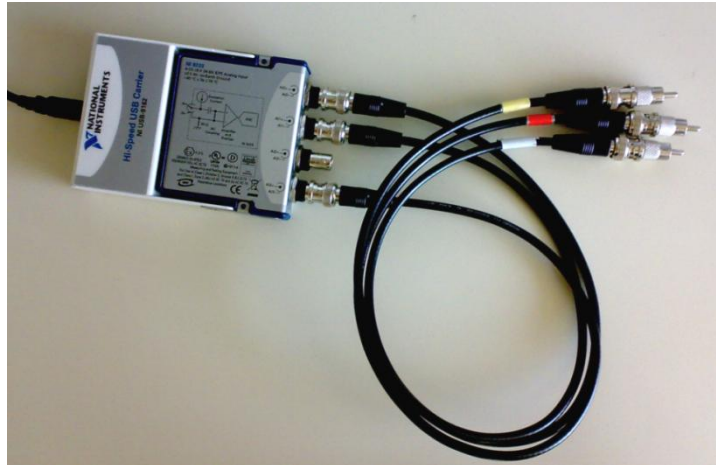


Ilustração 67 – Placa de aquisição da *National Instruments*

Esta placa tem como funcionalidade a conversão instantânea do sinal analógico à entrada em sinal digital à saída transformando assim os sinais de áudio emitidos pelo DSP para os motores em sinais reconhecíveis pelo *software* LabView instalado no computador de trabalho.

Esta transformação dos sinais permite ao programa LabView receber os sinais enviados para os motores do dirigível e enviá-los via TCP/IP para o simulador.

Como se pode visualizar na Ilustração 67 as entradas da placa são em formato BNC, cujas conexões foram transformadas em conexões RCA, de forma a tornar possível ligá-las ao DSP. A saída é digital via ligação USB.

4.4 Aplicação em Labview

O *software* de programação LabView, é um *software* de programação gráfica da *National Instruments*, que usa ícones em vez de linhas de texto para criar aplicações.

O código deste *software* é organizado como um diagrama de blocos onde o fluxo de dados determina a forma de execução do algoritmo implementado.

Foi elaborado um programa capaz de interpretar os sinais emitidos pelo DSP correspondentes aos ciclos de trabalho enviados para os motores reais do dirigível, transformando esses sinais em valores

contínuos correspondentes aos valores de velocidade simulada dos motores e enviando-os via TCP/IP para o computador responsável pela simulação do dirigível o qual contém instalado o simulador UsarSim (Subcapítulo 4.2).

O programa elaborado oferece o interface gráfico da Ilustração 68 que permite controlar os motores manualmente, alterar o computador de simulação e visualizar tanto os ciclos de trabalho recebidos do DSP bem como o ultimo comando enviado para o simulador.

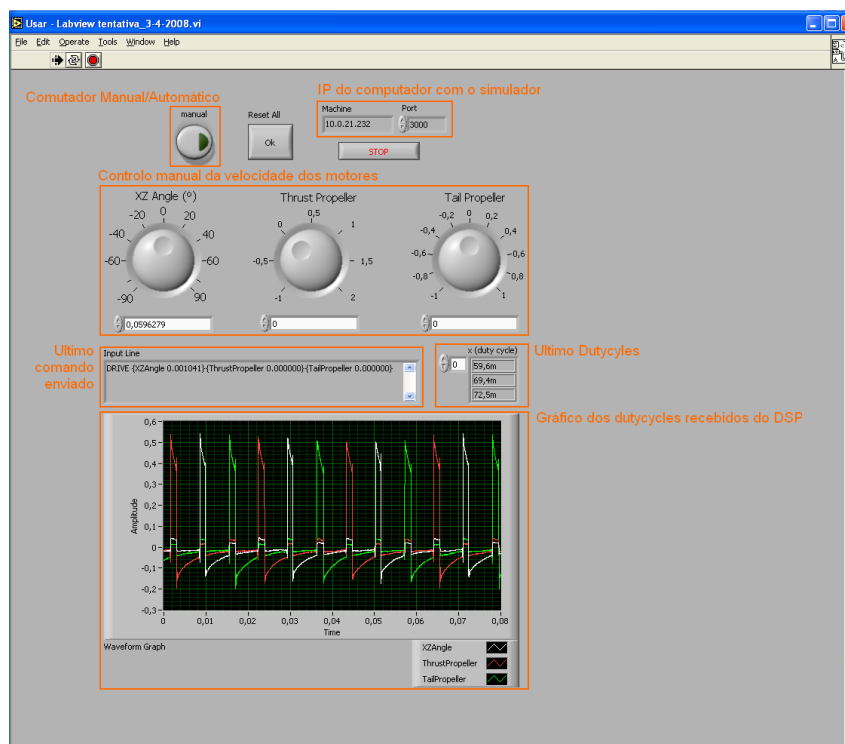


Ilustração 68 – Interface gráfico do software LabView

Com auxílio desta ferramenta foi possível simular o controlo do dirigível no simulador que se elaborou utilizando o DSP e todo o código implementado neste como fonte dos sinais de entrada do programa LabView, permitindo desta forma o teste dos algoritmos implementados no DSP sem ser necessário recorrer ao dirigível propriamente dito.

O *output* deste programa é um comando enviado para o simulador através de uma comunicação TCP/IP com dois tipos de formatos apresentados na Tabela 5.

Estes comandos são enviados regularmente para o simulador controlando o seu movimento.

Tabela 5 – Comandos enviados para o simulador

Inicialização do robot	INIT {ClassName <i>robot_class</i> } {Name <i>robot_name</i> } {Location <i>x,y,z</i> } {Rotation <i>r,p,y</i> }
Accionamento de movimento ao robot	DRIVE {AltitudeVelocity <i>float</i> } {LinearVelocity <i>float</i> } {LateralVelocity <i>float</i> } {RotationalVelocity <i>float</i> } {Normalized <i>bool</i> }

4.5 Calibrações, Testes e Experiências

Calibração do simulador

De forma a usufruir do simulador elaborado no programa UsarSim de modo a que a transição para o real seja feita com o mínimo de alterações foi necessário calibrar todos os parâmetros do mesmo tais como, o tempo de aceleração, o momento de inércia no arranque, paragem e rotação.

Para tal foram efectuados diversos testes com o dirigível real para cada uma das variáveis.

Estes testes consistiram em programar o dirigível com movimentos simples durante um período de tempo e através da imagem obtida pela câmara incorporada e das marcações no chão foi possível medir os tempos de cada movimento.

Foram elaborados diversas experiências para cada movimento do dirigível testado com o intuito de se obterem os valores de calibração mais fidedignos com a realidade.

Após a execução dos diversos testes foi feita uma análise aos resultados de forma a anular os que se desviavam em demasia dos valores obtidos, fruto de uma perturbação ambiente durante o decorrer do teste.

De seguida foi feita uma média dos resultados dos testes obtendo-se os valores finais, descritos através da relação *dutycycle* vs velocidade, que se encontram nas Tabelas 6 e 7, e nas Ilustrações 69 e 70 os respectivos gráficos dos resultados.

Tabela 6 – Resultados dos testes de velocidade

<i>Dutycycle</i> dianteiro	Velocidade em X (m/s)
50	-1,03
60	-0,62
68	0
70	0
72	0
77	1,02
80	1,26
82	1,47
88	1,90
95	2,33

Tabela 7 – Resultados dos testes de rotação

<i>Dutycycle</i> traseiro	Velocidade rotação (rad/s)
60	-0,87
65	-0,67
70	-0,42
72	0
73	0
74	0
77	0,31
84	0,56
90	0,71

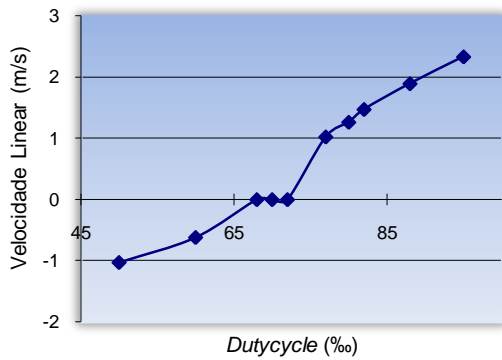


Ilustração 69 – Velocidade linear do dirigível em X vs *duty cycle* dos motores dianteiros

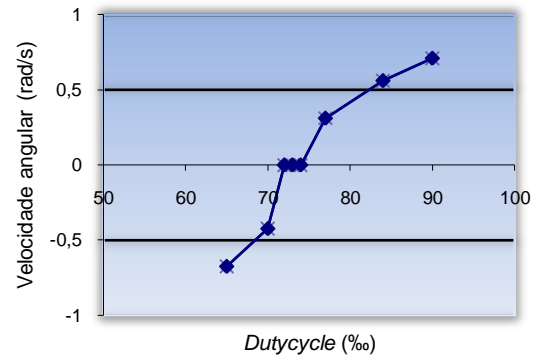


Ilustração 70 – Velocidade angular do dirigível vs *duty cycle* do motor traseiro

Com o auxílio da ferramenta existente no programa MatLab conhecida como *Curve Fitting* e as tabelas de resultados obtidos das experiências realizadas foi possível obter, como retorno da ferramenta utilizada, as equações de segunda ordem correspondentes às funções descritas pelos pontos resultantes.

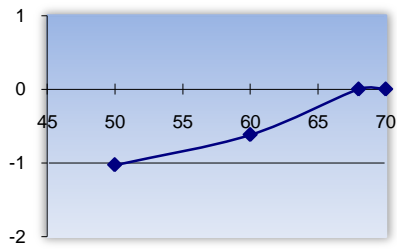
De forma a calcular a equação da função desejada do modo mais aproximado possível, foi feita uma divisão da função em três outras funções:

- a função do movimento no sentido horário do motor;
- a função do movimento contrário do motor;
- a função correspondente à “zona morta” do movimento do motor.

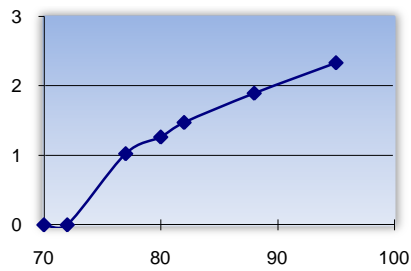
Sabendo que a função correspondente à “zona morta” seria implementada através do envio do valor 0 de *duty cycle*, só se apresentam as equações referentes às outras duas funções de cada um dos motores – Tabelas 8 e 9 –, onde x corresponde ao valor de *duty cycle* enviado pelo programa e y ao valor da velocidade do motor.

Tabela 8 – Equações de segundo grau do funcionamento dos motores dianteiros

Motores Dianteiros



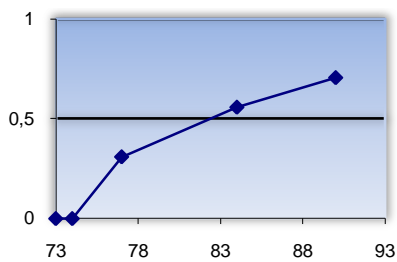
$$0,002798 \times x^2 + (-0,2668) \times x + 5,315 = y$$



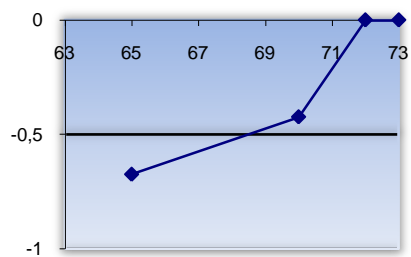
$$(-0,004184) \times x^2 + 0,8007 \times x + (-36,03) = y$$

Tabela 9 – Equações de segundo grau do funcionamento dos motores dianteiros

Motor Traseiro



$$(-0,00298) \times x^2 + 0,5341 \times x + (-23,24) = y$$



$$-0,005924 \times x^2 + (-0,7162) \times x + 20,8 = y$$

Com estas funções foi possível obter uma aproximação à realidade de todos os pontos não testados que se incluem dentro do intervalo de valores experimentados.

Após a obtenção das equações que descreviam o movimento do dirigível, estas foram implementadas no *software* LabView utilizado para enviar os comandos para o simulador possibilitando assim o teste de todos os algoritmos desenvolvidos para o DSP no simulador de forma mais realista.

Foram repetidos os testes reais no simulador e através da comparação visual entre os dois vídeos resultantes, o vídeo real e o do simulador, verificou-se a qualidade dos valores medidos.

Com esta calibração, foi possível fazer todos os ajustes dos controladores e algoritmos de forma rápida e simples, não tendo que recorrer ao sistema real, poupando inúmeras horas de experiências.

Calibração dos vermelhos

Para a calibração da gama de vermelhos que a luminosidade do local de testes oferece foi utilizado o mesmo processo descrito no Subcapítulo 3.2.2. Foram retiradas diversas imagens, de vários ângulos, ao robot ATRV Jr, em diferentes horas ao longo do dia, através da ligação directa do *input* ao *output* de vídeo do DSP. Seguidamente processaram-se essas imagens utilizando o programa em MatLab elaborado para o efeito, encontrando-se facilmente os *thresholds* das componentes YUV dos pixéis, adaptadas às características do local de trabalho.

Este método de pesquisa das gamas de cor permite uma rápida adaptação a qualquer local de trabalho e inclusive a uma mudança de cor que se pretenda seguir.

5 | RESULTADOS

5.1 Resultados em Ambiente de Simulação

Após as calibrações do ambiente real com o ambiente simulado, elaboraram-se diversos testes de modo a confirmar-se o correcto funcionamento de todo o sistema entre os quais se testou o comportamento do dirigível para situações extremas como, variações bruscas de movimentos, velocidades de movimento e altitudes de trabalho elevadas, proporcionando imagens difíceis de processar.

De forma a se simular a presença de possível ruído na imagem recebida no DSP criou-se no ambiente de simulação um contínuo e aleatório “nevão”, cujos flocos servem para interferirem no processamento da imagem recebida, obrigando o algoritmo a superar as interferências.

Após os testes dos algoritmos de seguimento de linhas e do robot ATRV Jr, verificou-se que o sistema completo funciona com bastante qualidade, mesmo quando o dirigível foi sujeito a situações mais extremas, como curvas com ângulos de 90° no seguimento de linhas e em casos de rampas cuja linha a seguir aumenta e diminui de largura devido à alteração da distância entre a linha terrestre e o dirigível.

No caso do dirigível se encontrar a seguir o robot terrestre, aquele supera facilmente as situações inesperadas, tais como mudanças de direcção por parte do ATRV Jr, ou em situações de rampa com a mesma problemática do caso das linhas.

De seguida são apresentadas algumas das experiencias efectuadas para teste do sistema completo, primeiro no Seguimento do robot ATRV Jr – Experiencias 1, 2 e 3 –, e de seguida no Seguimento de Linhas – Experiencia 4.

Experiencia 1:

Na Ilustração 71 encontra-se apresentada uma experiência de seguimento do robot ATRV Jr numa trajectória em forma de U onde ambos os robots partem da posição de repouso no instante $t = 0s$. A velocidade de seguimento desta experiência é de $0,58m/s$ e o dirigível opera a uma altitude de $6m$. Os gráficos apresentados são de Posição 3D e Posição/Orientação *versus* Tempo, durante a trajectória de teste de cerca de $240m$ em $400s$, denotando-se em todo o percurso um pequeno atraso temporal do dirigível em relação ao robot terrestre, derivado ao controlo do PASSAROLA ser efectuado em relação ao centro da imagem e não ao centro de massa do robot, de onde é retirado o valor da posição do robot no simulador.

Verifica-se que o seguimento da trajectória efectuada pelo robot ATRV Jr, é executado com sucesso pelo dirigível.

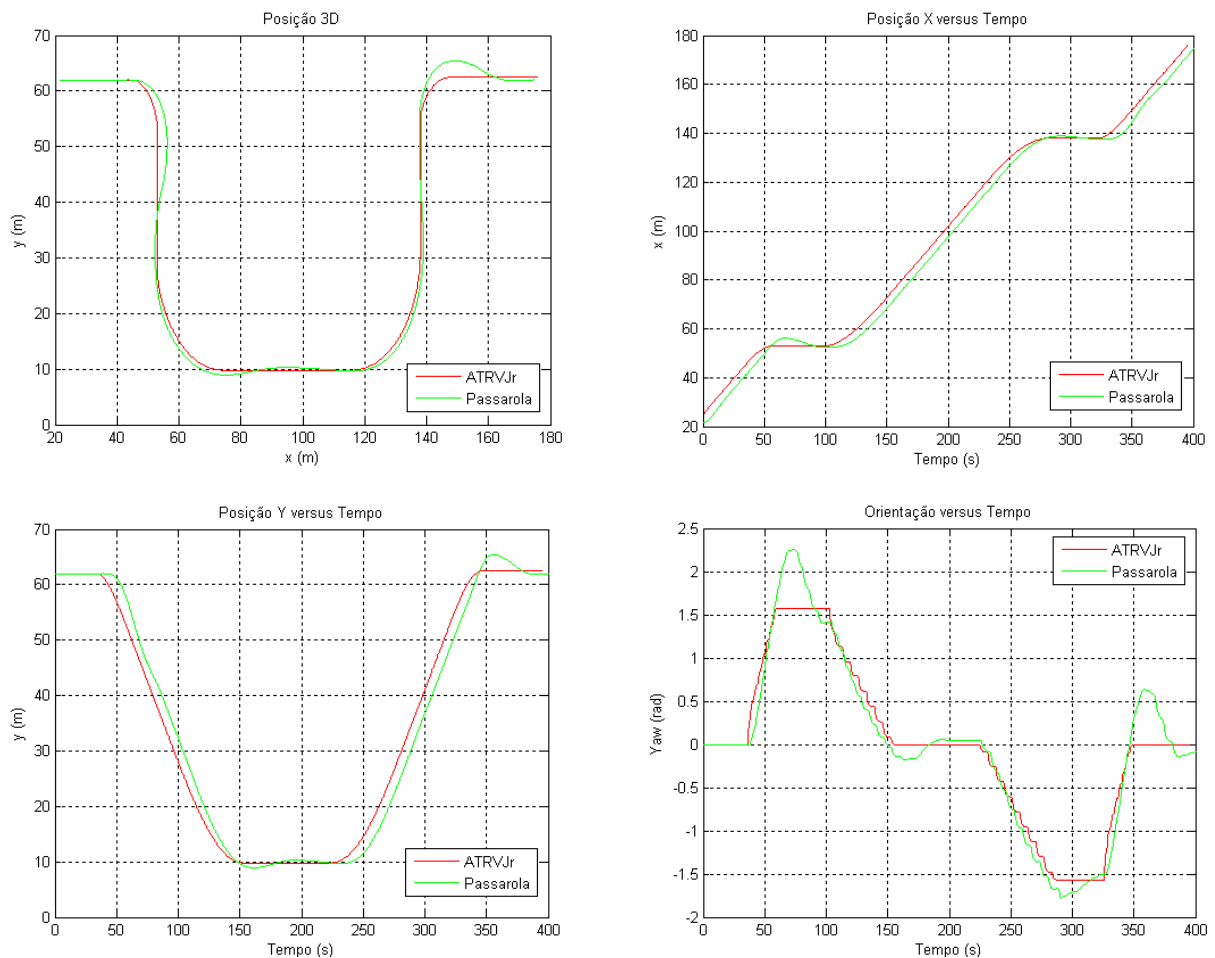


Ilustração 71 – Trajectória em forma de U, realizada com $V = 0,58m/s$ e a uma altitude de $6m$

Experiencia 2:

Nesta experiência realizou-se uma inversão de marcha ao robot ATRV Jr, a meio da realização de uma trajectória rectilínea segundo x , com uma velocidade de $0,58\text{m/s}$. Ambos os robots iniciam a experiência em $t = 0\text{s}$ a partir das suas posições de repouso. A atitude de operação do dirigível é novamente 6m .

Como se pode verificar através da visualização da Ilustração 72, o sistema de controlo implementado no dirigível é capaz de reagir e recuperar a movimentos bruscos efectuados pelo robot terrestre.

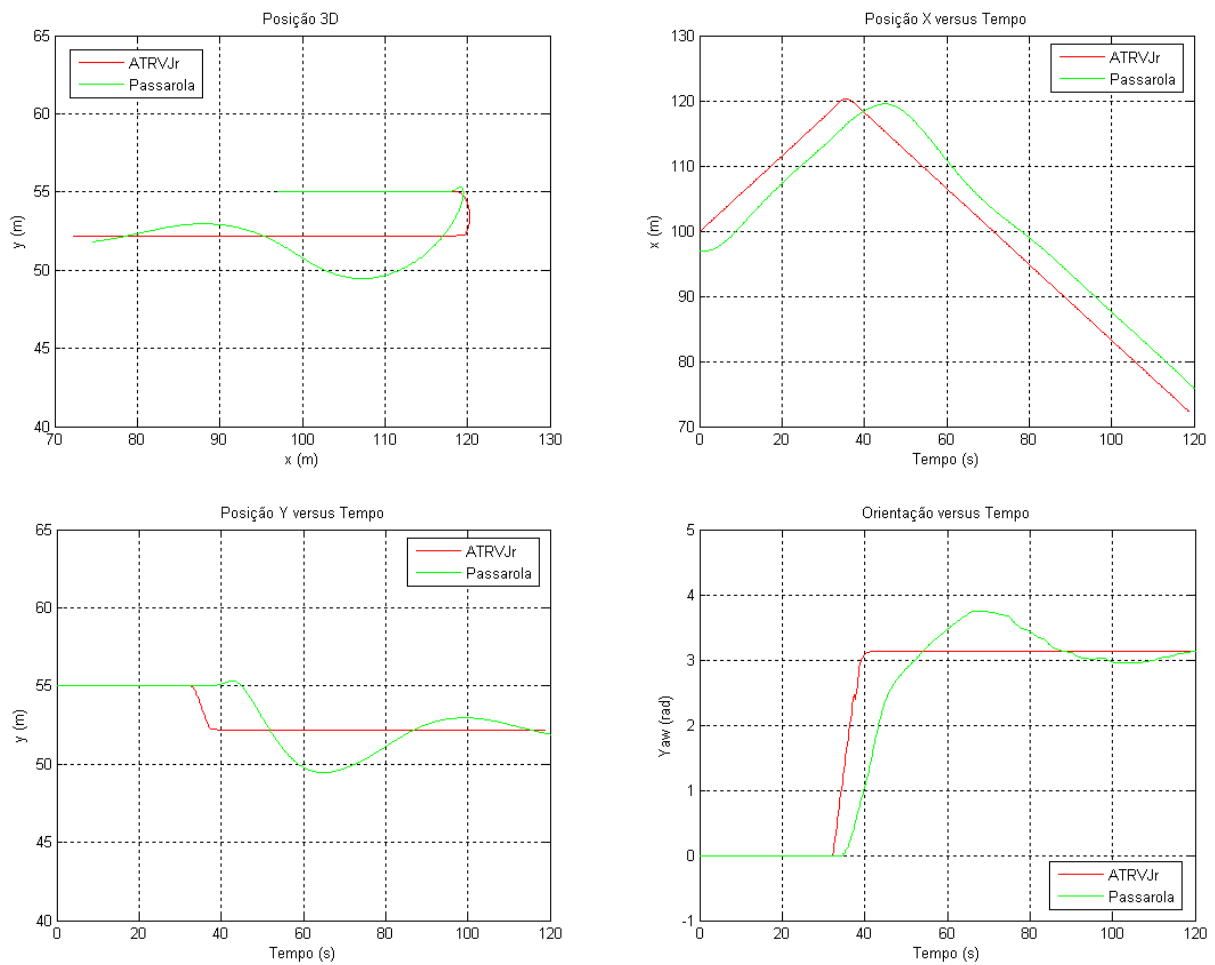


Ilustração 72 – Trajectória de inversão de marcha, realizada com $V = 0,58\text{m/s}$ e uma altitude de 6m

Experiencia 3:

O percurso da Experiencia 3 é um circuito fechado com o comprimento de 700m, efectuado a uma velocidade de 0,78m/s. Mais uma vez, os robots partem da posição de repouso, sem velocidade inicial e no instante $t = 0s$. A duração da experiencia é de 900 segundos.

A Ilustração 73 confirma a funcionalidade e resposta dos algoritmos implementados, face a um percurso típico e possível do robot ATRV Jr.

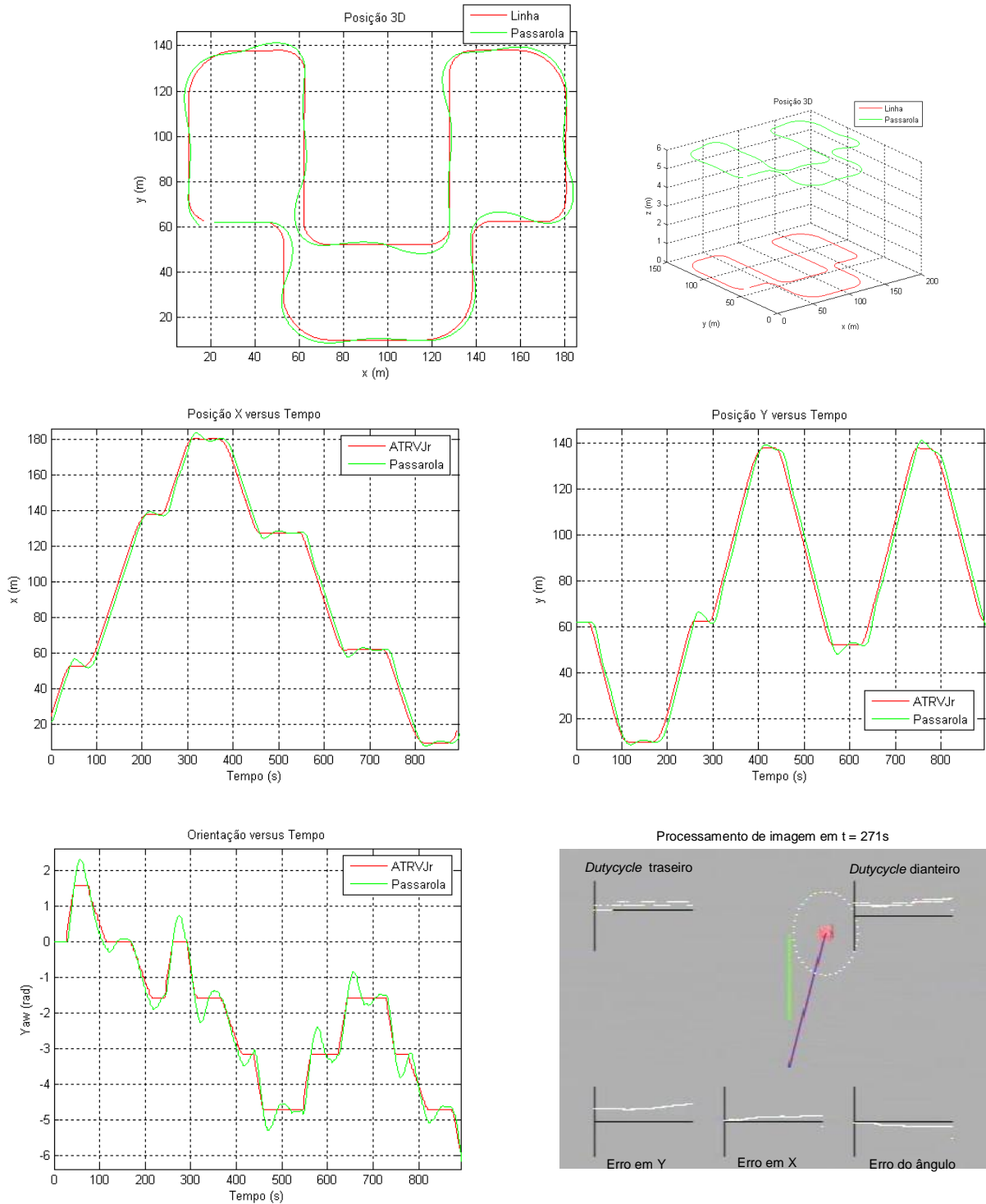


Ilustração 73 – Trajectória fechada realizada com $V = 0,78m/s$ e a uma altitude de 6m

Experiencia 4:

Esta experiência foi efectuada utilizando o algoritmo de Seguimento de Linhas elaborado num circuito fechado com comprimento de 600m e com uma forma que proporciona-se diversas curvas e contracurvas, de modo a se testar o desempenho do algoritmo implementado.

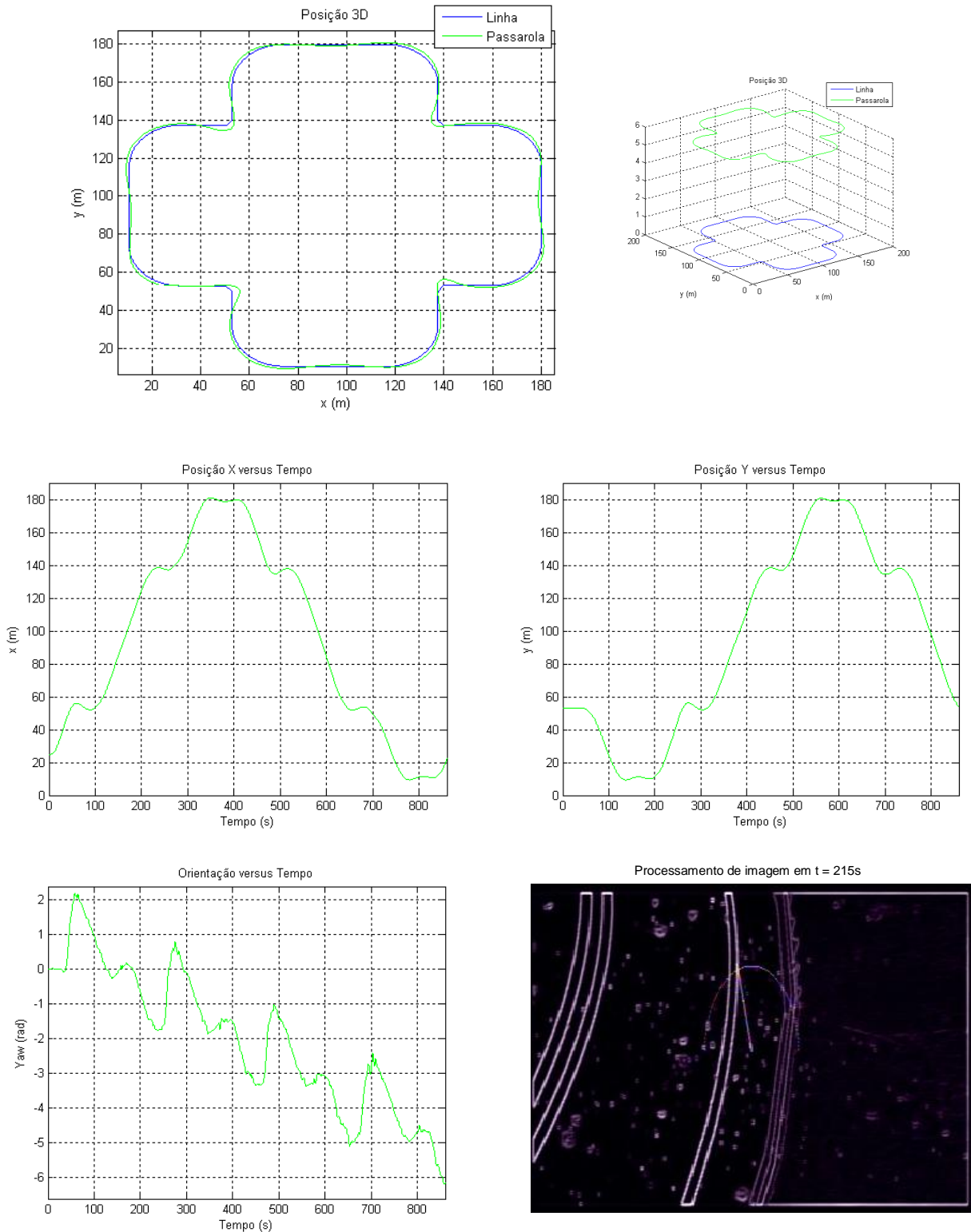


Ilustração 74 – Trajectória fechada realizada com $V_{med} \approx 0,73\text{m/s}$ e a uma altitude de 6m

5.2 Resultados em Ambiente Real

Após o correcto funcionamento do algoritmo implementado em simulação transitou-se todo o sistema para a realidade com pequenos ajustes nas calibrações dos controladores, de forma a otimizar o movimento do dirigível e adaptá-lo às condições do ambiente.

Foram efectuados testes aos dois programas de seguimento desenvolvidos que responderam com sucesso.

Como se esperava o dirigível actual revelou ser um sistema difícil de controlar derivado à sua pequena envergadura, peso e fraca potencia dos motores, tornando-o praticamente imune ao atrito do ar, sendo assim muito sensível a interferências externas, onde pequenos movimentos demoram a equilibrar.

Esta constatação já tinha sido comprovada nas experiências efectuadas pelo colega Thomas Krause, e descritas no seu Trabalho Final de Curso [2].

Ainda foi efectuada uma pesquisa através da Internet com o intuito de conhecer o mercado dos dirigíveis e seus componentes, com o propósito da possível aquisição de motores mais potentes ou até de um dirigível de maior envergadura, tornando possível a sua utilização no exterior, utilizando o GPS como meio de navegação, mas por falta de tempo no desenrolar do projecto e a não necessidade do mesmo para os fins propostos para este trabalho, esta aquisição terá que ficar para uma futura continuação do trabalho.

Outro problema que se encontrou advém do facto da dinâmica do dirigível ser imensamente influenciada pelo volume de hélio inserido no envelope, e que não se mantém ao longo de um dia de trabalho devido à permeabilidade do envelope, alterando consideravelmente o seu movimento, obrigando a pequenos reajustes ao nível do contrapeso da força de impulsão oferecida pelo hélio, e em casos mais discrepantes é necessário reajustar alguns valores do controlador.

Verificou-se que a visualização do *output* de imagem recebida pelo portátil de trabalho, onde se inclui o sistema de gráficos implementado foi vantajosa para o desenvolvimento do projecto, principalmente durante a calibração dos controladores, e oferece uma percepção das decisões e movimentos do dirigível de forma mais simples.

O interruptor do sistema também se concluiu que oferece segurança ao sistema e possibilita o controlo do dirigível em situações mais perigosas.

De seguida apresentam-se algumas imagens dos testes realizados com o sistema global tendo-se testado percursos semelhantes aos testados nas experiências de simulador de forma a se comparar os seus comportamentos.

De notar que derivado à dificuldade de captar as cores vermelhas do robot terrestre em contraste com a cor castanha do chão do pavilhão e pelo facto de existirem poucas zonas vermelhas na vista superior do robot, optou-se por colocar um saco de tecido de cor vermelha em cima do robot ATRV Jr, de modo a melhorar a captação da cor vermelha.

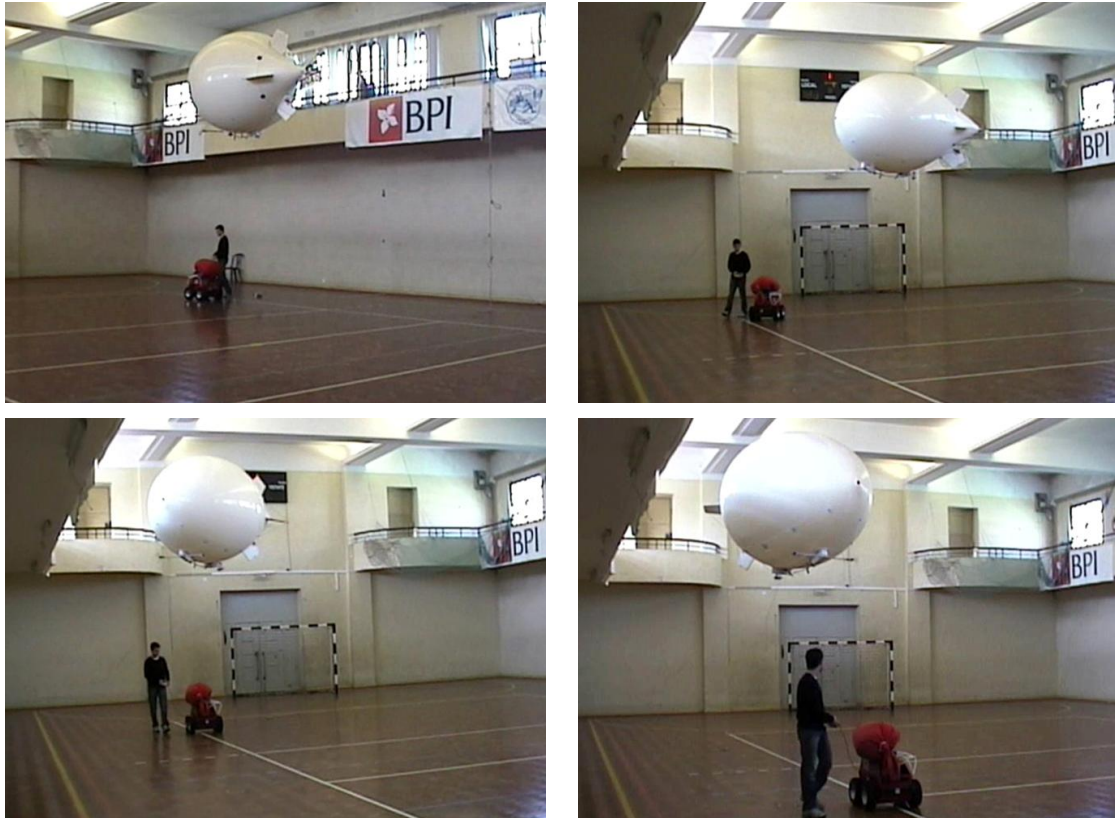


Ilustração 75 – Sequencia de imagens no decorrer de um percurso rectangular

Para as experiências do algoritmo de Seguimento de Linhas, foram construídas linhas com comprimentos e curvas desejadas para o teste do sistema, com o auxílio de uma fita branca. Verificou-se uma certa dificuldade em manter o seguimento correcto da linha derivado à existência de diversas linhas já existentes no chão do ginásio, referentes aos diversos campos, oferecendo assim muitas perturbações na imagem real.

Desta forma conclui-se que o processamento de imagem necessita de ser mais trabalhado, no que diz respeito à rapidez de execução dos cálculos de controlo bem como à robustez na detecção de linhas. De qualquer forma, conseguiu-se realizar experiências de seguimento de linhas que confirmam o funcionamento de todo o sistema.



Ilustração 76 – Sequencia de imagens no decorrer de uma curva

6 CONCLUSÕES

Este projecto permitiu alargar o nosso conhecimento em diversas áreas da Engenharia, principalmente na área de Electrónica e na de Robótica.

Durante o desenrolar do projecto familiarizámo-nos com importantes ferramentas de desenvolvimento de algoritmos de diversas áreas, como os programas LabView para a programação de placas de aquisição, VisualDSP++, para programação de placas de processamento, e o simulador UsarSim, para simulações realistas de robots.

Toda a automação do dirigível através do processamento da imagem captada foi conseguida, bem como o simulador realista elaborado para o desenvolvimento do trabalho.

Projectos Futuros e Utilizações

Para que no futuro este projecto se possa valorizar e evoluir deverá que ser equacionada a aquisição de um dirigível que possua um envelope de maiores dimensões, e em consequência, de motores mais potentes e velozes, para utilização no exterior.

Também o envelope poderá ser em poliuretano ou outro material mas com um tratamento que permita menor permeabilidade e perda do gás de enchimento.

A oportunidade de transitar o funcionamento do dirigível para o exterior permite a inclusão de um GPS a bordo, permitindo a programação de rotas pré-definidas e auxiliar na localização do dirigível no seguimento do robot ATRV Jr, onde ambos poderão comunicar entre si e trocar coordenadas de forma a otimizar os seus movimentos.

Com a utilização do sistema GPS, deixará de ser necessária, para seguimento do robot, a câmara incorporada podendo esta ser aproveitada para outras finalidades.

A incorporação do GPS no dirigível permite também o controlo em altitude do mesmo de forma muito simples.

A este sistema deverá ser incorporado um sensor de medição de distâncias, como por exemplo, um sensor que utilize a tecnologia por infravermelhos ou radiofrequência, que serviria para medir a distância entre o solo e o dirigível, de forma a incluir no controlo a informação da altura ao solo do dirigível.

Outro melhoramento que poderá mostrar-se útil principalmente na utilização exterior de um dirigível de maior dimensão o qual irá ser de mais difícil manobrabilidade, será a introdução de controladores independentes para os motores dianteiros, de forma a minimizar as diferenças de rotação entre as

duas hélices, oferecendo auxílio nas manobras de viragem e possibilitando o reajuste de direcção de modo a compensar forças externas, através do accionamento de maior ou menor rotação em cada um dos motores dianteiros, independentemente.

Uma área que oferece vantagem na utilização do dirigível, será a vigilância de incêndios em zonas florestais vastas, pois permite devido à sua altitude de trabalho, cobrir uma vasta área, com a vantagem de não ser poluente, ruidoso e ser económico em termos energéticos.

De forma a aumentar a autonomia do dirigível, é possível incorporar painéis solares, de preferência flexíveis, os quais poderão revestir todo o envelope de forma a gerar energia, aumentando o seu tempo de voo.

Para utilização do dirigível na detecção de focos de calor correspondentes a incêndios será necessário incluir *onboard* um sensor de calor, como por exemplo uma câmara de vídeo com detecção por infravermelhos, e um módulo emissor o qual enviará para a base o alerta de incêndio e a sua localização.

Outra área de aplicação do dirigível poderá ser em aplicações militares e de segurança, ou para transmissão de imagem de acontecimentos, como por exemplo desportivos, desfiles e festivais, sendo necessário apenas incluir uma câmara com maior definição e com estabilizador de imagem, a qual enviará vídeo através de um emissor RF.

Referências Bibliográficas

- [1] João Costal: *Formações de robot aéreos e terrestres*, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisboa, Portugal, 1 de Novembro de 2004
- [2] Thomas Krause: *Description of an aerial robot and the position control library*, Institute of Systems and Robotics (ISR), Lisboa, Portugal, 2004
- [3] Filipe Metelo, Luís Campos: *Vision based control of an autonomous Blimp (VideoBlimp)*, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisboa, Portugal, Setembro 2003
- [4] IdMind - Engenharia de Sistemas, Lda: *Blimp Zepper 40, with bubble shape for extra lift, Functional Report*, Lisboa, Portugal, Maio 2002
- [5] Ricardo Alcácer: *PASSAROLA – Dirigível Autónomo para Operações de Salvamento (Parte Robótica)*, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisboa, Portugal, Setembro 2008
- [6] ADSP-BF561 EZ-KIT Lite: *Evaluation System Manual*, Revision 2.0, Analog Devices, Norwood, USA, Janeiro 2005
- [7] João Alexandre, Sérgio Almeida: *Controlo de um Dirigível*, Escola Superior Tecnologia, Setúbal, 24 de Outubro de 2003
- [8] VisualDSP++ 4.5 *example: Read me - VideoInEdgeDetection_BF561*, Analog Devices
- [9] VisualDSP++ 4.5 *example: Read me - Talkthrough_TDM_C*, Analog Devices

Referências da Internet

- Site do Projecto RESCUE do Instituto de Sistemas e Robótica – <http://rescue.isr.ist.utl.pt/>
- Site da placa DSP da Analog Devices – <http://www.analog.com/>
- Site de fórum da placa DSP da Analog Devices – <http://www.blackfin.org/>
- Site de fórum da placa DSP da Analog Devices – <http://www.dsprelated.com/>
- Site da placa de aquisição da National Instruments – <http://www.ni.com/>
- Site de motores da marca Maxon – <http://www.maxonmotor.com/>
- Site de electrónica da marca Graupner – <http://www.graupner.de/>
- Site de electrónica para aerodelismo – <http://www.phillsmodels.com/>
- Site de electrónica para aerodelismo – <http://www.flima.com.pt>
- Site de electrónica para aerodelismo – <http://www.mpjet.com/>
- Site de electrónica para aerodelismo – <http://www.modelspot.com/>
- Site de electrónica para aerodelismo – <http://rc-butik.dk/>
- Site com referências sobre aerodelismo – <http://www.mat.uc.pt/~pedro/ncientificos/Ponteiros.html>
- Site sobre o formato de imagem YUV – <http://www.fourcc.org/yuv.php>