



Universidade do Porto

FEUP Faculdade de Engenharia

Versa Robot

Robot Móvel Versátil para Competição em Provas de Robótica

Estágio realizado no ISR –Porto, para a obtenção do grau de Licenciado em Engenharia Electrotécnica e de Computadores pela Faculdade de Engenharia da Universidade do Porto
(Ramo de Automação e Produção Industrial)

Sob a orientação dos Professores Doutores:
Armando Jorge Miranda de Sousa
Paulo José Cerqueira Gomes da Costa

Realizado por:
André Manuel Ferraz Carvalhosa
Tiago Leão Branco Leite



Programa Operacional Ciência e Inovação 2010





Universidade do Porto

FEUP Faculdade de Engenharia

Versa Robot

Robot Móvel Versátil para Competição em Provas de Robótica

Estágio realizado no ISR –Porto, para a obtenção do grau de Licenciado em Engenharia Electrotécnica e de Computadores pela Faculdade de Engenharia da Universidade do Porto
(Ramo de Automação e Produção Industrial)

Sob a orientação dos Professores Doutores:
Armando Jorge Miranda de Sousa
Paulo José Cerqueira Gomes da Costa

Realizado por:
André Manuel Ferraz Carvalhosa
Tiago Leão Branco Leite



Programa Operacional Ciência e Inovação 2010



Apoios:

Os autores foram,
durante a realização deste estágio,
bolseiros do POCI/N/07.01.02/0265/0039/2005

Financiamento FCT/POCI



União Europeia
FEDER

FCT

Fundação para a Ciência e a Tecnologia



Governo da República Portuguesa

Agência Inovação
2010

621.31047.2) / LEEC 2006 / CA20

10741
6 02 10

Em nome de André Carvalhosa:

Gostaria de agradecer aos meus pais e aos meus avós, Sãozinha, Serafim e Rita pelo apoio que prevaleceu nos momentos mais difíceis desta longa aventura.

Ao Tiago pelas horas passadas nos projectos que se transformaram em amizade.

Em nome de Tiago Leite:

Agradeço à minha família todo o apoio e confiança dado ao longo deste ciclo da minha vida.

Nunca vou ganhar um Óscar, pois não tenho o mais pequeno jeito para discursos e agradecimentos... Mas o sentimento está lá...

Em nome dos Versáteis:

Um agradecimento especial ao Prof. Dr. Armando Sousa por desde sempre ter acreditado em nós, pela oportunidade, por todo o apoio que nos deu, por mais uma vez nos fazer sentir que “há uma grande diferença entre saber o caminho e percorre-lo”

Ao Prof. Dr. Paulo Costa, pelas intervenções cirúrgicas que nos fizeram reflectir sobre o melhor caminho.

Ao Prof. Dr. António Paulo Moreira, pela ajuda indispensável na construção do robot.

A todos estes, o nosso muito obrigado pelo companheirismo e confiança depositada.

Ao Eng. Hélder Oliveira pela ajuda que nos deu durante o projecto. Nunca te esqueças, “... o 5DPO é um projecto grandioso...”.

Ao aluno Pedro Machado, o cabeludo, pelo apoio prestado especialmente antes e durante o Robótica 2006 e pela tua paixão por robots. Desejamos-te muito boa sorte nessas andanças.

Ao Pedro Alves, pela ajuda nos dramas do Orcad.

Aos nossos colegas do RIAA, pelo companheirismo. Ana, não te livras de nós...

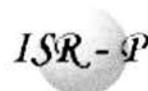
À equipa 5DPO, pelo apoio prestado durante a execução do projecto e principalmente pela oportunidade e experiência de podermos vibrar durante uma semana com um momento que nos ficará para sempre. (Robocup 2006 - BREMEN)

“...Vamos lá pessoal, já só faltam 4 jogos...3...2...1...”

“...Congratulations great chipkick...”

“...Tanto tempo! Para umas compras?...”

agora finalmente a revelação do porque... Perdemos-nos com as Alemãs...



“A melhor forma de se aprender ciência, é acreditar que se é ignorante”
Hippocrates (entre 460 AC e 377 AC)



Resumo

A necessidade de autonomia dos sistemas robóticos incrementa a sua complexidade e o seu projecto cuidadoso torna-se assim ainda mais importante.

Este trabalho apresenta o projecto de um sistema robótico autónomo para navegação num ambiente semiestruturado. A autonomia em robótica, associada à mobilidade, leva à necessidade de localização em Tempo Real.

O sistema de localização é baseado no processamento, filtragem e fusão de diversos dispositivos de sensorização. É feita a caracterização de cada medida e demonstrada a importância da fusão de informação, para reunir toda a informação válida, no processo de decisão.

O conceito de minimização do erro de hometria é apresentado como sendo a base do sistema de localização, adequado ao ambiente semiestruturado. Tal sistema é baseado em visão e permite a localização completa do robot em (x, y, θ) .

Os métodos desenvolvidos para a localização dinâmica no ambiente semiestruturado são validados experimentalmente num ambiente de competição.

Após a análise dos resultados obtidos, discute-se a aplicabilidade do projecto desenvolvido em aplicações reais.

Ao caracterizar-se adequadamente este problema contribui-se para a focalização dos investigadores e desta forma incentiva-se o desenvolvimento científico e tecnológico nesta área.



Abstract

The need for autonomy in robotic systems augments its complexity and makes careful design very important. This work presents a project of an autonomous robotic system to navigate in a semistruturated environment. Robotic autonomy, together with mobility, has the need for Real Time localization.

The localization system is based in the processing, filtering and fusion of all data sensors. Each measurement is characterized and data fusion importance is demonstrated for a correct decision.

The concept of odometry's minimization error is presented as the basis of localization system, useful for structured environment localization. This system is based uniquely in vision and allows for complete (x, y, θ) localization.

The presented methodologies for dynamic localization are tested in competition environment.

After discuss the obtained results, the applicability of the project in real life is also discussed.

The correct formulation of this problem contributes to the focalization of the investigators and for the scientific and technologic development in this area.



Índice

Resumo	5
Abstract	6
Índice	7
Índice de Figuras	10
1. Introdução	12
1.1 Motivação	12
1.2 Contexto	12
1.3 Estrutura do documento.....	13
2. Projecto do sistema Robótico	14
2.1 Análise de Sistemas Robóticos	14
2.1.1 Áreas de projecto	14
2.1.2 Projecto Global	15
2.1.3 Configuração do robot e sua locomoção.....	16
2.1.4 Sistema de Alimentação.....	18
2.1.5 Sistema embebido	19
2.1.5.1 Sistemas Tempo Real.....	19
2.1.5.2 Classificação de Sensores	20
2.1.5.3 Sensores Comuns em Robótica.....	20
2.1.5.4 Actuadores	21
2.1.5.5 Implementação de Subsistemas	21
2.1.5.6 Comunicações Internas	21
2.1.6 Navegação.....	22
2.1.7 Decisão.....	24
2.1.8 Fusão de Informação.....	24
2.2 Análise de requisitos	25
2.2.1 Particularidades do Festival de Robótica2006.....	25
2.2.2 State of art.....	25
2.3 Especificação de requisitos.....	27
2.3.1 Arquitectura do sistema robótico	27
2.3.2 Sistema de Locomoção	28
2.3.3 Sistema de Localização.....	28
2.3.4 Sistema de Processamento de dados e Decisão	28
2.3.5 Funcionalidades do software de controlo.....	29
2.3.6 Comunicação entre sistemas.....	30
2.3.7 Configuração e tipo de locomoção	30



3. Estrutura mecânica	31
3.1 Projecto em CAD	31
3.2 Implementação	32
4. Estrutura electrónica	34
4.1 Sistema de Alimentação.....	34
4.2 Sensores Utilizados	35
4.2.1 Câmaras <i>firewire</i>	35
4.2.2 <i>Encoders</i>	36
4.2.3 Sonares.....	36
4.3 Actuadores.....	37
4.3.1 Modelos dos motores.....	37
4.3.1.1 Registo dos resultados obtidos.....	38
4.4 Drivers.....	40
4.4.1 Controlador PID.....	41
4.4.1.1 Ajustação dos PIDs	41
5. Plataforma de desenvolvimento e validação do software de controlo	44
5.1 Estrutura do Software de controlo.....	46
5.2 Planeamento de trajectórias.....	48
5.2.1 Equações de movimento do robot.....	48
5.2.2 Rotinas de controlo de movimento	48
5.2.2.1 Action_Gotoxy (Xdestino,Ydestino,W,V).....	49
5.2.2.2 Action_FollowLine(Xinicial,Yinicial,Xfinal,Yfinal,W,V).....	49
5.2.2.3 Action_FollowCircle(Xcentral,Ycentral,Raio,W,V)	49
5.2.2.4 Action_Rotate(Wref,Ganho,W,V)	49
5.2.2.5 Action_Stop(W,V).....	49
5.2.2.6 Action_Acelera(Vref,Vout)	49
5.3 Simulador	51
6. Sistemas de Navegação	52
6.1 Sistema de visão	52
6.2 <i>Debayer</i> da imagem capturada	52
6.3 Algoritmo de detecção, localização e orientação de formas	53
6.3.1 Binarização	54
6.3.1.1 <i>Threshold</i> dinâmico	54
6.3.2 Filtro de detecção de transições	55
6.3.3 Algoritmos de detecção de objectos	56
6.3.3.1 Balizamento (<i>Bounding</i>) de objectos	56
6.3.4 Calculo dos momentos centrados da imagem e filtragem de ruído	58
6.3.4.1 Área (momento de ordem 0).....	58
6.3.4.2 Centro de massa (momento de ordem 1)	58
6.3.4.3 Orientação.....	59
6.3.4.4 Filtragem de ruído.....	59
6.4 Decisão e atribuição de formas	60



6.5	Determinação da localização das formas no mundo.....	61
6.6	Algoritmo de actualização do estado do robot no mundo	62
6.7	Algoritmo de processamento do sinal de semáforo.....	63
6.7.1	Afinação de cores.....	64
6.7.2	Protocolo de comunicação Master PC – Slave PC	67
6.8	Hodometria.....	68
6.8.1	Afinação.....	68
6.8.1.1	Análise de erros da hodometria.....	69
6.8.2	Algoritmo de minimização do erro de hodometria	70
6.8.3	Protocolo de comunicação o entre <i>drivers</i> e sistema de controlo	71
6.9	Sonares.....	72
7. Fusão, filtragem de informação e Decisão		73
7.1	Fontes de informação.....	73
7.2	Filtragem e decisão	74
8. Projecto de um sistema de carga rápida		75
8.1	Análise de requisitos	75
8.2	Especificação do sistema de desenvolvido.....	76
8.3	Dimensionamento e implementação	77
8.3.1	Circuito de monitorização e comando.....	77
8.3.1.1	Interface local	77
8.3.1.2	Dimensionamento dos componentes.....	77
8.3.1.3	Interface e controlo remoto	78
8.3.1.4	Protocolo de comunicação entre PC-Microcontrolador.....	78
8.3.2	Circuito de potência	79
8.3.2.1	Condicionamento de sinal.....	79
8.3.3	Controlo do circuito de potência.....	80
8.3.3.1	Software de controlo.....	80
8.4	Verificação e validação experimental.....	80
9. Verificação e validação experimental		81
10. Conclusões		82
11. Referências bibliográficas		83
ANEXO A (Documento de apoio capítulo 3 – Estrutura mecânica)		84
ANEXO B (Documento de apoio capítulo 4 – Estrutura eléctrica/electrónica)		95
ANEXO C (Documento de apoio capítulo 8 – Sistema de carga rápida)		97
ANEXO D (Acrónimos e Siglas)		99
ANEXO E (Ficha Técnica)		100



Índice de Figuras

Figura 1 - Conceitos de base na Mecatrónica.....	15
Figura 2 - Locomoção tipo diferencial.....	16
Figura 3 - Locomoção tipo triciclo.....	16
Figura 4 - Locomoção tipo <i>Ackerman</i>	17
Figura 5 - Locomoção tipo <i>Synchronous Drive</i>	17
Figura 6 - Locomoção tipo omnidireccional.....	17
Figura 7 - Roda omnidireccional.....	17
Figura 8 - Exemplos de um robot Omnidireccional.....	18
Figura 9 - Arquitectura típica de um sistema de navegação.....	23
Figura 10 - Arquitectura geral do sistema robótico.....	27
Figura 11 - Diagrama de blocos do sistema robótico.....	28
Figura 12 - Arquitectura do módulo de decisão.....	29
Figura 13 - Vista em perspectiva do robot.....	31
Figura 14 - Vista de baixo do robot.....	32
Figura 15 - Vista em perspectiva do sistema de polias.....	32
Figura 16 - Esquemas dos sistemas eléctricos do robot.....	34
Figura 17 - Câmara <i>Firewire</i> (DBK 21F04).....	35
Figura 18 - Formato da imagem capturada (Bayer).....	35
Figura 19 - Esquema de um <i>encoder</i> incremental.....	36
Figura 20 - Imagem dos <i>encoders</i> usados (ref. E6A2-CW5C).....	36
Figura 21 - Sonar (ref. SRF08).....	36
Figura 22 - Imagem dos motores utilizados (ref. 5995 da RS).....	37
Figura 23 - Método utilizado para a determinação da planta do motor.....	37
Figura 24 - Gráfico do erro quadrático da aproximação pelo método dos mínimos quadrados.....	38
Figura 25 - Aproximação da resposta dos motores para ordem 3.....	38
Figura 26 - Gráfico do erro quadrático da aproximação pelo método dos mínimos quadrados.....	39
Figura 27 - Aproximação da resposta dos motores para ordem 3.....	39
Figura 28 - Foto do <i>driver</i>	40
Figura 29 - Estrutura do PID implementado no microcontrolador do driver.....	41
Figura 30 - Parâmetros para a afinação do PID na resposta do sistema.....	41
Figura 31 - Gráfico da resposta ao degrau do motor 1.....	42
Figura 32 - Gráfico da resposta ao degrau do motor 2.....	43
Figura 33 - Plataforma de controlo (<i>Main</i>).....	45
Figura 34 - Plataforma de controlo (Processamento de imagem).....	45
Figura 35 - Esquema Alto-nível do processamento dos dados dos sensores e actuação nos motores.....	47
Figura 36 - Esquema geral do Software de controlo.....	47
Figura 37 - Modelo do robot.....	48
Figura 38 - Validação das rotinas de controlo de movimento.....	50
Figura 39 - Simulador do controlo e dos sinais dos dispositivos de sensorização.....	51



Figura 40 - Formato Bayer	52
Figura 41 - Esquema Alto-nível do processamento de imagem para a detecção de formas e respectiva orientação	53
Figura 42 - Histograma de luminosidade da imagem capturada	54
Figura 43 - Exemplo de uma imagem <i>binarizada</i>	55
Figura 44 - Exemplo do algoritmo de detecção de transições.....	56
Figura 45 - Menu com o cálculo de momentos da imagem.....	56
Figura 46 - Imagem com ruído	57
Figura 47 - Estrutura da filtragem de objectos	59
Figura 48 - Imagem do resultado do balizamento (<i>bounding</i>)	60
Figura 49 - Menu de afinação de parâmetros de mapeamento da imagem para coordenadas do mundo	61
Figura 50 - Método de Tsai	61
Figura 51 - Esquema Alto-nível do processamento de imagem do semáforo	63
Figura 52 - Resultados do processamento do sinal de semáforo	64
Figura 53 - Menu de afinação de cores.....	65
Figura 54 - Componentes RGB de uma cor	65
Figura 55 - Menu de afinação de raios	66
Figura 56 - Esquema da comunicação Master-Slave PC.....	67
Figura 57 - Histograma de medidas de translação.....	69
Figura 58 - Histograma de medidas de rotação	69
Figura 59 - Actualização do estado no robot no mundo.....	70
Figura 60 - Esquema alto-nível actuação em caso de detecção de obstáculos.....	72
Figura 61 - Detecção e desvio de obstáculos.....	72
Figura 62 - Esquema da unidade de filtragem e decisão	73
Figura 63- Menu de afinação de parâmetros do filtro	74
Figura 64 - Curvas características de um sistema de carga rápida	75
Figura 65 - Esquema alto-nível do carregador de baterias	76
Figura 66 - Esquemático da interface local	77
Figura 67 - Interface remota de controlo e monitorização	78
Figura 68 - Esquema do circuito de potência	79
Figura 69 - Esquema do circuito de condicionamento	79
Figura 70 - Esquema da máquina de estados desenvolvida para o processo de carga ...	80
Figura 71 - Fotos do Festival de robótica 2006	81
Figura 72 - Troféu do 2ºLugar obtido	81



Capítulo 1

Introdução

1.1 Motivação

Imagine-se num veículo equipado com um sistema de navegação que o simples premir de um botão permite passar para um piloto automático que o leve para casa. Certamente que o desenvolvimento de tal sistema iniciaria uma revolução nos sistemas de transporte, numa perspectiva do aumento da qualidade de vida.

Certamente que no futuro será assim, no entanto tudo começa pela base e por isso o objectivo deste projecto consiste em projectar um robot móvel baseado num PC portátil que seja versátil para poder concorrer pela FEUP em competições de robótica tal como o festival nacional de robótica, especificado neste caso, para a prova de condução autónoma.

1.2 Contexto

Este trabalho está inserido na área de sistemas robóticos, mais especificamente na área dos sistemas autónomos. O desenvolvimento de um projecto deste tipo, implica a implementação dos conhecimentos adquiridos durante a formação académica, uma vez que abrange áreas que começam na modelização do robot e que se estendem até aos sistemas electrónicos, sistemas de controlo, sistemas de actuação, sistemas de sensorização, interligados num sistema embebido controlado por uma plataforma de software.

Acresce ainda que o ambiente da prova de condução autónoma é adequado à validação experimental do projecto, pois conduz a conclusões realistas que podem ser facilmente portadas para outras aplicações. As dificuldades sentidas neste tipo de competições, são uma aproximação às principais dificuldades a ultrapassar para a implementação eficaz de um sistema deste tipo no uso quotidiano.

Os conhecimentos que resultam da investigação nesta área dos sistemas robóticos são hoje em dia aplicados a diversas aplicações da sociedade civil, nomeadamente nos sistemas de auxílio à condução automóvel.



1.3 Estrutura do documento

As diferentes áreas tecnológicas abordadas durante o desenvolvimento do projecto estão organizadas em capítulos, pela ordem que foram abordados.

No capítulo seguinte será feita a análise de requisitos ao sistema a desenvolver, focando em especial os requisitos da prova de condução autónoma em que o robot participará.

O terceiro capítulo será dedicado à estrutura mecânica desenvolvida. Neste capítulo é apresentado o projecto mecânico desenvolvido em CAD e serão justificadas as escolhas dos materiais usados.

No quarto capítulo será apresentada e descrita toda a estrutura eléctrica/electrónica de alimentação e controlo do robot.

O quinto capítulo inclui a explicação do sistema de controlo implementado, onde se inclui a plataforma de software desenvolvida para a simulação e controlo do robot.

O sexto capítulo está focado para os sistemas de sensorização responsáveis pela navegação do robot. Este capítulo será também dedicado à explicação pormenorizada de todo o processamento efectuado aos dispositivos de sensorização do robot, onde se inclui o processamento de imagem, *encoders* e sonares.

No sétimo capítulo será explicado como é realizada a fusão e filtragem da informação proveniente dos dispositivos de sensorização. Neste capítulo será também pormenorizado o método de decisão na actuação no robot.

O oitavo capítulo será dedicado ao projecto de um sistema de carga rápida de baterias, que para além de se aplicar ao robot desenvolvido, será também utilizado para os robots do 5DPO.

O nono capítulo é dedicado aos métodos de verificação e validação do sistema desenvolvido.

No último capítulo é feita uma análise dos resultados obtidos e dos aspectos a melhorar neste projecto como base para o continuo desenvolvimento do mesmo.



Capítulo 2

Projecto do sistema Robótico

2.1 Análise de Sistemas Robóticos

Ao projectar um sistema robótico deve-se ter em atenção a sua forma e tamanho. O robot ao interagir com o meio que o rodeia não o pode danificar nem ser danificado. Devem ser tomadas medidas de segurança como por exemplo prever um sistema para a rápida e fácil imobilização do sistema.

Como qualquer sistema mecatrónico, o robot necessita de manutenção. Na fase de projecto devem ser tomadas opções de forma a minimizar e facilitar as intervenções para a reparação e manutenção. O sistema deverá ser modular e o mais invariante ao hardware possível.

Como qualquer outro projecto de engenharia, terá certamente que se encontrar compromissos que permitam cumprir os orçamentos. O custo da engenharia de pôr todo o conjunto a funcionar deve então ser tido em linha de conta, quer a nível de custo, quer a nível de tempo de projecto.

2.1.1 Áreas de projecto

Após o levantamento de todos os requisitos anteriores, será de admitir que a robótica é uma disciplina abrangendo diversas áreas científico-tecnológicas que atravessam a Mecânica, a Electrotecnia e a Informática. Procurando detalhar as áreas mais importantes para a robótica, pode-se identificar:

- **Mecânica**
 - Sistemas de tracção
 - Projecto mecânico:
 - *Chassis* e disposição dos subsistemas
 - Arrumação e acessibilidade aos sistemas internos

- **Electrotecnia e Informática**
 - Sistemas de alimentação (ex: baterias)
 - Sensores, actuadores e manipuladores eléctricos
 - Electrónica:
 - Analógica (ex: condicionamento de sinal)
 - Digital (ex: sistemas de interface)
 - De potência (ex: comando de actuadores)



- Comunicações:
 - Internas (ex: redes de sensores e actuadores)
 - Plataformas de processamento:
 - Sistemas Tempo Real
 - Teoria dos sistemas e controlo de sistemas dinâmicos
 - Processamento de sinal e de imagem
 - Sistemas Operativos e ferramentas informáticas
 - Decisão, Controlo e Inteligência Artificial

Algumas áreas pertencem claramente à mecânica ou à electrotecnia ou à informática. Outras áreas porém, atravessam diversos domínios e poderão ser incluídas como questões onde o projecto deve ser partilhado de tal forma que a melhor solução para o conjunto seja atingida. Estas interações são representadas de forma simplificada no próximo tópico.

2.1.2 Projecto Global

Uma das principais dificuldades do projecto robótico actual é a diversidade de áreas a abordar.

Generalizando, pode então dizer-se que o projecto de sistemas multidisciplinares beneficia da abordagem mecatrónica onde é dada elevada importância ao projecto integrado e à interacção entre diferentes áreas.

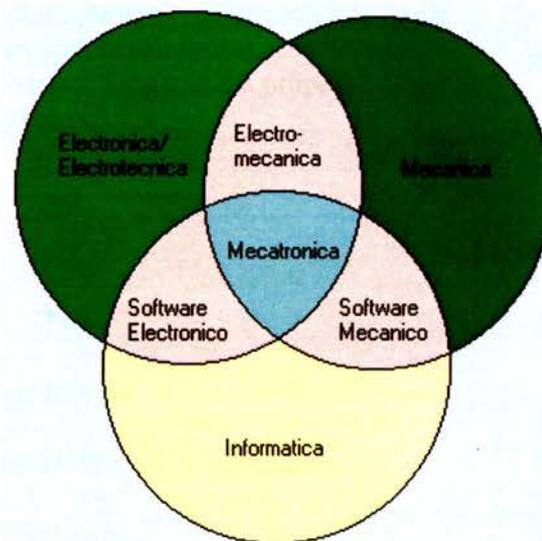


Figura 1 - Conceitos de base na Mecatrónica

A Mecatrónica é um área técnica multidisciplinar baseada na engenharia mecânica clássica e na engenharia electrónica/electrotécnica, estando estas duas interligadas pela ciência da computação e a engenharia do software.



Dentro das áreas dos computadores, do controlo e da electrónica são essenciais conhecimentos de modelização, processamento e análise de sinais, métodos numéricos, controlo de processos dinâmicos e ainda de comunicações e electrónica.

A área da mecânica diz respeito às respostas dos materiais sobre o efeito de forças e inclui as leis relativas ao movimento.

Uma vez estabelecida a filosofia que o projecto deve ter, estuda-se de seguida vários tipos de configurações para sistemas robóticos.

2.1.3 Configuração do robot e sua locomoção

Existem diversos tipos de robots com elevada variedade de funcionalidades. Uma questão essencial é a forma como o robot se deslocará, isto é, a configuração do seu sistema de locomoção incluindo motricidade e direcção.

Os veículos movidos por rodas convencionais são frequentemente utilizados em sistemas robóticos. Aqui, apenas estes serão fruto de análise, pois os restantes tipos de locomoção não são adequados ao âmbito do projecto. As configurações de locomoção sobre rodas mais comuns são:

- Locomoção diferencial:
 - Duas rodas motorizadas independentemente
 - Um terceiro ponto de apoio dá estabilidade (pode ser uma roda ou apenas um apoio)
 - Permite que o robot rode sobre si próprio (em torno do ponto C)

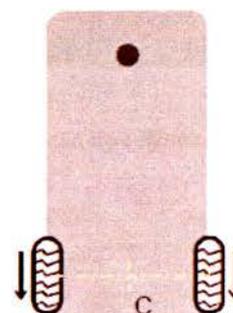


Figura 2 – Locomoção tipo diferencial

- Locomoção tipo triciclo:
 - A mesma roda dá direcção e motoriza o veículo
 - Duas rodas livres atrás
 - Curvatura em torno do ponto C

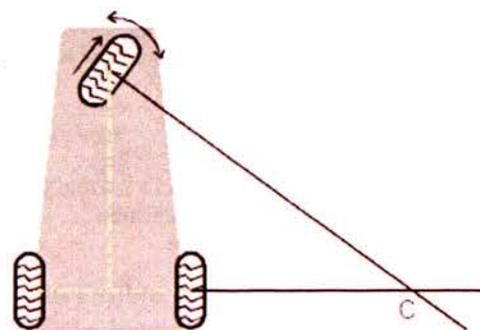


Figura 3 - Locomoção tipo triciclo



- Locomoção *Ackerman*:

- Duas rodas guiam a direcção do carro
- Muitas vezes tracção nas rodas traseiras com diferencial mecânico
- Pode também ter tracção à frente nas rodas de direcção
- Permite apenas raio de curvatura limitado ao robot

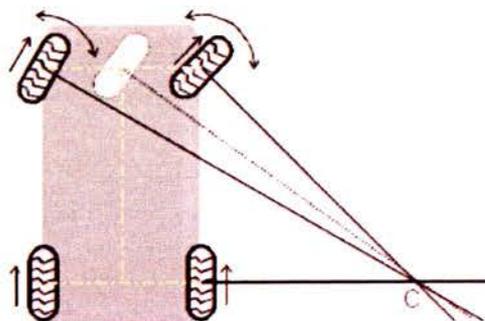


Figura 4 - Locomoção tipo *Ackerman*

- Locomoção *Synchronous Drive*:

- Todas as rodas são motoras e direccionais
- Um conjunto de correias faz girar todas as rodas sincronizadamente
- Um outro conjunto de correias dá tracção a todas as rodas
- Pode mover-se em qualquer direcção após rotação das rodas

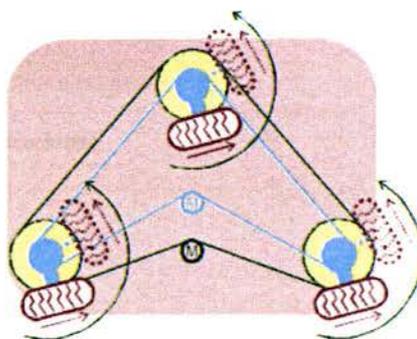


Figura 5 - Locomoção tipo *Synchronous Drive*

- Locomoção omnidireccional

Actualmente, inúmeros sistemas são dotados de rodas omnidireccionais.

As rodas omnidireccionais são constituídas por uma roda maior (principal) e enumeras rodas pequenas no seu perímetro colocadas perpendicularmente.



Figura 7 - Roda omnidireccional

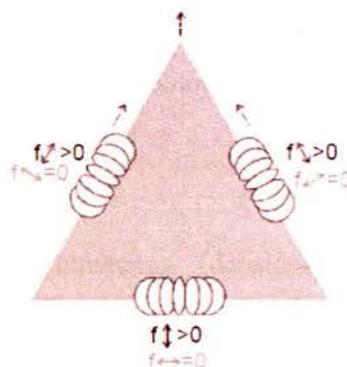


Figura 6 - Locomoção tipo omnidireccional

Estas permitem que o sistema se movimente em qualquer direcção e sobre si mesmo.

São particularmente vantajosas em sistemas cuja necessidade de mudar rapidamente a direcção do movimento é fulcral. Um exemplo de aplicação muito frequente desta tipologia é o futebol robótico.

A figura 8 ilustra um robot com mobilidade omnidireccional.

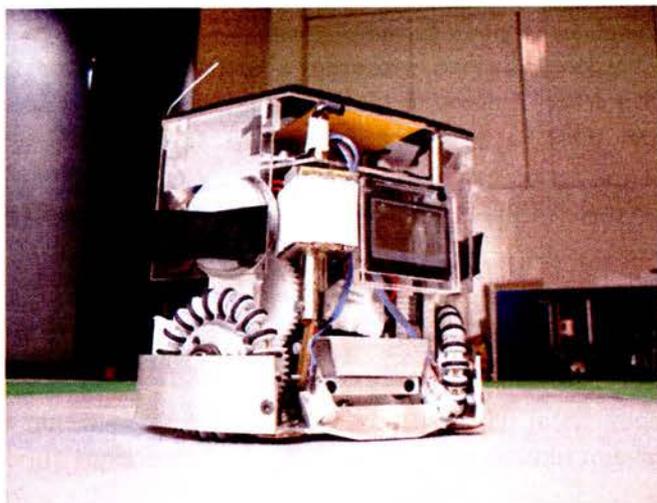


Figura 8 – Exemplo de um robot Omnidireccional

2.1.4 Sistema de Alimentação

Alguns sistemas robóticos podem conviver com limitações de mobilidade inerentes à sua fonte de energia ser externa. Se for viável o robot depender de energia exterior, então esta energia necessita apenas de ser condicionada e distribuída. Os robots de maior autonomia não podem, porém, ter cabos que os liguem permanentemente ao exterior. O robot deve então ter algum nível de independência em termos energéticos: este deve ser capaz de armazenar ou produzir energia.

A escolha das fontes de energia primárias deve ter em conta vários factores que são independentes da natureza da energia em causa:

- Características de carga e descarga (tempos e intensidades)
- Questões de manutenção e armazenamento
- Quantidade de energia acumulada e questões volumétricas (densidade, disponibilidade em forma adequadas, etc.)
- Preço inicial e durabilidade (número de ciclos, envelhecimento, etc.)
- Restrições mecânicas (peso, dimensões e questões de arrumação)
- Restrições quanto à recarga da energia a acumular

A utilização de motores de explosão enfrenta graves limitações quer a nível de segurança quer a nível da poluição.



Principalmente por estes motivos são pouco interessantes para aplicações em veículos autónomos, ainda mais se estes alguma vez entrarem em ambientes fechados.

As células de combustível são uma alternativa recente às baterias para veículos autónomos. Estes equipamentos não armazenam energia eléctrica, produzem-na. O subproduto da reacção química inerente à célula de combustível é a água. A procura de veículos com emissões nocivas nulas leva a que já existam carros e autocarros utilizando estas tecnologias.

A utilização de células de combustível enfrenta os seguintes obstáculos:

- Consumo de um combustível
- Produção de água
- Podem existir problemas de segurança

Apesar destas questões, as células de combustível são interessantes pois permitem densidades de energia superiores às das baterias, sendo portanto relevantes para área dos sistemas robóticos.

Após o estudo do projecto mecânico e do sistema que fornece energia ao sistema robótico, estuda-se todo o equipamento a bordo do robot.

2.1.5 Sistema embebido

Designa-se por sistema embebido o conjunto de sistemas a bordo do robot que lida com as interfaces robot-ambiente e processamento dos respectivos sinais. Este sistema lida com a conversão de energia de e para o exterior incluindo sensores, actuadores e respectivos circuitos de condicionamento. A aquisição de sinal dos sensores faz parte deste sistema, assim como o seu pré-processamento, eventual conversão e eventual disponibilização adequada, por exemplo, numa rede interna ao robot. Também a geração de sinais adequados aos actuadores é da responsabilidade deste importante subsistema do robot.

2.1.5.1 Sistemas Tempo Real

Um sistema tempo real (*Real Time*) é um sistema em que a correcção do resultado não depende apenas da correcção lógica da computação mas depende também do tempo em que essa computação foi completada. Se o limite temporal não foi cumprido então diz-se que ocorreu uma falha do sistema.

O sistema embebido é um sistema que está sujeito a restrições de Tempo Real visto que o atraso de processamento na zona de interface robot-mundo cria atrasos na execução de ordens e/ou atrasos na percepção do ambiente e compromete ainda o ciclo de decisão e controlo.



2.1.5.2 Classificação de Sensores

Os sensores podem ser classificados quanto à forma como interagem com o ambiente:

- **Sensores passivos** não adicionam energia ao ambiente circundante durante o seu funcionamento podendo, no entanto, de alguma forma retirar energia do ambiente;
- **Sensores activos** enviam energia para o meio e medem essa interacção com o ambiente.

Um exemplo de um sensor activo é o sonar que envia energia para o meio e mede o tempo que decorre até à chegada do respectivo eco. Uma câmara é um sensor passivo no sentido em que a sua medida é baseada em energia que não é emitida por si.

2.1.5.3 Sensores Comuns em Robótica

Os subsistemas de sensorização utilizados em sistemas robóticos são muitas vezes compostos por sensores e sistemas adicionais. Alguns subsistemas e respectivos mecanismos adicionais frequentemente utilizados são:

- Sistema de Visão – câmaras
- Ultra-Sons
- Sensores ópticos de Infra Vermelhos (IR - *Infra Red*)
- Radares - *RADAR - Radio Detection And Ranging*
- Sensores de contacto
- Acelerómetros
- Bússolas
- Inclinómetros e pêndulos
- Giroscópios
- Codificadores (*encoders*) absolutos ou relativos

Determinados subsistemas de sensorização podem fazer uso de múltiplos princípios elementares de medida, procurando assim uma medida com melhores características que a medida isolada.

Podem até receber comandos de configuração para que os resultados do subsistema tenham o máximo de informação útil. Exemplos deste tipo de sistemas são os sensores configuráveis, que podem ser considerados subsistemas de sensorização. Um exemplo de um sensor deste tipo é um sonar que permita a configuração do seu raio de acção.



2.1.5.4 Actuadores

Os actuadores eléctricos e os actuadores pneumáticos são muito frequentes em sistemas robóticos. Os actuadores eléctricos são muito utilizados principalmente devido à sua facilidade de comando e flexibilidade. Os actuadores pneumáticos permitem, no entanto, forças muito elevadas, superiores às que se conseguem com facilidade utilizando actuadores eléctricos.

Os motores eléctricos rotativos são quase omnipresentes para motorização de veículos móveis. Existem também motores eléctricos lineares que podem ser utilizados para actuadores robóticos, embora a sua utilização seja menos frequente.

Os motores eléctricos de corrente contínua são a solução mais frequente para tracção de veículos móveis. Devido às características próprias destes actuadores de alta rotatividade, muitas vezes utilizam-se caixas desmultiplicadoras para obter maiores binários. É também possível utilizar motores de corrente alternada mas a sua utilização em sistemas robóticos é menos frequente devido à complexidade acrescida do seu comando e funcionamento.

2.1.5.5 Implementação de Subsistemas

Cada subsistema do robot deve traduzir uma repartição em blocos coerentes. Todos os sistemas devem possuir uma interface bem documentada com o exterior e com as outras partes do robot.

Na implementação prática destes subsistemas existe muitas vezes discussão acerca da possibilidade de executar determinadas tarefas por electrónica dedicada ou por software. Esta é uma decisão tecnológica importante que acarreta consequências a nível dos tempos de resposta que o sistema terá com implicações ao nível do Tempo Real. Realizar tarefas por software, por exemplo dentro de um microcontrolador (MCU), pode ser uma solução flexível e de baixo custo. A alternativa é a implementação do sistema em hardware dedicado que permitirá tempos de processamentos muito menores.

2.1.5.6 Comunicações Internas

A comunicação mais frequente dentro de sistemas robóticos é, possivelmente, RS232.

Existem diversas variantes desta norma, sendo talvez a mais utilizada a norma RS485 que permite comunicação em rede.

Os mais recentes MCU's oferecem, no entanto, diversas possibilidades de comunicação dentro do mesmo integrado. Muitos incluem mesmo suporte para protocolos de rede algo complexos. Estas redes denominadas como **redes de sensores e actuadores** permitem distribuir sinais de forma simples pelo interior de um sistema.



Estas redes apresentam algum nível de suporte para funcionamento sobre restrições de Tempo Real. Alguns exemplos de redes de sensores e actuadores integradas em MCU's são:

- CAN – *Controller Area Network*
- I2C ou TWI – *Two Wire Interface*
- USB/USB2 – *Universal Serial Bus*
- *Ethernet*

Após o estudo físico do sistema, o estudo seguinte diz respeito à estratégia que o robot utiliza para cumprir os seus objectivos, em particular, ser capaz de chegar ao seu destino.

2.1.6 Navegação

A navegação envolve diversas capacidades como por exemplo o contorno de obstáculos e actualização dinâmica do mapa actual. As técnicas mais utilizadas para navegação são:

- Utilização de “mapas de estradas” de segmentos utilizáveis
- Decomposição em sub-blocos do mapa actualizando blocos com informação ocupado/livre mais recente
- Campos de Atracção/Repulsão (obstáculos repelem o robot que é atraído pelos pontos onde pretende chegar).

A navegação pode ser considerada como a plataforma intermédia que faz cumprir a ordem de chegar a um certo destino. O nível de autonomia que se pretende dar a esta camada intermédia de decisão é muito variável conforme a aplicação. Pode acontecer que um dado problema de navegação não tenha solução perante o nível de autonomia concedida a este sistema. Neste caso novas decisões de nível superior são necessárias para que o robot atinja os seus objectivos.

Considerando que é possível uma localização satisfatória para o problema em causa, o problema seguinte prende-se em fazer com que o robot se desloque para cumprir os seus objectivos. É então necessário saber onde o robot está, para onde deve seguir e como o conseguir, isto é, é necessário que o robot navegue através do caminho escolhido até ao seu destino.



A arquitectura típica de um sistema de navegação pode ser decomposta tal como mostra a figura 9.

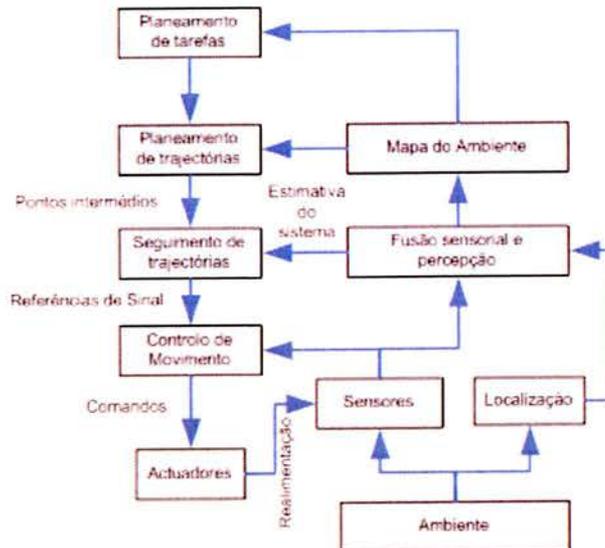


Figura 9 - Arquitectura típica de um sistema de navegação

Sistemas com este tipo de arquitectura permitem fechar a malha de comando a vários níveis o que permite por sua vez:

- Contorno e desvio de obstáculos – a baixo nível;
- Fusão de dados para criar uma estimativa global do sistema – a nível intermédio;
- Actualização do mapa corrente – a nível superior.

Tipicamente a navegação acaba por ser abordada como sendo o seguimento de referências intermédias até alcançar destinos finais onde são atribuídas novas tarefas. A camada de software da navegação recebe ordens do nível hierárquico superior, que é apresentado de seguida.



2.1.7 Decisão

A arquitectura do software de nível mais elevado é crítica para o desempenho satisfatório do robot que se pretende projectar. Chama-se ao software com mais elevado nível de abstracção o software de **decisão**.

O módulo de **decisão** toma as suas decisões com base em informações do mundo real fornecidas pelo bloco de entradas do robot e muda esse mesmo estado do mundo por intermédio das saídas do sistema.

As considerações mais importantes que o software de decisão necessita de ter em conta são:

- Como guardar informação acerca do seu próprio estado
- Como guardar informação acerca do ambiente exterior
- Como tomar decisões.

As abordagens mais frequentes à decisão em sistemas robóticos são:

Puramente reactiva – sem planeamento a prazo, para cada condição há uma reacção programada, sem estado interno do sistema;

Hierárquico – controlo e planeamento utilizando diferentes níveis de abstracção a informação do mundo é utilizada para gerar sequências de comandos a cumprir

Controlo híbrido – combina as duas abordagens anteriores num sistema hierárquico onde o nível inferior é reactivo e o nível superior produz sequências de acções

Comportamental – incluem partes reactivas mas utilizam representações complexas do mundo; utiliza-se por vezes coordenação, competição e concorrência para decidir a lista de acções a tomar.

A nível de implementação, grande parte dos robots utiliza PCs a bordo do robot para implementação das camadas superiores de decisão em sistemas robóticos.

São amplamente utilizados os Sistemas Operativos genéricos Windows e Linux. Menor número de aplicações utiliza sistemas operativos mais adequados a desempenho em Tempo Real tal como Tornado, QNX, LinxOS, VxWorks, AperiOS e ainda diversas variantes Tempo Real do Linux.

2.1.8 Fusão de Informação

A fusão de informação é um processo segundo o qual a utilização de vários sensores permite obter elevados níveis de qualidade da informação

A existência de múltiplos sensores é a única maneira de assegurar decisões correctas na presença de dados errados, desde que existam dados em quantidade suficiente. A nível de exemplo deve ser referido que no caso de ter 3 relógios para saber as horas, uma média não é uma boa medida de fazer a fusão de informação.



Para uma melhor fusão de informação é essencial a caracterização de cada uma das medidas e em particular a incerteza relativamente a cada uma delas.

A caracterização de cada um dos sensores é então essencial:

- Precisão, Exactidão e Sensibilidade
- Resposta dinâmica
- Caracterização da eventualidade de leituras erradas
- Caracterização do comportamento sob avarias

2.2 Análise de requisitos

2.2.1 Particularidades do Festival de Robótica2006

O desafio proposto consiste na participação na prova de condução autónoma a realizar no festival de robótica 2006.

Para tal será necessária a elaboração de um robot móvel e autónomo capaz de percorrer um percurso ao longo de uma pista fechada. Esta prova apresenta semelhanças marcantes com a condução de um veículo automóvel numa estrada convencional. A pista é delimitada por duas linhas brancas, tem duas faixas separadas por uma linha tracejada, a forma aproximada de um 8, um cruzamento no centro com passadeira, um par de painéis sinaléticos e um túnel sobre uma das curvas.

A competição decorre em três mangas, sendo o principal objectivo percorrer por duas vezes a totalidade da pista no mais curto espaço de tempo possível e sem penalizações. O grau de dificuldade vai crescendo em cada nova manga: na primeira é efectuada um contra-relógio ao longo de duas voltas completas à pista; na segunda são acrescentados os semáforos que indicam situações de paragem obrigatória, continuação do percurso pela via da esquerda ou continuação do percurso em frente, a obrigatoriedade de o robot terminar a sua prova num parque de estacionamento e a colocação de um obstáculo na pista; na terceira manga, são ainda acrescentados um túnel e uma zona de obras delimitada por um conjunto de cones de obras interligados por uma fita colorida.

2.2.2 State of art

Existem actualmente inúmeras equipas a participar na prova de condução autónoma sendo em grande parte projectos continuados por vários anos.

O grau de complexidade dos robots é muito variado, existem equipas a participar com *mindstorms* que possuem uma estrutura mais simples, até aos robot mais complexos que foram desenvolvidos ao longo dos últimos anos.

Em relação à visão artificial, a maioria das equipas que concorrem na prova de condução autónoma usam câmaras *firewire* embora hajam alguns casos em que são utilizadas *webcams*. No caso particular dos *mindstorms*, a utilização de câmaras é tecnologicamente inviável, ficando estas equipas sujeitas às penalizações devido à incapacidade de reconhecimento dos semáforos que fazem parte da prova.



A arquitecta mais utilizada é sem duvida a diferencial, embora existam alguns robots com arquitectas do tipo *Ackerman* e triciclo.

No que diz respeito à estrutura de comando dos robots participantes, verifica-se que a estrutura mais usual é a reactiva embora exista a possibilidade de se implementar uma estrutura do tipo estado do mundo.

Tendo como ponto de partida este contexto, a ambição deste projecto passa por, por um lado, compreender as diversas formas de encarar o problema da condução autónoma que actualmente se apresentam, mas por outro, ter sempre presente o espírito inovador que permita encontrar novas soluções.

2.2.3 Especificações da prova de condução autónoma

Nesta alínea é dada a conhecer as particularidades que deverão ser respeitadas por todos intervenientes, pois são “as condições inerentes à área de competição” que se decompõem nos seguintes parâmetros:

- **Dimensões**
O robô tem que caber integralmente num paralelepípedo rectângulo com 60cm de largura, 100cm de comprimento e 80cm de altura.
- **Autonomia**
O robô é um veículo completamente autónomo. Todas as decisões são tomadas pelos sistemas nele incluídos e todos os dispositivos de armazenamento de energia estão lá instalados.
- **Segurança**
O robô deverá possuir mecanismos de segurança adequados à sua potência e modo de locomoção que permitam a sua imobilização em situações que possam representar perigo para pessoas ou bens.
- **Cores**
O chão da pista é de cor preta. As linhas laterais são brancas. A zona exterior à pista, mais adjacente à linha externa, numa banda com pelo menos 20cm de largura, é da mesma cor do chão da pista
- **Zona de Obras**
A zona de obras é limitada à entrada e à saída por dois pares de cones cor de laranja, colocados sobre a linha delimitadora da pista. O percurso em obras será delimitado à esquerda e à direita por uma sequência de cones. Cada par de cones consecutivos encontra-se unido por uma fita com 5cm de largura, colocada na vertical e de cor alternadamente vermelha e branca.
- **Obstáculos**
Será colocado na pista um obstáculo que impedirá a circulação dos robôs por uma das faixas de rodagem. O obstáculo é um paralelepípedo rectângulo de base quadrada com 60cm de lado e uma altura mínima de 20cm.
- **Painel Sinalético**
Sobre a passadeira serão montados dois painéis TFT com 17” de dimensão diagonal, em posição invertida e verticalmente alinhados com cada uma das faixas de rodagem.



2.3 Especificação de requisitos

Tendo em conta a análise de requisitos e a pesquisa de soluções efectuada anteriormente, foi projectado um sistema robótico cuja arquitectura será descrita na alínea seguinte.

2.3.1 Arquitectura do sistema robótico

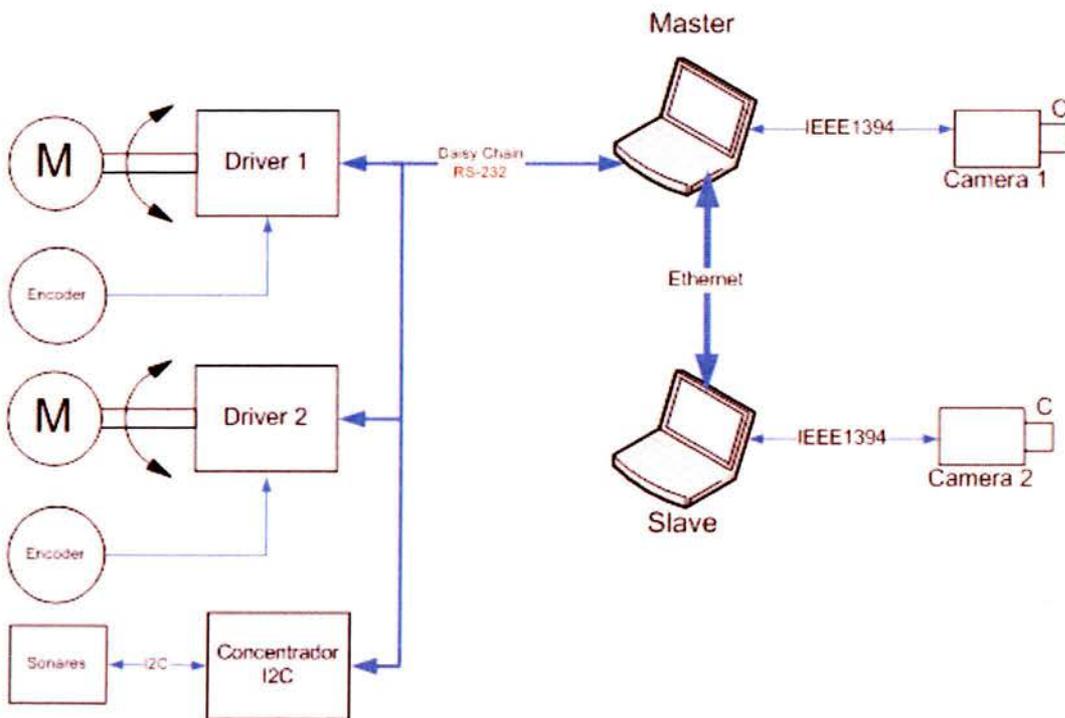


Figura 10 - Arquitectura geral do sistema robótico

O sistema robótico a desenvolver pode ser dividido nos seguintes blocos:

- Sistema de Localização
- Sistema de Processamento de dados e decisão
- Sistema de Locomoção



A figura 11 ilustra a divisão do sistema robótico em blocos:

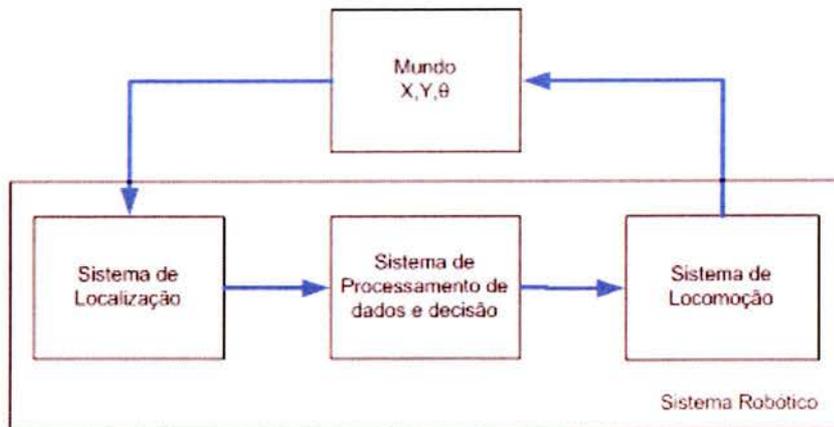


Figura 11 - Diagrama de blocos do sistema robótico

2.3.2 Sistema de Locomoção

O sistema de locomoção será constituído por actuadores e *drivers* de controlo.

Os actuadores usados serão motores DC uma vez que se adequam ao sistema de alimentação do robot e pelo facto do controlo de velocidade poder ser realizado através de um sistema de PWM, razão pela qual serão utilizados *drivers* de controlo que disponibilizam saídas em PWM.

O sinal de PWM será gerado nos próprios *drivers* recorrendo a um microcontrolador.

2.3.3 Sistema de Localização

O sistema de localização será constituído pelo subsistema de visão, pelo subsistema de hodometria e pelo subsistema de detecção de obstáculos.

O subsistema de visão será constituído por duas câmaras *firewire*, para o processamento das faixas de rodagem e do sinal de semáforo.

O subsistema de hodometria será constituído por *encoders* incrementais.

A detecção de obstáculos será realizada recorrendo ao uso de sonares com comunicação via I2C.

2.3.4 Sistema de Processamento de dados e Decisão

O sistema de processamento de dados e decisão será constituído por dois PC. Um dos PC (Master) será responsável pela fusão e filtragem dos dados provenientes dos restantes sistemas. A decisão de actuação também será realizada nesta unidade de processamento.

A função do segundo PC (Slave) será a de processar o sinal de semáforo.



O software de controlo do robot será realizado numa aplicação desenvolvida na unidade de processamento principal do robot, usando a plataforma de programação, Kylix3 sobre o sistema operativo Linux. Pelo facto de ser uma linguagem orientada a objectos, permite o fácil acesso aos dispositivos e sistemas periféricos, assim como a fácil construção de interfaces para o utilizador.

O módulo de decisão está incorporado no software de controlo do robot e será desenvolvido segundo a abordagem **hierárquica**, ou seja, efectuando o controlo e planeamento utilizando diferentes níveis de abstracção; a informação do mundo é utilizada para gerar seqüências de comandos a cumprir.



Figura 12 - Arquitectura do módulo de decisão

2.3.5 Funcionalidades do software de controlo

A aplicação desenvolvida para o controlo do robot deverá apresentar as seguintes funcionalidades:

- Desenvolvimento de uma maquina de estados para o controlo do robot
- Desenvolvimento de rotinas de controlo de movimento
- Planeamento de trajectórias
- Simulação e validação do controlo desenvolvido
- Simulação do processamento dos dispositivos de sensorização
- Ajustagem dos PID's de controlo da malha de velocidade
- Monitorização da posição do robot no mundo
- Monitorização das velocidades dos motores
- Monitorização dos dados de processamento dos dispositivos de sensorização (sonares, *encoders*, câmaras *firewire*)
- Configuração dos parâmetros de captura de imagem
- Configuração dos parâmetros de funcionamento dos sonares



2.3.6 Comunicação entre sistemas

A comunicação entre todos os sistemas especificados será feita através das seguintes redes:

A comunicação entre o sistema de processamento, sistema de locomoção e o subsistema de detecção de obstáculos, que faz parte do sistema de localização, será realizada através de uma rede *Serial Daisy-Chain*.

A comunicação entre o sistema de processamento e o subsistema de visão será realizado através de um barramento *firewire*.

A comunicação entre a unidade de processamento principal e secundária será realizada através de uma rede *ethernet*.

A comunicação entre os sonares e unidade de processamento destes dados será realizada através de um barramento I2C.

Em suma, a comunicação no sistema robótico, será realizada recorrendo às seguintes redes de sensores e actuadores:

- *Serial Daisy-Chain*
- *I2C*
- *Firewire*
- *Ethernet*

2.3.7 Configuração e tipo de locomoção

A configuração do sistema de locomoção do robot será do tipo diferencial, por ser extremamente versátil, de fácil construção e controlo.

A justificação dos requisitos especificados necessários à implementação do projecto, assim como a tecnologia utilizada, será realizada nos capítulos 3 e 4 deste documento.



Capítulo 3

Estrutura mecânica

O projecto mecânico de um sistema robótico é parte fundamental para o bom funcionamento global. Este deve ter em conta as necessidades e características de cada uma das partes envolventes. É fundamental ter especial atenção às dimensões, distribuição do peso, à resistência física do sistema como à modularidade. Assim o tempo e esforço para o diagnóstico e reparação de eventuais avarias são otimizados.

3.1 Projecto em CAD

A utilização de ferramentas CAD ajudam a executar e validar o projecto mecânico. Estas possibilitam a análise de eventuais erros de concepção tais como intercepção entre partes do robot, fornecendo também várias perspectivas do sistema, permitindo assim a optimização da estrutura mecânica projectada.

Alguns sistemas CAD, permitem a ligação a sistemas CAM possibilitando a produção das diferentes peças em máquinas de controlo numérico.

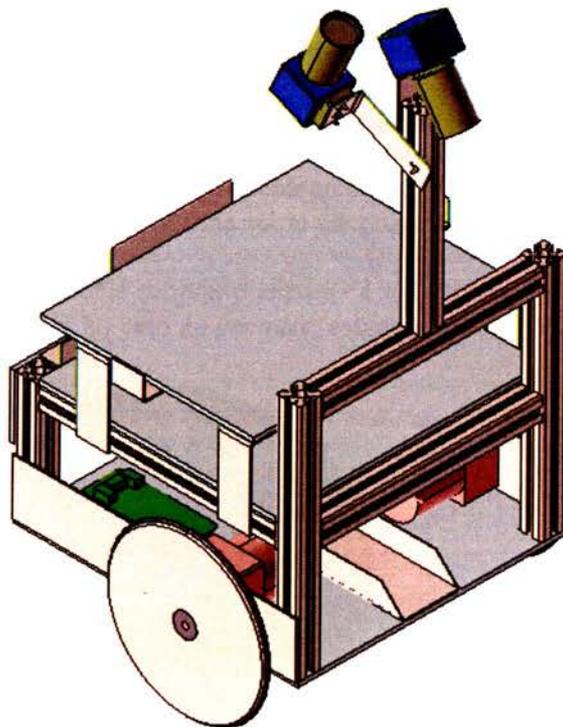


Figura 13 - Vista em perspectiva do robot



3.2 Implementação

A estrutura física do robot é feita em calha de alumínio (Maytec tipo 30x30, 4F, *light*). A escolha deste material deve-se ao seu peso e robustez. É também utilizado acrílico de 6mm para o suporte dos portáteis e várias peças em alumínio maquinadas nas máquinas de controlo numérico disponibilizadas pelo 5DPO.

Durante o projecto foi tido em atenção a distribuição do peso do robot, tendo-se o cuidado de colocar os motores um pouco descentrados para que o centro de massa tendesse ligeiramente para o castor.

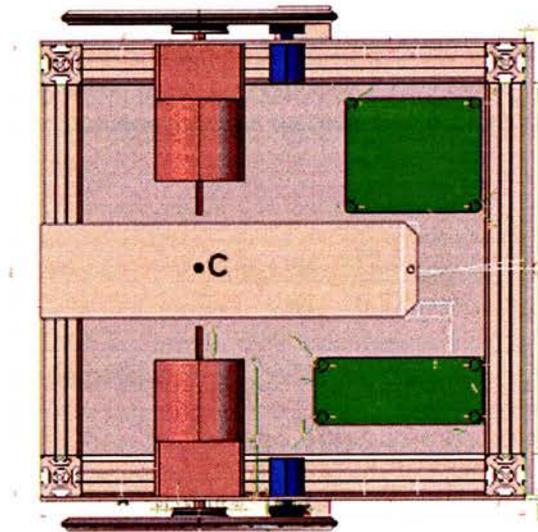
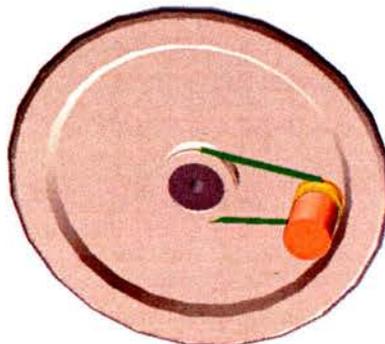


Figura 14 - Vista de baixo do robot

O robot foi construído utilizando a configuração diferencial, pois este tipo de configuração é extremamente versátil, de fácil construção e controlo, sendo constituída por duas rodas motorizadas independentemente e um terceiro apoio (castor). Esta configuração permite que o robot rode sobre ele próprio (sobre o ponto C).

Com a necessidade de acoplar o *encoder* à roda foram projectadas e construídas polias. A polia está presa ao veio do *encoder*, estando ligada à roda por um elástico.



Legenda

Laranja: *encoder*
Amarelo: polia
Verde: elástico
Cinzento: roda

Figura 15 - Vista em perspectiva do sistema de polias



- Dimensionamento das polias

A razão de transformação entre a velocidade angular da roda e do *encoder* deve ser alvo de um estudo cuidado. Existe o compromisso entre a frequência de amostragem a que os impulsos do *encoder* são adquiridos e a resolução do mesmo.

Para que o erro da medida do *encoder* seja mínimo, é conveniente que a razão de transformação seja pequena, por sua vez a razão deve ser elevada para maximizar a diferença entre a frequência de saída dos sinais do *encoder* e a frequência que estes são amostrados.

A frequência de amostragem é de 10kHz. Como na saída do *encoder* existem dois sinais com a mesma frequência mas desfasados de 90°, a frequência máxima dos sinais amostrados é de 2,5kHz, o que corresponde a 12.5 rad/s.

Sabendo que a velocidade máxima da roda é de 6.21 rad/s, a razão de transformação é:

$$\frac{r_2}{r_1} = \frac{w_2}{w_1} = \frac{12.5}{6.21} \cong 2$$

Como tal, o raio da polia é:

$$r_1 = \frac{r_2}{2} = \frac{19,5}{2} = 9.75mm$$

No anexo A deste documento está o projecto em CAD de todos os componentes que fazem parte da estrutura mecânica do robot.



Capítulo 4

Estrutura electrónica

4.1 Sistema de Alimentação

Na figura seguinte apresenta-se os esquemas eléctricos do robot. Destacado a azul, está o circuito de alimentação geral que permite comutar entre a alimentação da fonte e as baterias. Estas deveram ser ligadas ao circuito a vermelho no conector J1 e J3.

Este circuito para além de alimentar o robot quando este não está ligado à fonte de alimentação, também permite a alimentação do PC portátil e das câmaras a uma tensão fixa e estável de 20V. É de salientar a utilização de um conversor DC-DC (PTN78020), que regula a tensão de saída para 20V numa gama de entrada que pode variar de 15V a 36V.

Marcado a verde, está o circuito de alimentação dos diversos *drivers*.

O circuito também permite ligar/desligar os motores sem cortar a alimentação geral, para que se possa arrastar o robot sem o perigo de danificar os *drivers* devido à F.E.M que os motores produzem.

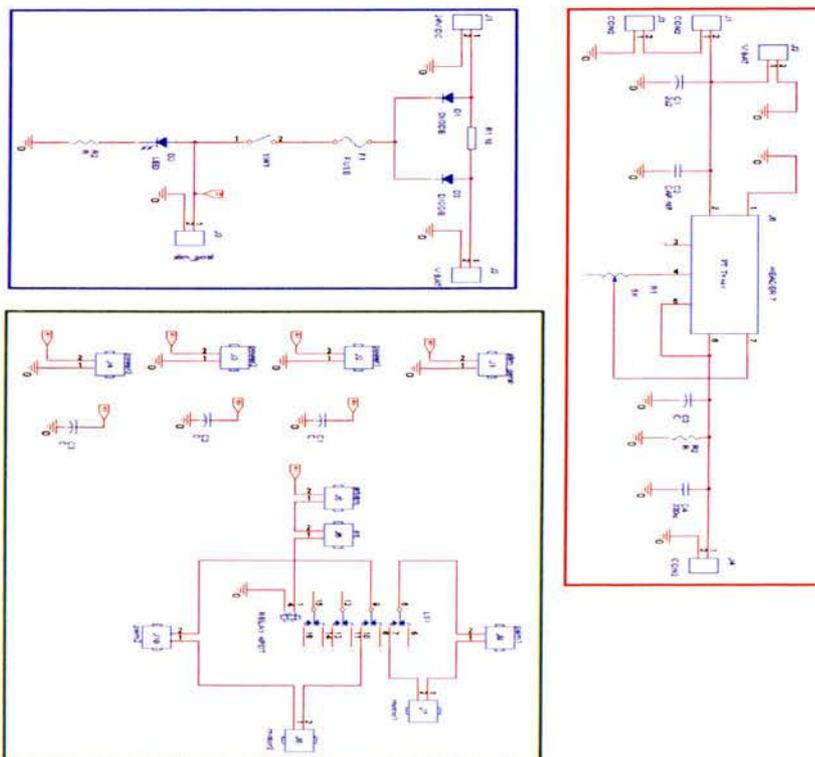


Figura 16 - Esquemas dos sistemas eléctricos do robot



4.2 Sensores Utilizados

4.2.1 Câmaras *firewire*

São utilizadas duas câmaras *firewire* da *ImagingSource*, modelo DBK 21F04. Este tipo de câmaras possui um sensor CCD que transforma fotões em electrões.

Em particular, na câmara utilizada, a saída não é em RGB, mas sim em formato *Bayer*. Isto deve-se ao facto de que por razões económicas a câmara apenas possui um sensor CCD em vez de três, um para cada componente de cor.

Assim, cada pixel possui apenas uma das componentes de cor, ou *RED*, ou *GREEN*, ou *BLUE*. As restantes componentes de cor do pixel são calculadas no software por interpolação das componentes dos pixels vizinhos.



Figura 17 - Câmara Firewire (DBK 21F04)

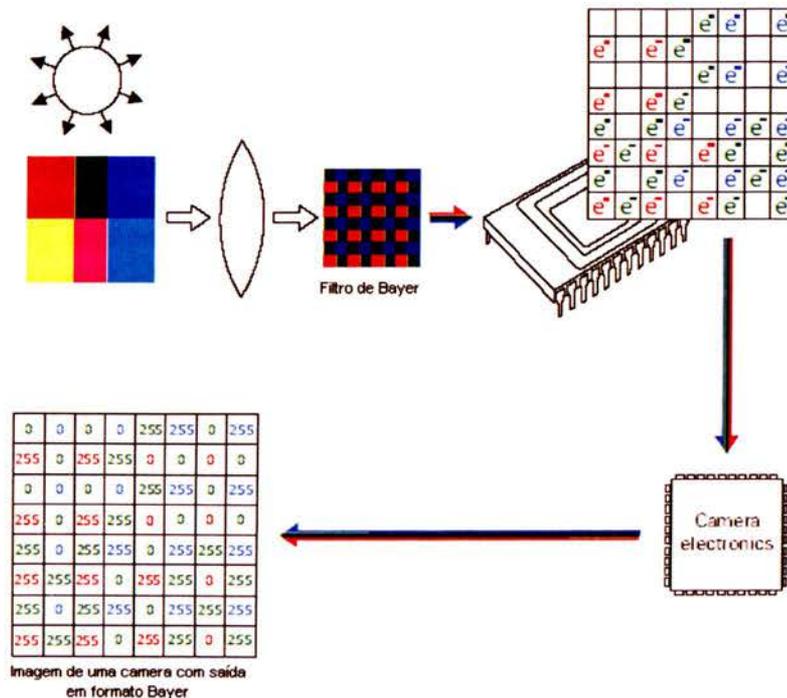


Figura 18 – Formato da imagem capturada (Bayer)



4.2.2 Encoders

O *encoder* é um transdutor que converte um movimento angular ou linear numa série de impulsos digitais. Esses podem ser usados para determinar velocidade, taxa de aceleração, distância, rotação ou posição.

O sistema de leitura é baseado num disco (*encoder rotativo*), formado por janelas radiais transparentes e opacas alternadas. Este é iluminado perpendicularmente por uma fonte de luz infravermelha, quando esta atravessa as ranhuras, é projectada no receptor. O receptor converte a luz em impulsos eléctricos.



Figura 19 - Esquema de um *encoder* incremental

- *Encoders* incrementais

O *encoder* incremental fornece normalmente dois pulsos quadrados desfasados em 90° , que são chamados usualmente de canal A e canal B. A leitura de somente um canal fornece apenas a velocidade, enquanto que a leitura dos dois canais fornece também o sentido do movimento.

A resolução do *encoder* incremental é dada por impulsos/volta (normalmente chamado de PPR), isto é, o *encoder* gera um certo número de impulsos por volta (no caso de um *encoder* rotativo).

Para determinar a resolução basta dividir o número de impulsos por 360° , por exemplo, um *encoder* fornecendo 200 impulsos/ revolução, geraria um impulso eléctrico a cada $1,80^\circ$ mecânicos.

Na figura 20 está ilustrado o modelo dos *encoders* utilizados, ref. E6A2 – CW5C da OMRON.



Figura 20 – Imagem dos *encoders* usados (ref. E6A2–CW5C)

4.2.3 Sonares

Devido à necessidade de detectar obstáculos utilizou-se cinco sonares ref. SRF08.

Foi escolhido este modelo devido à sua interface I2C que possibilita a ligação à porta I2C da placa dos *drivers*.

As linhas SDA e DCL necessitam de resistências de pull-up neste caso implementadas nos próprios *drivers*.

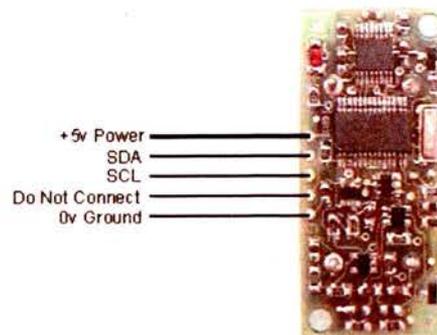


Figura 21 - Sonar (ref. SRF08)



Uma vez que os sonares estão todos no mesmo barramento, há a necessidade que tenham endereços diferentes.

Por defeito, todos os sonares tem o endereço 0xE0 que deverá ser alterado. Para efectuar esta operação, deverá existir apenas um sonar no barramento, em seguida, escreve-se uma sequência de três comandos.

1. Endereço Actual
2. 0xA0
3. 0xAA
4. 0xF2
5. Endereço Pretendido

Outra particularidade que estes sonares possuem é a possibilidade de alterar o ganho e o alcance. Tais valores deverão ser escritos respectivamente nos registos 1 e 2 do sonar.

4.3 Actuadores

Para a locomoção do robot, foram utilizados motores DC, ref. 5995 da RS, com uma caixa redutora de relação 5:1, ref. 718-853 da RS.

São utilizados este tipo de motores, uma vez que se adequam ao sistema de alimentação do robot e pelo facto do controlo de velocidade poder ser realizado através de um sistema de PWM.

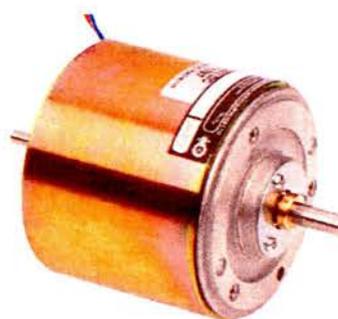


Figura 22 - Imagem dos motores utilizados (ref. 5995 da RS)

4.3.1 Modelos dos motores

Para o desenvolvimento do simulador do controlo é necessário determinar a planta dos motores.

A técnica utilizada para determinar a planta dos motores, consiste em introduzir ruído branco na entrada do motor, registar a resposta e através do método dos mínimos quadrados, determinar a planta dos motores.

O modelo dos motores é utilizado como uma primeira aproximação para a determinação dos parâmetros do PID, assim como para a modelização do robot, necessária para o desenvolvimento do simulador.

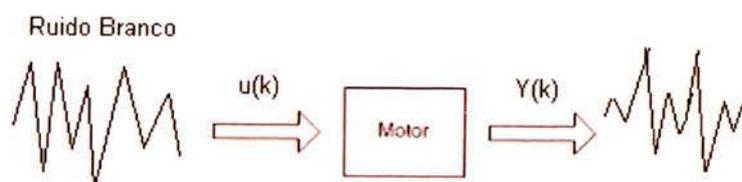


Figura 23 - Método utilizado para a determinação da planta do motor



4.3.1.1 Registo dos resultados obtidos

Para este teste foi utilizado um período de amostragem de 40 milisegundos, sendo registadas 255 amostras, o que parece ser um número aceitável para se inferir sobre os parâmetros da planta.

Apesar de serem utilizados dois motores do mesmo modelo, foi determinada a planta dos dois motores, uma vez que o robot efectua uma trajectória curvilínea quando se aplica a mesma referência de PWM em malha aberta. Tal comportamento é indicador que apresentam diferenças que não se podem desprezar para efeitos de afinação dos PIDs.

Para o Motor 1:

Como se pode verificar na figura 24, a partir da aproximação de terceira ordem, o erro deixa de ser significativo relativamente à resposta real da planta.

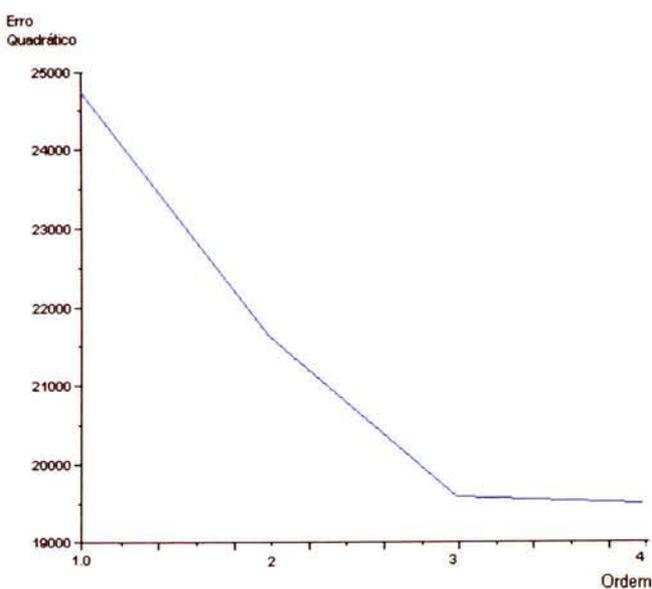


Figura 24 - Gráfico do erro quadrático da aproximação pelo método dos mínimos quadrados

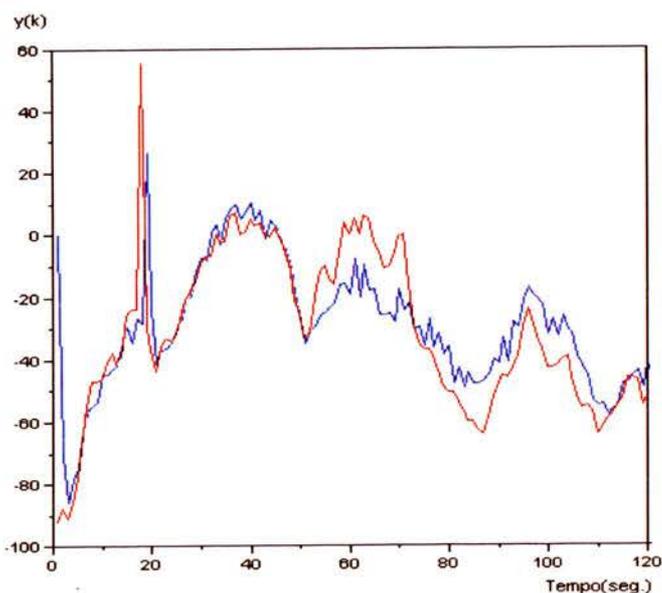


Figura 25- Aproximação da resposta dos motores para ordem 3

Legenda
 Resposta Real
 Resposta estimada

O modelo do motor 1 fica então definido pela seguinte equação no domínio discreto:

$$\frac{y(k)}{u(k)} = \frac{0.051 + 0.052z - 0.008z^2}{-0.066 - 0.263z - 0.677z^2 + z^3}$$



Para o Motor 2:

Como se pode verificar na figura 26, a partir da aproximação de terceira ordem, o erro deixa de ser significativo relativamente à resposta real da planta.

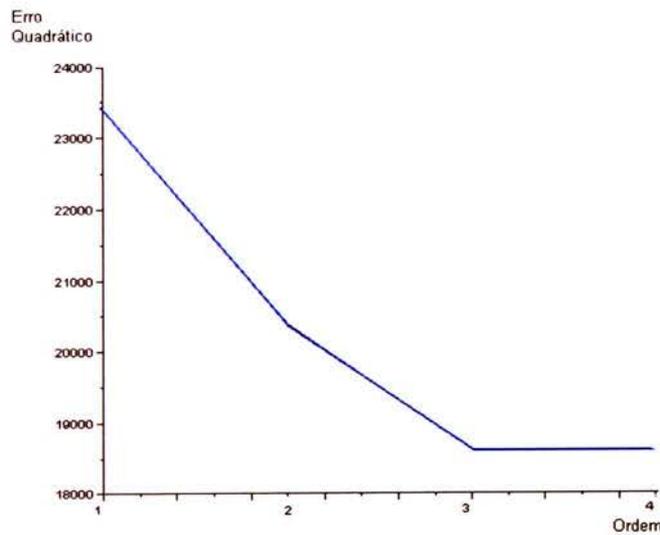


Figura 26 - Gráfico do erro quadrático da aproximação pelo método dos mínimos quadrados

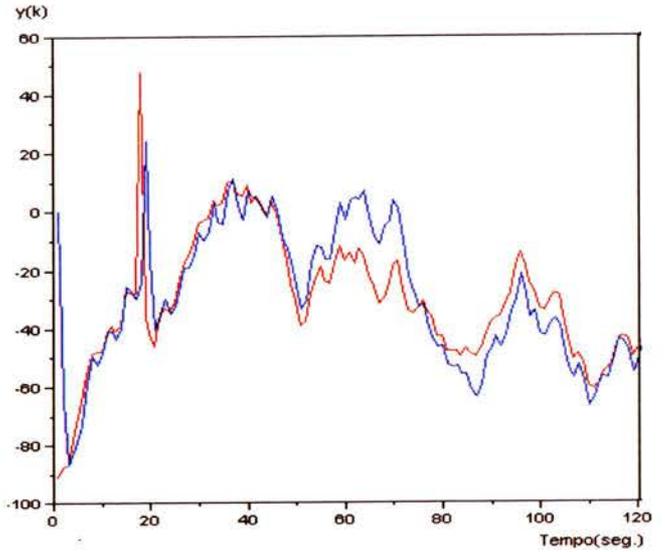


Figura 27 -Aproximação da resposta dos motores para ordem 3

Legenda	Resposta Real
	Resposta estimada

O modelo do motor 2 fica então definido pela seguinte equação no domínio discreto:

$$\frac{y(k)}{u(k)} = \frac{0.055 + 0.053z - 0.009z^2}{-0.024 - 0.292z - 0.692z^2 + z^3}$$



4.4 Drivers

Os *drivers* utilizados para o controlo dos motores foram cedidos pelo departamento de futebol robótico 5DPO, da FEUP.

Estes *drivers*, para além da electrónica de potência necessária para o controlo dos motores, incorporam um microcontrolador que gera os sinais em PWM para o controlo dos semicondutores de potência. No microcontrolador está também implementado um PID responsável pelo controlo em malha fechada da velocidade dos motores.

Na alínea seguinte será explicado a estrutura do PID implementado no microcontrolador, assim como a afinação de parâmetros do mesmo.

A comunicação entre os *drivers* e a unidade de processamento principal é realizada através de um protocolo sobre uma rede *Serial Daisy-Chain*. O protocolo de comunicação será explicado ao pormenor na alínea 6.7.1 deste documento.

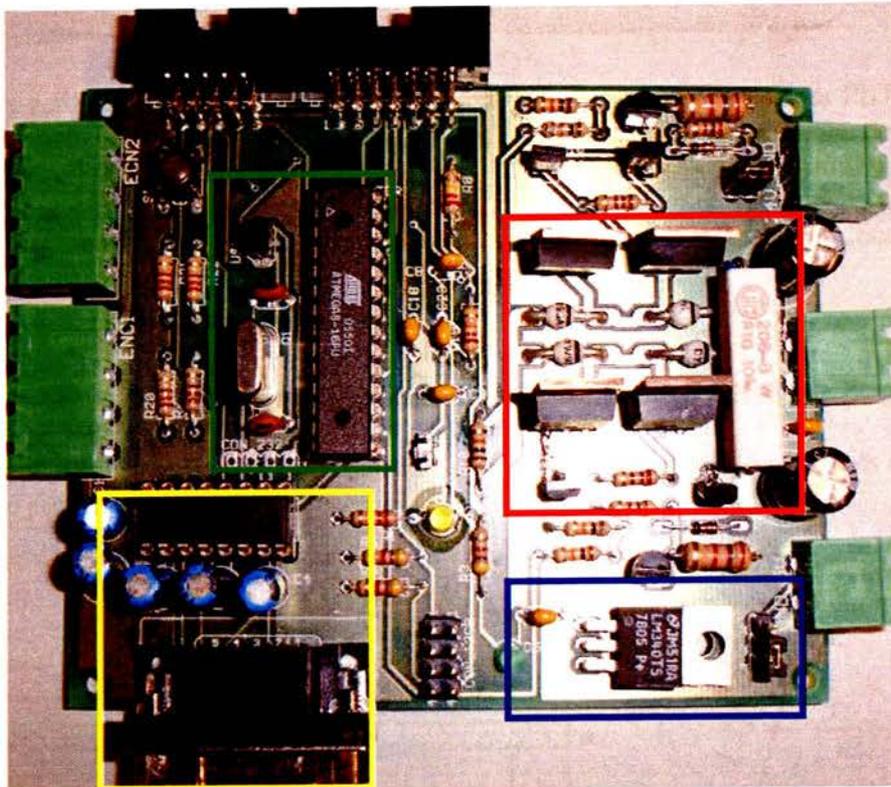


Figura 28 – Foto do *driver*

Assinalado a vermelho está o circuito de potência do *driver*, que controla o sentido de rotação e tensão entregue aos motores através de uma *ponte em H*.

Assinalado a verde está o circuito de processamento de sinal, constituído por um microcontrolador atmega8 que implementa o PID acima mencionado e que gera o sinal de PWM de controlo para a *ponte em H*.

Assinalado a amarelo está o circuito que suporta a comunicação via RS-232. Assinalado a azul está o circuito de alimentação dos *drivers*, constituído por um rectificador de tensão que gera o nível de tensão adequado ao circuito de sinal (5V).

4.4.1 Controlador PID

O PID implementado no microcontrolador dos *drivers* apresenta a seguinte estrutura:

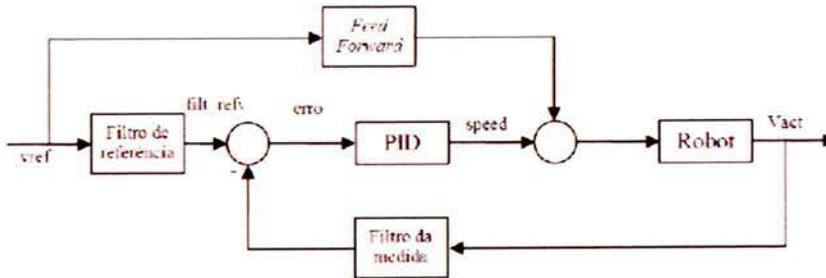


Figura 29 - Estrutura do PID implementado no microcontrolador do driver

Na verdade, o controlador implementado não é apenas um controlador PID, mas contém também um filtro de medida, um filtro de referência e ainda uma malha de *feed-forward*. Por existir, na camada de software, mais uma camada de controlo que também necessita de ser afinada, optou-se por só usar o controlador PID uma vez que, para a aplicação em causa, a afinação dos outros parâmetros não traz mais valias significativas.

4.4.1.1 Afinação dos PIDs

Os parâmetros do controlador PID foram afinados utilizando o método de Ziegler-Nichols para o ensaio em malha aberta.

A figura 30 ilustra como se pode determinar os parâmetros anteriores, através da reposta do sistema.

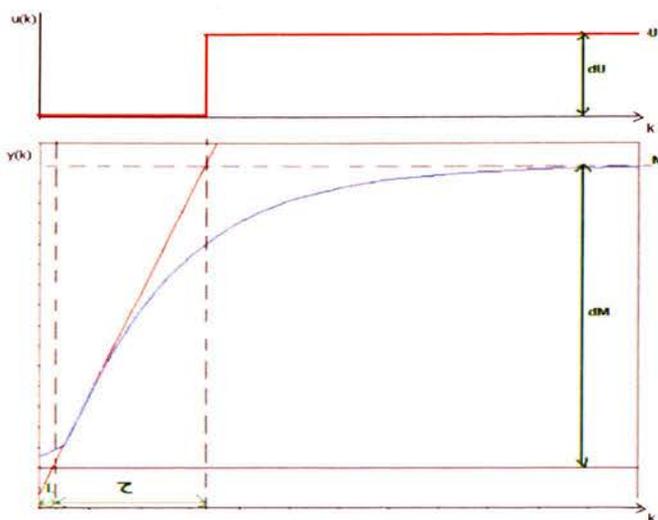


Figura 30 - Parâmetros para a afinação do PID na resposta do sistema

Para o ensaio em malha aberta:

$$K_p = \frac{\tau}{L * \frac{dM}{dU}}$$

K_p , Ganho da planta

Para afinar o controlador PI:

$$K_c = 0.9 * K_p$$

$$T_i = 3.3 * L$$

K_c , Ganho do controlador



Como dito anteriormente observou-se que o robot efectua uma trajectória curvilínea quando se aplica a mesma referência de PWM em malha aberta. Tal comportamento indica que os PIDs têm que ser afinados com diferentes parâmetros.

Aplicou-se um degrau de entrada com a referência de 2m/s e obteve-se a seguintes respostas em velocidade:

Para o motor 1:

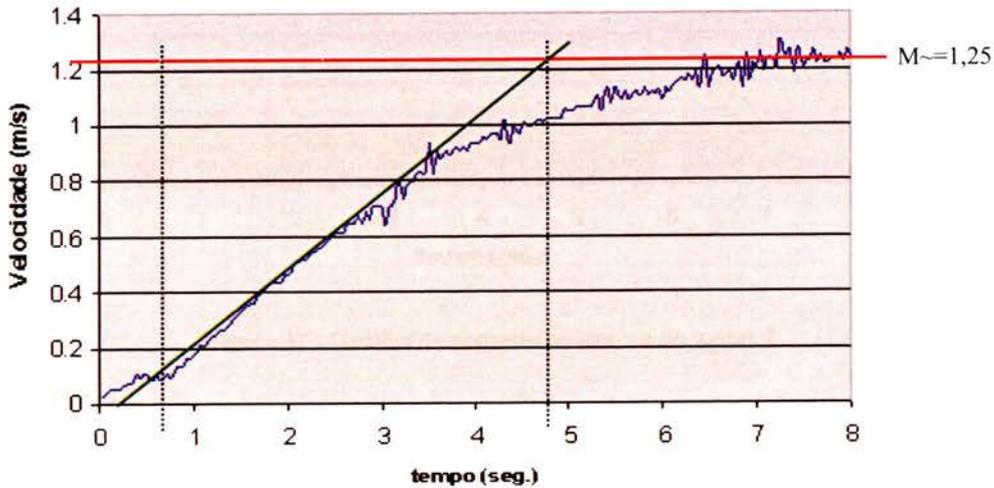


Figura 31 - Gráfico da resposta ao degrau do motor 1

Através dos resultados registados, o controlador PID para o motor 1 ficou ajustado com os seguintes parâmetros:

$$L \cong 0.8 \text{ segundos} \quad \tau \cong 4 \text{ segundos} \quad K_p = \frac{4}{0.8 * \frac{1.25}{2}} = 8$$

$$K_c = 0.9 * 8 = 7.2$$

$$T_i = 3.3 * 0.8 = 2.64 \text{ segundos}$$



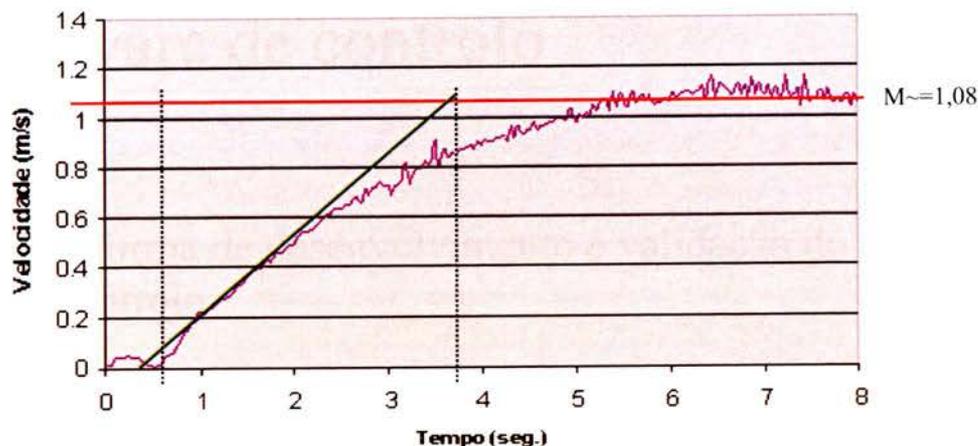
Para o motor 2:

Figura 32 - Gráfico da resposta ao degrau do motor 2

Através dos resultados registados, o controlador para o motor 2 ficou ajustado com os seguintes parâmetros:

$$L \cong 0.6 \text{ segundos} \quad \tau \cong 3.2 \text{ segundos} \quad K_p = \frac{3.2}{0.6 * \frac{1.08}{2}} = 9.8$$

$$K_c = 0.9 * 9.8 = 8.8$$

$$T_i = 3.3 * 0.6 = 1.98 \text{ segundos}$$

No anexo B deste documento estão listados todos os PCBs que fazem parte da estrutura eléctrica/electrónica do robot.



Capítulo 5

Software de controlo

5. Plataforma de desenvolvimento e validação do software de controlo

O objectivo desta plataforma consiste na implementação de uma aplicação implementada no sistema embecido que permita desenvolver, simular e validar o software de controlo do robot.

A aplicação foi desenvolvida em Linux, utilizando o ambiente de programação kylix3, que incorpora a linguagem de programação pascal, orientada ao objecto, o que permite efectuar de forma simples o desenvolvimento de interfaces com dispositivos de sensorização e o desenvolvimento de interfaces para o utilizador.

A aplicação permite o desenvolvimento de rotinas de controlo de movimento, o processamento dos sensores incorporados no robot (Câmaras *firewire*, *encoders* e *sonares*), assim com a fusão, filtragem de informação e decisão da actuação no robot.

Através desta aplicação é também possível a configuração dos dispositivos de sensorização (*sonares*, parâmetros de captura de imagem das câmaras *firewire*) assim como a afinação dos parâmetros dos PIDs implementados nos *drivers*.

A interface desenvolvida permite que o utilizador tenha acesso ao resultado do processamento dos diversos dispositivos de sensorização, assim como à evolução da máquina de estados de controlo do robot.

Esta plataforma integra também um simulador que permite validar as rotinas de controlo desenvolvidas, assim como validar a máquina de estados projectada para o controlo do robot.

As figuras abaixo indicadas ilustram a interface gerada para o utilizador onde estão assinaladas as diversas áreas onde a informação lhe é disponibilizada.



Monitorização do processamento dos dados de hodometria

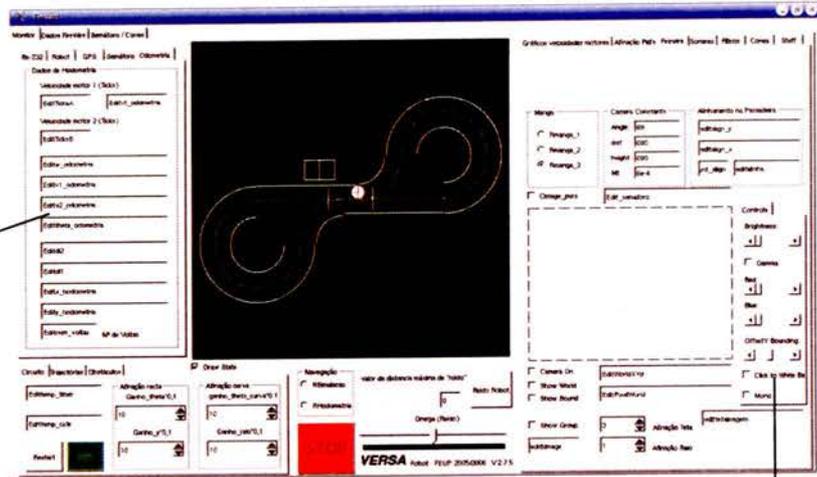


Figura 33 - Plataforma de controlo (Main)

Configuração dos parâmetros de captura de imagem

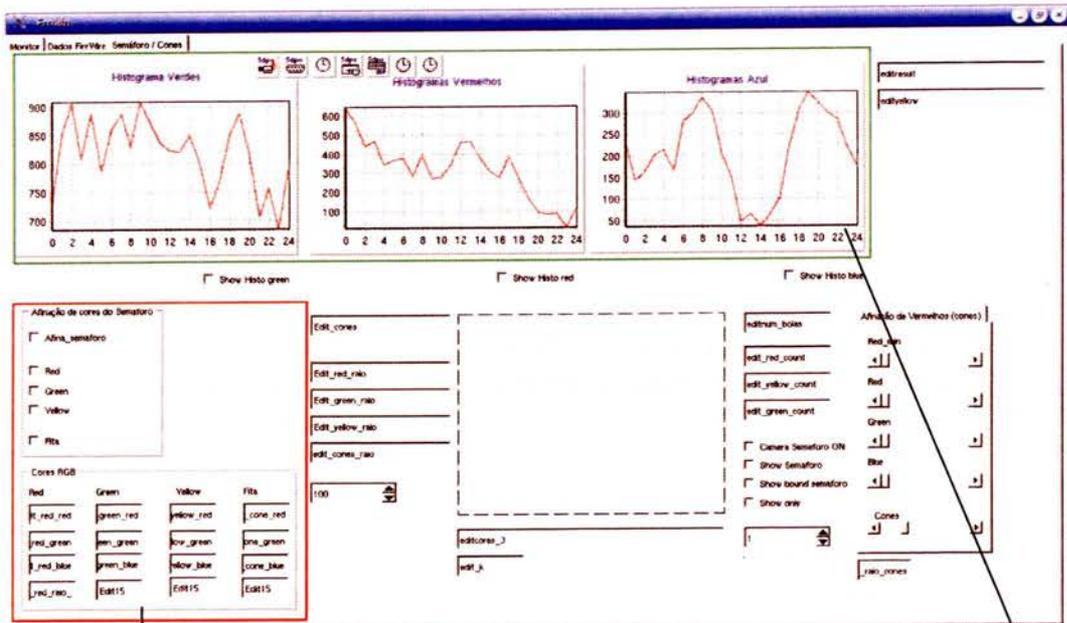
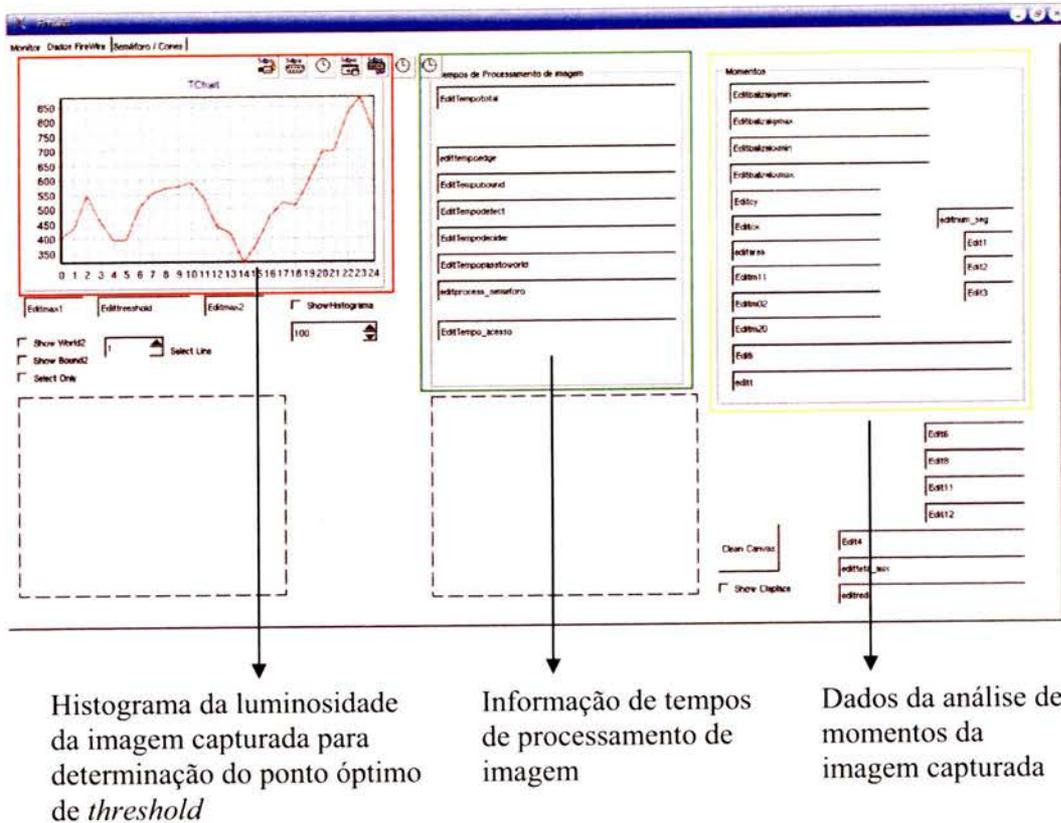


Figura 34 - Plataforma de controlo (Processamento de imagem)

Afinação de cores das imagens capturadas

Histogramas de cores para análise e decisão do estado do semáforo





5.1 Estrutura do Software de controlo

O software de controlo está subdividido em diferentes módulos de software (*units*) que realizam tarefas com objectivos distintos. (por ex: processamento dos sensores).

A informação resultante de cada uma destas unidades é posteriormente fundida e processada para determinar a posição x,y e θ do robot no mundo, sendo depois filtrada numa unidade de decisão que gera os comandos para a actuação nos motores.

Todas as rotinas de controlo de movimento têm como saídas a velocidade tangencial e angular a aplicar ao robot. Posteriormente a velocidade a enviar para cada motor é calculada tendo em conta o tipo de locomoção (neste caso diferencial). Desta forma consegue-se que o controlo seja o mais indiferente possível ao *hardware*.

A sequência de tarefas que rege a evolução do estado do robot no mundo está projectada numa máquina de estados que afecta e é afectada pelo módulo de decisão.

O módulo de decisão e actuação, assim como a máquina de estados projectada serão descritos pormenorizadamente no capítulo 7 deste documento.



As figuras 35 e 36 ilustram a arquitectura do software de controlo do robot.

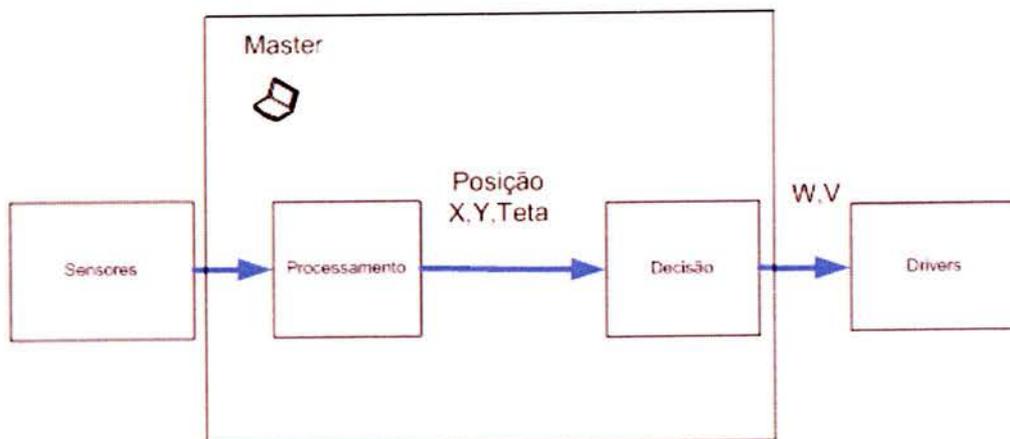


Figura 35 - Esquema Alto-nível do processamento dos dados dos sensores e actuação nos motores

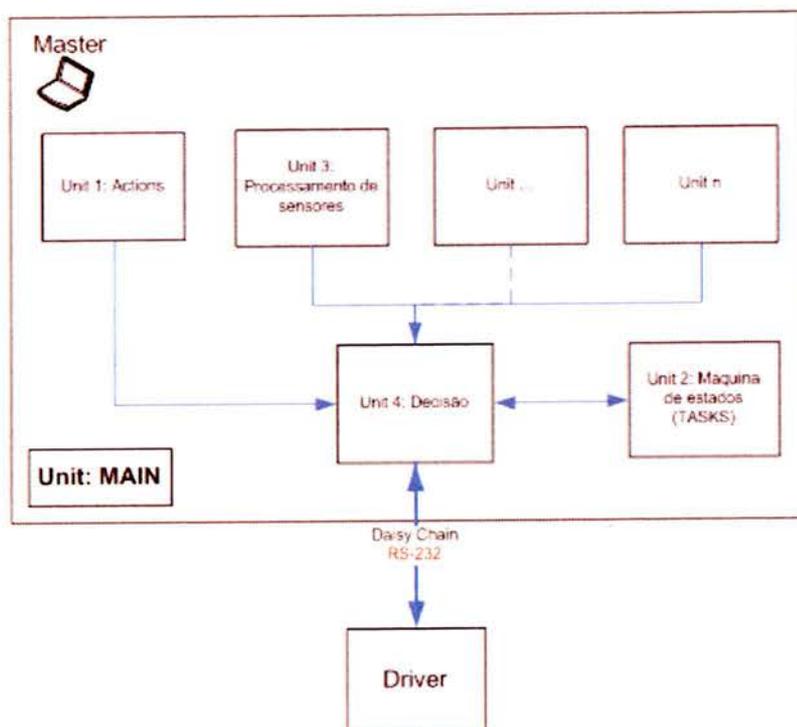


Figura 36 - Esquema geral do Software de controlo



5.2 Planeamento de trajetórias

As rotinas descritas no ponto seguinte foram desenvolvidas tendo em conta o tipo de locomoção do robot.

5.2.1 Equações de movimento do robot

As funções de controlo de movimento desenvolvidas são baseadas nas seguintes equações de movimento do robot:

$$V_1 = V_{motor1} = \frac{impulsos_{encoder_1} * K}{dT}$$

$$V_2 = V_{motor2} = \frac{impulsos_{encoder_2} * K}{dT}$$

$$V_t = \frac{V_1 + V_2}{2}$$

$$W = \frac{V_1 - V_2}{2 * r}$$

$$\theta_f = \theta_i + W * dT$$

$$V_x = V_t * \cos(\theta)$$

$$V_y = V_t * \sin(\theta)$$

$$X_f = X_i + V_x * dT$$

$$Y_f = Y_i + V_y * dT$$

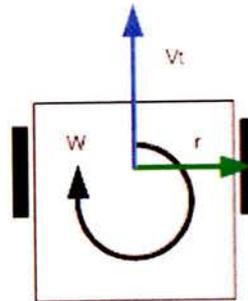


Figura 37 - Modelo do robot

V_t – Velocidade tangencial

W – Velocidade angular

r – Raio do robot

K – Razão de transformação entre velocidade angular do motor e do *encoder*.

5.2.2 Rotinas de controlo de movimento

Foram desenvolvidas as seguintes rotinas de controlo de movimento:

- Action_Gotoxy(Xdestino, Ydestino, W, V);
- Action_FollowLine(Xinicial, Yinicial, Xfinal, Yfinal, W, V);
- Action_FollowCircle(Xcentral, Ycentral, Raio, W, V);
- Action_Rotate(Wref, Ganho, W, V);
- Action_Stop(W, V);
- Action_Acelera(Vref, Vout);



5.2.2.1 Action_Gotoxy (Xdestino,Ydestino,W,V)

Esta função tem como objectivo gerar os valores de velocidade tangencial e velocidade angular a enviar para os motores (W,V) que levem o robot a atingir um determinado ponto X,Y definidos como parâmetros de entrada da função (Xdestino,Ydestino).

5.2.2.2 Action_FollowLine(Xinicial,Yinicial,Xfinal,Yfinal,W,V)

Esta função tem como objectivo gerar os valores de velocidade tangencial e velocidade angular a enviar para os motores (W,V) que levem o robot a seguir um vector definido pelo par de coordenadas (Xinicial,Yinicial) , (Xfinal,Yfinal) definidos como parâmetros de entrada da função.

5.2.2.3 Action_FollowCircle(Xcentral,Ycentral,Raio,W,V)

Esta função tem como objectivo gerar os valores de velocidade tangencial e velocidade angular a enviar para os motores (W,V) que levem o robot a seguir a trajectória de um circulo com um determinado raio, centrada no ponto (Xcentral,Ycentral) definidos como parâmetros de entrada da função.

5.2.2.4 Action_Rotate(Wref,Ganho,W,V)

Esta função tem como objectivo gerar os valores de velocidade tangencial e velocidade angular a enviar para os motores (W,V) que levem o robot a rodar sobre o próprio eixo de rotação a uma velocidade angular de referência (Wref), definida como parâmetro de entrada da função.

5.2.2.5 Action_Stop(W,V)

Esta função tem como objectivo gerar os valores de velocidade tangencial e velocidade angular a enviar para os motores (W,V) que levem o robot a permanecer imóvel.

5.2.2.6 Action_Acelera(Vref,Vout)

Esta função tem como objectivo a implementação de rampas de aceleração, para diminuir os deslizamentos durante as acelerações e compensar a diferença em termos de resposta temporal dos motores. A velocidade de referência (Vref) é o parâmetro de entrada da função.



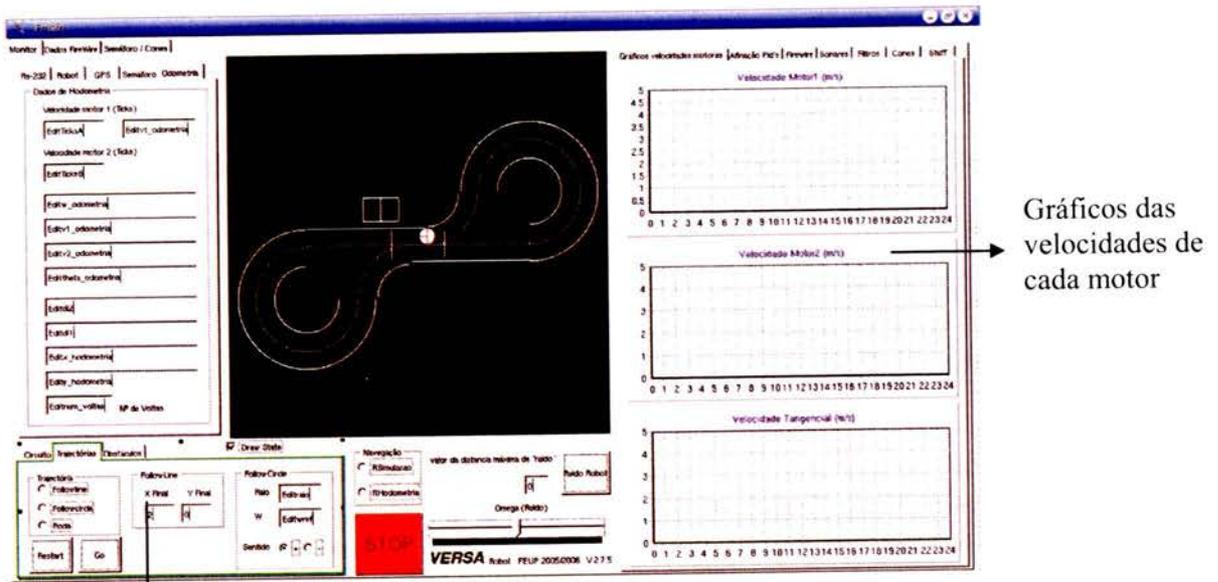


Figura 38 - Validação das rotinas de controlo de movimento

A aplicação desenvolvida permite a escolha de um modo de actuação, denominado “trajectórias”, onde as rotinas anteriormente mencionadas assim como a máquina de estado desenvolvida podem ser validadas.



5.3 Simulador

Para validar o software de controlo de movimento do robot e avaliar o comportamento do mesmo, foi criado um ambiente de simulação integrado na plataforma de controlo do robot.

O ambiente de simulação permite simular os sinais gerados pelos sensores do robot (sonares e câmaras *firewire*), o que possibilita que o comportamento e evolução do estado do robot sejam previstos.

Esta possibilidade permite por isso que para além da validação das rotinas de controlo seja também validada a máquina de estados que rege a evolução do robot no mundo.

O último parágrafo descreve a grande vantagem da utilização da simulação no controlo de robots para navegação em ambientes semi-estruturados, que é o facto de se prever o estado de evolução do robot para um dado conjunto de entradas e com isso verificar a validade da máquina de estados desenvolvida.

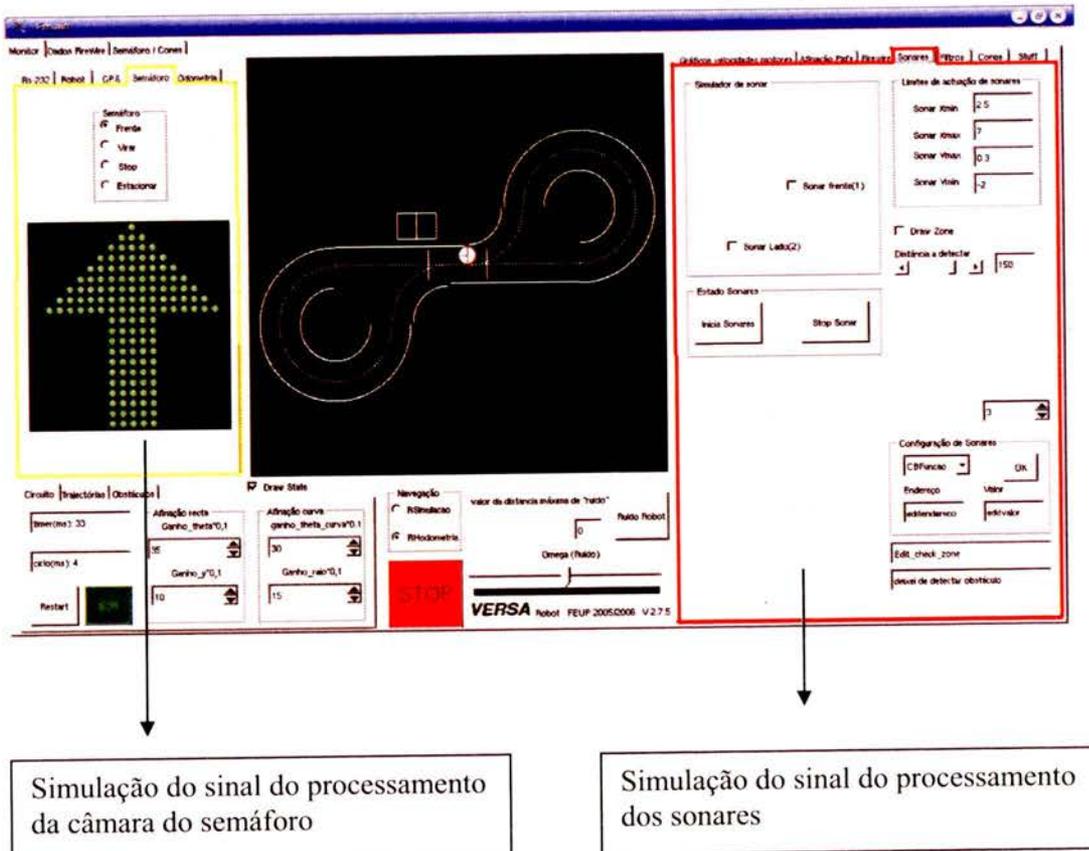


Figura 39 - Simulador do controlo e dos sinais dos dispositivos de sensorização



Capítulo 6

Sistemas de Navegação

Neste capítulo será pormenorizado o processamento efectuado a todos os dispositivos a bordo do robot, responsáveis pela localização do robot no mundo.

6.1 Sistema de visão

Para a captura de imagem recorreu-se à tecnologia *firewire* utilizando os componentes do futebol robótico da equipa 5DPO da FEUP.

Para a localização e navegação do robot são utilizadas duas câmaras *firewire*, uma das câmaras é utilizada para processar as faixas da estrada onde o robot se move, tendo que detectar além das mesmas a presença de passadeiras.

A segunda câmara é utilizada para processar o estado dos sinais de trânsito que o robot terá que respeitar.

Os dos tipos de processamento serão descritos em pormenor nas alíneas 6.2 e 6.3 deste documento.

6.2 *DeBayer* da imagem capturada

Recorrendo à tecnologia *firewire* e uma vez que a câmara utilizada não realiza qualquer tipo de pré-processamento, a imagem capturada chega ao dispositivo de processamento no formato denominado *Bayer*.

Neste formato um pixel é formado pela informação de quatro bytes, cada um contendo informação relativa à componente RGB do pixel.

A figura 40 ilustra o formato *Bayer*, acima descrito:

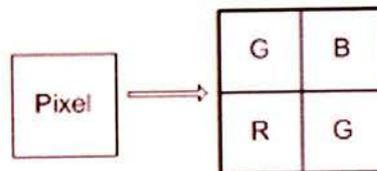


Figura 40 - Formato Bayer

Torna-se por isso necessário desenvolver uma função que realize o processo de guardar num array de inteiros o valor de cada pixel. Este processo é normalmente denominado de *DeBayer*.



A propriedade de um pixel é então dada pelas seguintes equações:

$$Pixel.Green = (G + G) / 2$$

$$Pixel.Red = R$$

$$Pixel.Blue = B$$

Depois de efectuado o processo de *DeBayer*, a imagem capturada é guardada num vector de inteiros tridimensional, com dimensão da resolução que a câmara impõe, neste caso 640 x 320 pixels.

Posteriormente à captura da imagem foi necessário processar a informação proveniente das duas câmaras a bordo do robot.

A resolução das imagens proveniente de cada uma das câmaras é de 640 x 320 pixels, no entanto para o processamento da informação de uma das câmaras e numa perspectiva de optimização do tempo de processamento, foi diminuída para metade por software, uma vez que as câmaras utilizadas não permitem escolher a resolução da imagem capturada.

6.3 Algoritmo de detecção, localização e orientação de formas

Pelo facto das linhas que delimitam a estrada como as da passadeira poderem ser considerados objectos com uma dada forma e diferentes orientações, foi desenvolvido um algoritmo de navegação e localização do robot, que se baseia na análise da forma e orientação dos objectos detectados.

A figura 41 ilustra a estrutura do algoritmo desenvolvido para o processamento de objectos.

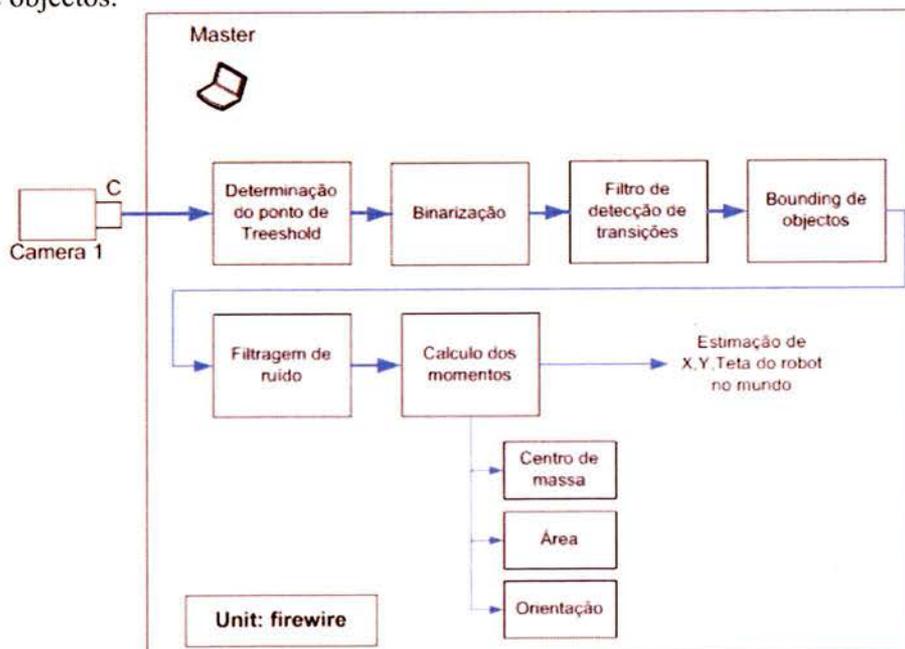


Figura 41 - Esquema Alto-nível do processamento de imagem para a detecção de formas e respectiva orientação



6.3.1 Binarização

O primeiro passo do processamento consiste em transformar a imagem em escalas de cinza, a equação que permite determinar a componente cinza de cada pixel é a seguinte:

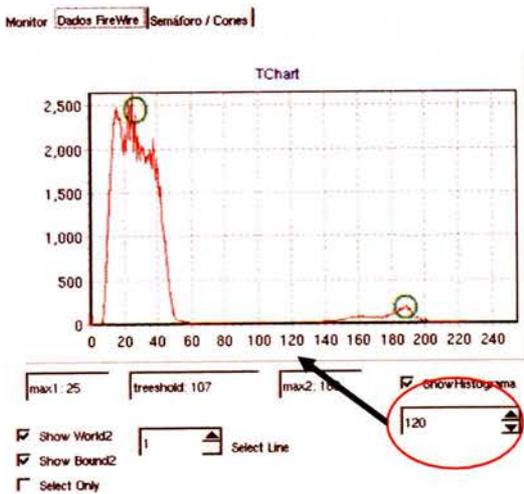
$$Pixel.Gray_{x,y} = 0.3 * Pixel.Red_{x,y} + 0.59 * Pixel.Green_{x,y} + 0.11 * Pixel.Blue_{x,y}$$

Esta função retorna um valor que varia entre 0 e 255 que quantifica a luminosidade de cada pixel.

6.3.1.1 *Threshold* dinâmico

Para determinar o nível de transição e transformar a imagem em escalas de cinza numa imagem com dois níveis de cor, preto e branco, é calculado o ponto de transição (*threshold*) recorrendo à análise do histograma da luminosidade da imagem.

A figura 42 exemplifica o histograma de uma imagem em escalas de cinza.



Na figura ao lado, assinalada a verde é possível observar-se os valores máximos de luminosidade.

Assinalado a vermelho está o valor resultante do cálculo do valor de transição (*threshold*).

Figura 42 - Histograma de luminosidade da imagem capturada

O algoritmo que realiza o histograma da imagem, consiste em varrer todos os pixels da imagem e guardar num vector de inteiros com 255 posições o número de ocorrências de cada um dos valores de luminosidade.

Depois de obtido o histograma de luminosidade é necessário calcular o ponto de transição (*threshold*) para transformar a imagem numa imagem de dois tons, preto e branco.

O algoritmo desenvolvido, detecta os dois máximos do vector e retorna como valor de transição (*threshold*) a média dos dois valores.



A vantagem de se determinar este ponto de forma dinâmica consiste no facto da luz onde a imagem é capturada poder variar, sem que seja necessário ajustar esse ponto de transição (*threshold*) de cada vez que a luz do meio ambiente se altera.

A nova imagem, a preto e branco, fica então definida pela seguinte função:

$$Pixel.BW_{x,y} = \begin{cases} 0 & \text{se } Pixel.Gray_{x,y} < threshold \\ 1 & \text{se } Pixel.Gray_{x,y} > threshold \end{cases}$$

A figura 43 ilustra o exemplo do resultado da binarização de uma imagem capturada.

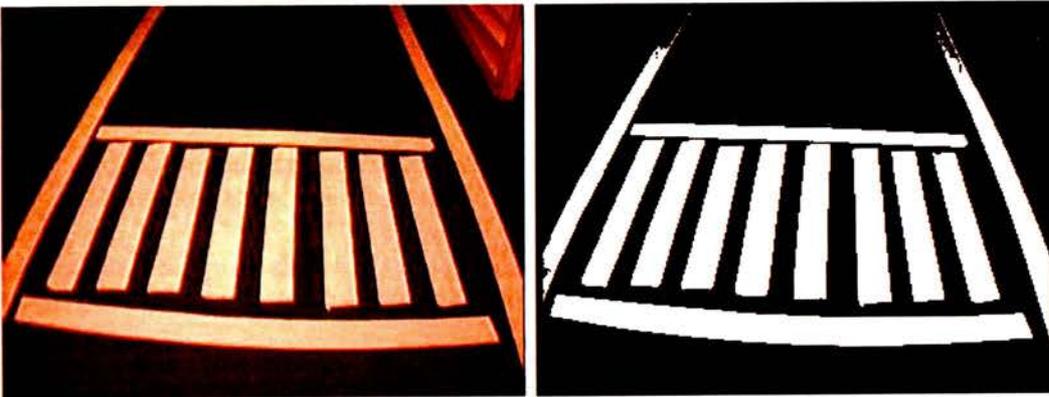


Figura 43 - Exemplo de uma imagem *binarizada*

6.3.2 Filtro de detecção de transições

O passo seguinte do processamento consiste em detectar as transições que permitem localizar os objectos da imagem.

Para tal é aplicado um filtro à imagem binarizada que consiste numa convulsão da mesma com uma matriz que realça as transições horizontais e verticais.

O algoritmo que realiza esta operação é simplesmente a implementação da equação abaixo indicada.

$$C(m,n) = a(m,n) \circ b(m,n) = \sum_{j=-\infty}^{j=\infty} \sum_{k=-\infty}^{k=\infty} a(j,k)b(m-j,n-k)$$

Onde A é a imagem original, B a matriz com o filtro de detecção de transições e C a imagem resultante da convulsão da matriz A com a matriz B.

Para detectar transições foi utilizada a seguinte matriz:

$$B = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Esta matriz devido às suas propriedades realça as transições presente na imagem.



Na figura 44 está ilustrado um exemplo deste algoritmo de processamento.



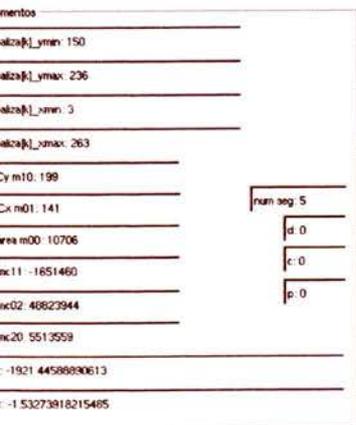
Figura 44 - Exemplo do algoritmo de detecção de transições

6.3.3 Algoritmos de detecção de objectos

Com todos as transições assinaladas é necessário isolar e localizar os diferentes objectos presentes na imagem e para tal recorreu-se à implementação do denominado “algoritmo do pixel vizinho” que consiste em percorrer toda a vizinhança de um pixel até encontrar um pixel de cor idêntica. Com esta técnica é possível percorrer os limites de um dado objecto isolando-os em diferentes áreas da imagem.

6.3.3.1 Balizamento (*Bounding*) de objectos

Como dito na alínea anterior, cada objecto presente na imagem é isolado através da identificação das áreas que cada um ocupa na imagem. Cada uma das áreas identificadas é guardada num array multidimensional com a seguinte estrutura:



Id	Ymin	Ymax	Xmin	Xmax	Cy	Cx	área	mc11	mc02	mc20	t	i
----	------	------	------	------	----	----	------	------	------	------	---	---

- Id – Identificador do objecto
- Cy- Coordenada y do centro de massa do objecto
- Cx- Coordenada x do centro de massa do objecto
- mc_{pq} - Momento centrado de ordem $p+q$
- t – eixo de inércia mínima
- i – Orientação do objecto

Nota: O cálculo destes parâmetros será explicado na alínea 6.2.4 deste documento.

Figura 45 - Menu com o cálculo de momentos da imagem

Na figura 45 é possível observar-se o resultado do processamento que efectua o balizamento (*bounding*) aos objectos presentes na imagem. Os campos visíveis na figura são guardados em memória num array.



Os parâmetros Y_{min} , Y_{max} , X_{min} e X_{max} definem a área dentro da imagem em que o objecto se encontra.

A partir daqui todo o processamento é realizado dentro das áreas anteriormente isoladas e por isso o correcto balizamento (*bounding*) dos diferentes tipos de objectos presentes na imagem é um processo muito importante uma vez que permite o processamento localizado, que se reflecte na optimização do tempo de processamento da imagem.

No entanto como a imagem captada está sujeita a ruído verifica-se que para além dos objectos que são realmente necessários analisar, são também identificados objectos que são fruto do ruído introduzido na imagem.

A análise destas zonas de ruído constitui por isso um desperdício de processamento que afecta todo o processo de controlo.

Um exemplo do ruído (assinalada a vermelho) introduzido na imagem está ilustrado na figura 46.

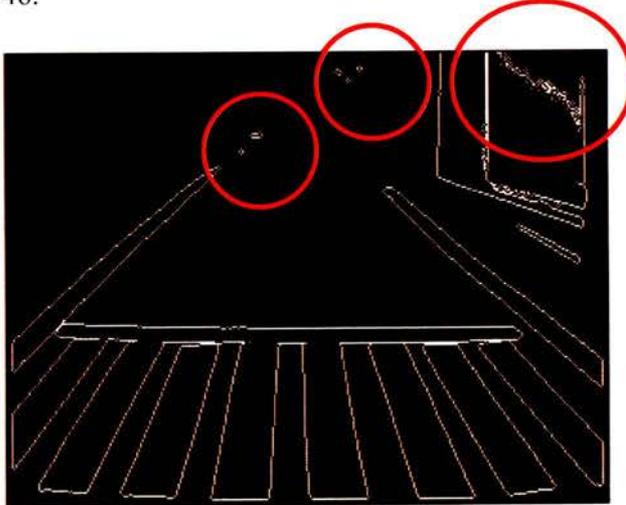


Figura 46 - Imagem com ruído

É por isso necessário efectuar uma filtragem das áreas de processamento identificadas, processo que será pormenorizado na alínea seguinte.

6.3.4 Cálculo dos momentos centrados da imagem e filtragem de ruído

Para filtrar e isolar as zonas da imagem anteriormente identificadas como objectos, recorreu-se à análise de momentos dos mesmos, que permite, entre outros parâmetros, determinar o centro de massa, a área e a orientação de cada um dos objectos.

Com estes parâmetros é então possível implementar um algoritmo capaz de filtrar as áreas de ruído.

A seguinte equação permite calcular os momentos de Ordem N de uma imagem.

$$m_{pq} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} x^p y^q f(x, y)$$

$$f(x, y) = \frac{\text{Pixel.BW}_{x,y}}{1}, \text{ f}(x,y) \text{ é uma função que retorna 0 ou 1}$$

6.3.4.1 Área (momento de ordem 0)

O algoritmo que detecta a área dos objectos recorre à implementação da seguinte equação matemática:

$$m_{00} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} x^0 y^0 f(x, y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y)$$

6.3.4.2 Centro de massa (momento de ordem 1)

O algoritmo que detecta o centro de massa dos objectos recorre à implementação da seguinte equação matemática:

$$Cx = m_{10} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} x^1 y^0 f(x, y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} x * f(x, y), \text{ Cx é a coordenada x do pixel central.}$$

$$Cy = m_{01} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} x^0 y^1 f(x, y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} y * f(x, y), \text{ Cy é a coordenada y do pixel central.}$$



6.3.4.3 Orientação

O algoritmo que detecta a orientação dos objectos recorre à implementação da seguinte equação matemática:

$$I = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} ((x-a) \sin(t) - (y-b) \cos(t)) * f(x,y)$$

$$a = Cx \quad b = Cy \quad t = \frac{1}{2} \arctg\left(2 \frac{mc_{11}}{mc_{20} - mc_{02}}\right)$$

6.3.4.4 Filtragem de ruído

A filtragem recorre à informação dos parâmetros anteriores e o algoritmo que a implementa apresenta a seguinte estrutura:

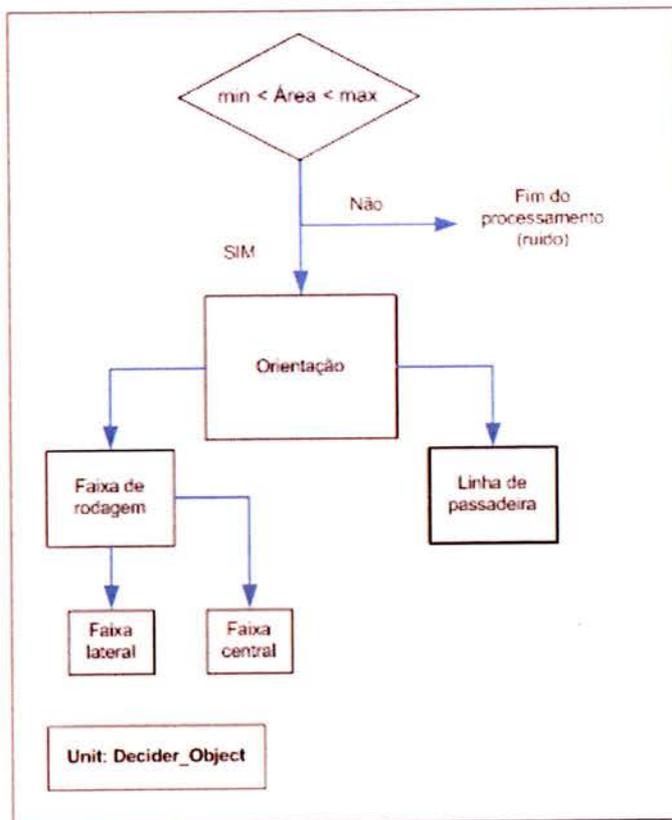


Figura 47 - Estrutura da filtragem de objectos

O primeiro factor de filtragem é a área do objecto identificado, se este parâmetro estiver dentro de uma dada gama é então considerada como um objecto, caso contrário é considerado ruído ou um objecto que não interessa processar.



6.4 Decisão e atribuição de formas

A decisão e atribuição de formas a cada uma dos objectos identificados é o passo seguinte à filtragem e segue a estrutura ilustrada na figura 47.

Neste caso os objectos filtrados podem ser identificados como sendo linhas da faixa de rodagem ou linhas da passadeira.

O parâmetro que distingue a faixa de rodagem de uma linha de passadeira é a orientação do objecto, uma vez que a linha de passadeira tem a característica única de ter uma inclinação perpendicular à orientação do robot.

No caso de ser identificado como uma faixa de rodagem, a decisão se é uma faixa lateral ou se é a uma faixa central é tomada tendo em conta a inclinação e o centro de massa do objecto.

Desta maneira são atribuídas formas aos objectos identificados na imagem e o balizamento (*bounding*) apresenta diferentes cores que distinguem as formas identificadas:

- Amarelo – Linha de passadeira
- Vermelho – Faixa Lateral
- Azul – Faixa Central

A figura 48 exemplifica o processo de identificação e balizamento (*bounding*) com diferentes cores.



Figura 48 - Imagem do resultado do balizamento (*bounding*)

6.5 Determinação da localização das formas no mundo

Para determinar a localização das formas identificadas no mundo, foi implementado um algoritmo que realiza o mapeamento da imagem no mundo e que permite saber com precisão qual a posição relativa ao robot das formas identificadas.

Para tal recorre-se a uma função que implementa o método simplificado de Tsai que recebe como parâmetros de entrada a altura a que a câmara se encontra, a distancia focal, o ângulo em relação ao plano de focagem e o factor de correcção do efeito em barril.

Com estes parâmetros e através de uma translação é determinada em coordenadas do referencial relativo ao robot a posição das formas identificadas.

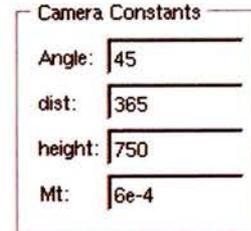


Figura 49 - Menu de afinação de parâmetros de mapeamento da imagem para coordenadas do mundo

A figura 50 ilustra o processo de mapeamento da imagem capturada que a função implementa:

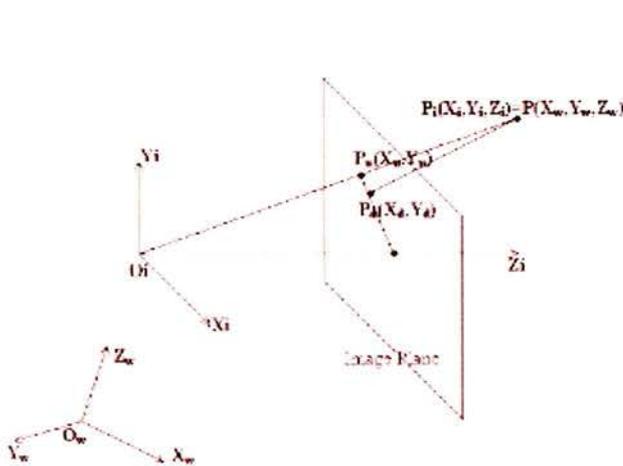


Figura 50 - Método de Tsai

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T \quad R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \quad T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

$$\begin{aligned} r_1 &= \cos(R_y) \cos(R_z) \\ r_2 &= \cos(R_z) \sin(R_x) \sin(R_y) - \cos(R_x) \sin(R_z) \\ r_3 &= \sin(R_x) \sin(R_z) + \cos(R_x) \cos(R_z) \sin(R_y) \\ r_4 &= \cos(R_y) \sin(R_z) \\ r_5 &= \sin(R_x) \sin(R_y) \sin(R_z) + \cos(R_x) \cos(R_z) \\ r_6 &= \cos(R_x) \sin(R_y) \sin(R_z) - \cos(R_z) \sin(R_x) \\ r_7 &= -\sin(R_y) \\ r_8 &= \cos(R_y) \sin(R_x) \\ r_9 &= \cos(R_x) \cos(R_y) \end{aligned}$$

R_x, R_y, R_z – Ângulo de rotação para a translação entre as coordenadas do mundo e as coordenadas da câmara.

T_x, T_y, T_z – Componentes de translação para a transformação entre as coordenadas do mundo e as coordenadas da câmara.

A função que implementa o método acima descrito foi desenvolvida pela equipa de futebol robótico, 5DPO, da FEUP.



6.6 Algoritmo de actualização do estado do robot no mundo

Para determinar a posição do robot no mundo através dos dados fornecido pelo sistema de visão, foi implementado um algoritmo que utiliza o método descrito na alínea anterior, no entanto para determinar a posição absoluta do robot é necessário mais algum processamento uma vez que o algoritmo anterior fornece posições relativas.

A posição absoluta do robot no mundo é determinada através das medições da posição do robot relativamente às faixas de rodagem, seguidas de um processo de rotação e translação.

Nas zonas de recta, através do sistema de visão é actualizada a posição y e θ do robot.

$$\text{A posição em } y = \frac{(y1 + y2)}{2}$$

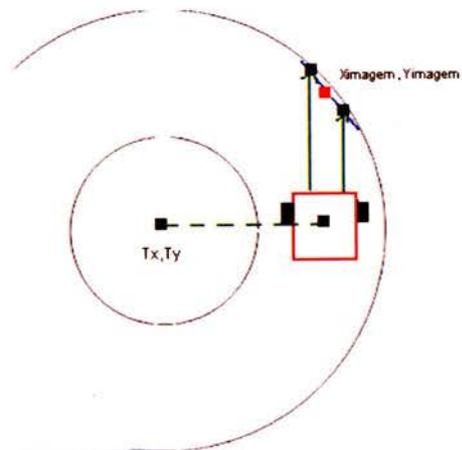
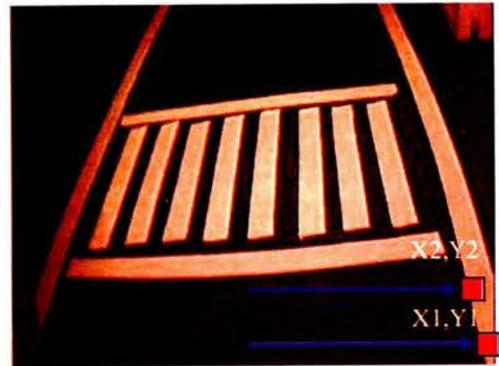
$$\text{A posição em } \theta = \arctan\left(\frac{y2 - y1}{x2 - x1}\right)$$

Nas zonas de curva é actualizada a posição x e y do robot, através de uma rotação, seguida de uma translação.

$$\begin{bmatrix} X_{mundo} \\ Y_{mundo} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} * \begin{bmatrix} X_{imagem} \\ Y_{imagem} \end{bmatrix} + \begin{bmatrix} Tx \\ Ty \end{bmatrix}$$

Onde T_x e T_y , são os pontos centrais da circunferência em que o robot se encontra.

θ é dado pelo processamento dos dados de hometria, uma vez que o calculo de θ nas zonas de curva apresentava erros que não podiam ser desprezados, fruto da aproximação da curva a segmentos de recta .



6.7 Algoritmo de processamento do sinal de semáforo

O processamento do semáforo recorre ao mesmo algoritmo descrito nos parágrafos anteriores, no entanto os dados resultantes da análise de momentos da imagem, são usados com objetivos diferentes. Neste caso, a cor do semáforo é determinada pelo centro de massa de todos os objectos presentes na imagem, ou seja, a decisão do estado do semáforo tem em conta a cor que predomina em todos os pixels analisados e cuja coordenada é o resultado da análise dos centros de massa.

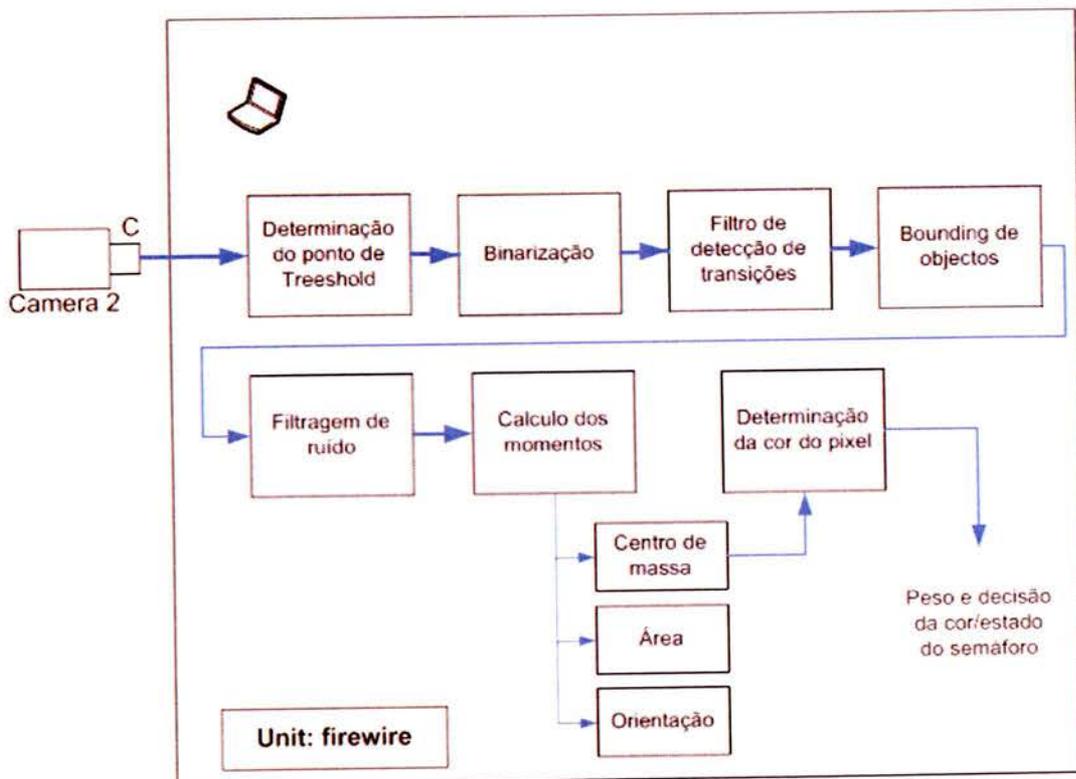


Figura 51 - Esquema Alto-nível do processamento de imagem do semáforo



A figura 52 ilustra um exemplo do processamento do semáforo.

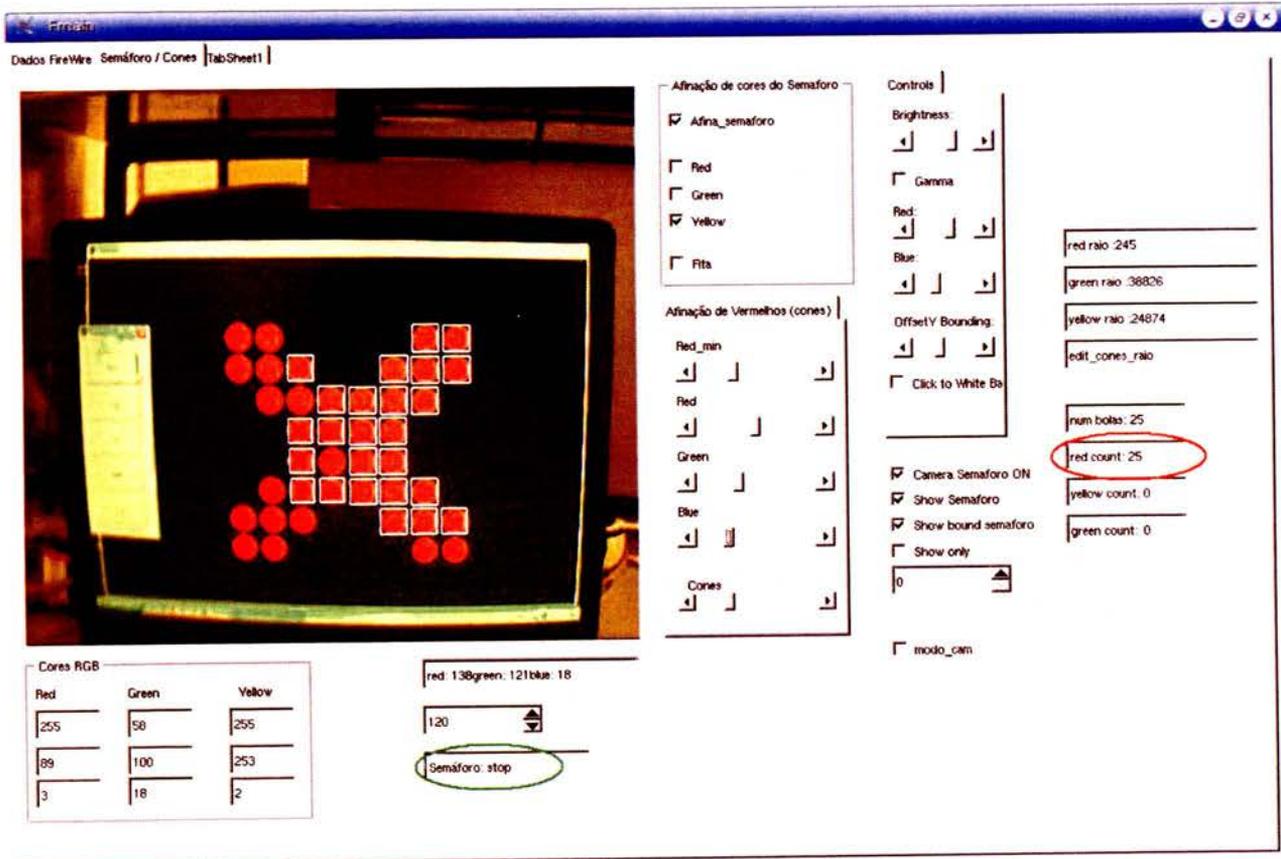


Figura 52 – Resultados do processamento do sinal de semáforo

Na figura é possível observar-se o resultado do processamento efectuado ao semáforo. A vermelho está assinalado o número de balizamentos (*boundings*) efectuados assim como a cor atribuída a cada um deles, neste caso vermelho.

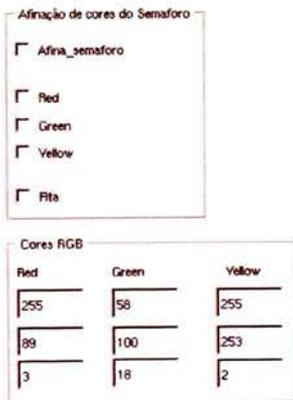
O estado atribuído ao semáforo está assinalado a verde, neste caso o estado stop.

A figura 52 ilustra um caso onde, devido à variação de luminosidade do meio de captura de imagem, não é feito o balizamento (*bounding*) total do sinal de paragem. No entanto, através deste método de processamento, a decisão do estado do semáforo é realizada de forma correcta, o que seria mais difícil de implementar se o algoritmo se baseasse na detecção e análise de formas, uma vez que o objecto não é detectado na sua totalidade.

6.7.1 Afinação de cores

Para uma correcta decisão do estado do semáforo é necessário que as cores da imagem capturada estejam afinadas, ou seja, é necessário que o algoritmo de processamento saiba distinguir entre diferentes cores (verde, vermelho e amarelo) e quais os limites para que a propriedade de um dado pixel possa ser identificado como uma dessas cores.





Na aplicação desenvolvida, o utilizador tem que escolher a cor desejada e clicar num pixel da imagem que deseja associar a uma determinada cor.

Para cada cor escolhida (verde, vermelho e amarelo) é criada uma estrutura com as propriedades RGB do pixel seleccionado pelo utilizador.

Figura 53 - Menu de afinação de cores

Para cada cor definida pelo utilizador é então criada a seguinte estrutura:

Vermelho	Verde	Amarelo
Red.Red	Green.Red	Yellow.Red
Red.Green	Green.Green	Yellow.Green
Red.Blue	Green.Blue	Yellow.Blue

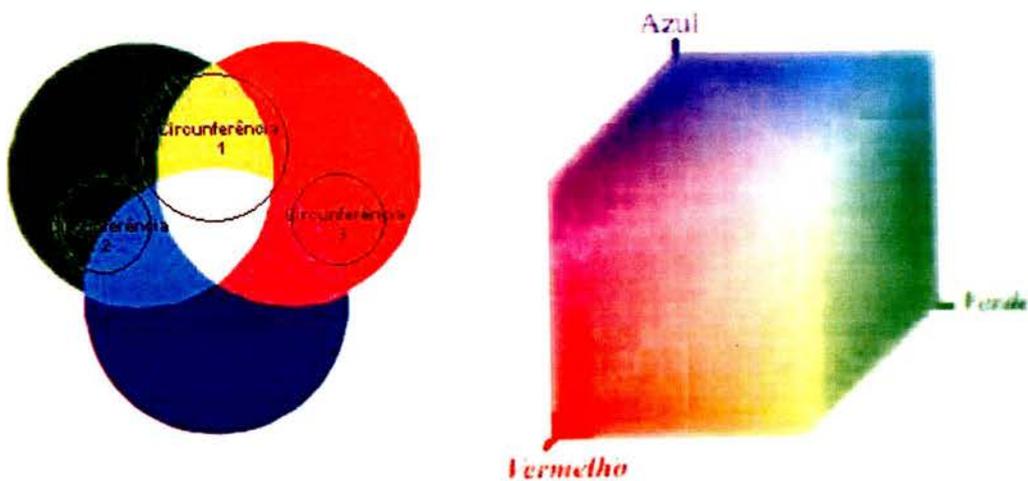


Figura 54 - Componentes RGB de uma cor

O algoritmo desenvolvido para a determinação do estado do semáforo, verifica se as propriedades RGB do pixel estão dentro de alguma das esferas, recorrendo à implementação das seguintes equações matemáticas:

Para que seja atribuída a um pixel a cor amarela, a seguinte equação tem que ser respeitada:

$$\left(Raio_{esfera\ 1} \right)^2 \geq \left(pixel.Red_{x,y} - Yellow.red \right)^2 + \left(pixel.Green_{x,y} - Yellow.Green \right)^2 + \left(pixel.Blue_{x,y} - Yellow.Blue \right)^2$$

Para que seja atribuída a um pixel a cor verde, a seguinte equação tem que ser respeitada:

$$\left(Raio_{esfera\ 2} \right)^2 \geq \left(pixel.Red_{x,y} - Red.red \right)^2 + \left(pixel.Green_{x,y} - Red.Green \right)^2 + \left(pixel.Blue_{x,y} - Red.Blue \right)^2$$

Para que seja atribuída a um pixel a cor vermelha, a seguinte equação tem que ser respeitada:

$$\left(Raio_{esfera\ 3} \right)^2 \geq \left(pixel.Red_{x,y} - Green.red \right)^2 + \left(pixel.Green_{x,y} - Green.Green \right)^2 + \left(pixel.Blue_{x,y} - Green.Blue \right)^2$$

O raio de cada circunferência é também definido pelo utilizador na aplicação.



Figura 55 - Menu de afinação de raios



6.7.2 Protocolo de comunicação Master PC – Slave PC

Para que o PC portátil onde é feito a fusão e decisão de informação tenha acesso ao estado do semáforo, processado no PC portátil secundário, foi implementado um protocolo simples de comunicação via rede *ethernet* do tipo master-slave.

Quando o robot chega à zona de análise do semáforo, o PC principal envia uma pergunta ao PC secundário com a requisição do estado do semáforo.

Para prevenir que o PC principal fique infinitamente à espera de uma resposta, foi implementado um mecanismo de *time-out* que repete a pergunta até um número finito de vezes.

No caso do PC secundário não responder, o PC principal assume por defeito que o semáforo indica para seguir em frente.

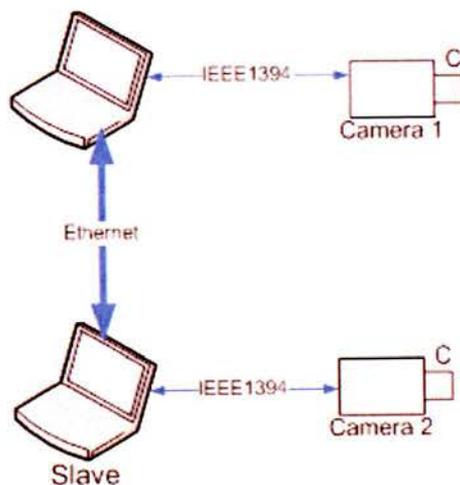
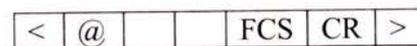


Figura 56 - Esquema da comunicação Master-Slave PC

Estrutura da trama do protocolo de comunicação:



Bytes para o comando de pergunta/resposta

CR – Caracter de terminação
FCS – *Frame Check Sequence*

Comandos:

TT - Pergunta: estado do semáforo
SS - Resposta: Stop
FF - Resposta: Frente
LL - Resposta: Virar



6.8 Hodometria

A hodometria é uma técnica para medir distâncias percorridas. Sabendo a inclinação, o ponto de partida e a distância percorrida por cada roda é possível conhecer o estado do robot, ou seja, a sua posição no mundo.

Para que o robot se localize no mundo e conheça com precisão as suas velocidades, é necessário que a hodometria esteja bem afinada.

6.8.1 Afinação

A afinação da hodometria é um processo extremamente importante. A sua incorrecta afinação causa erros grosseiros ao fim de alguns metros percorridos devido ao erro ser acumulativo. Para se proceder à sua afinação é necessário ajustar três parâmetros. O raio da roda direita, da esquerda e o raio do robot. Este último não influencia nos movimentos translação, portanto deve-se inicialmente ajustar o raio das rodas.

Começa-se a afinação pondo o robot a andar em frente, por exemplo 4 metros. De seguida lê-se no programa de controlo o valor da distância percorrida por cada uma das rodas. Afina-se então o valor do raio da roda direita e esquerda para que os valores da distância percorrida por cada uma das rodas seja o mais próximos possível de 4 metros.

Este processo pode ter de ser repetido várias vezes até que os valores dos raios das rodas fiquem bem afinados.

Finalmente, já com os raios das rodas bem calibrados, procede-se ao ajuste do raio do robot. Para tal ordena-se que o robot realize movimentos circulares sobre ele próprio. Inicialmente dá-se meia volta e ajusta-se o raio de forma a que o θ seja aproximadamente 180° . De seguida ordena-se que o robot dê uma volta completa e volta-se a ajustar o raio do robot de forma a que θ seja próximo de zero.

Pode-se considerar afinado se o robot efectuar uma circunferência completa e voltar ao ponto de partida. Durante este processo a velocidade deve ser baixa para evitar que o robot derrape e para minimizar os efeitos inerciais.



6.8.1.1 Análise de erros da hometria

Com a necessidade de avaliar a afinação da hometria, realizaram-se vários testes. Cada teste foi repetido quarenta vezes. O primeiro teste consistiu em andar com o robot quatro metros em frente e confrontar a distância medida pelo robot com a real. Verificou-se que em média o robot percorria 4,15 metros com o desvio padrão de 13 centímetros. O erro percentual é de 3%.

A figura a baixo ilustra as medidas obtidas.

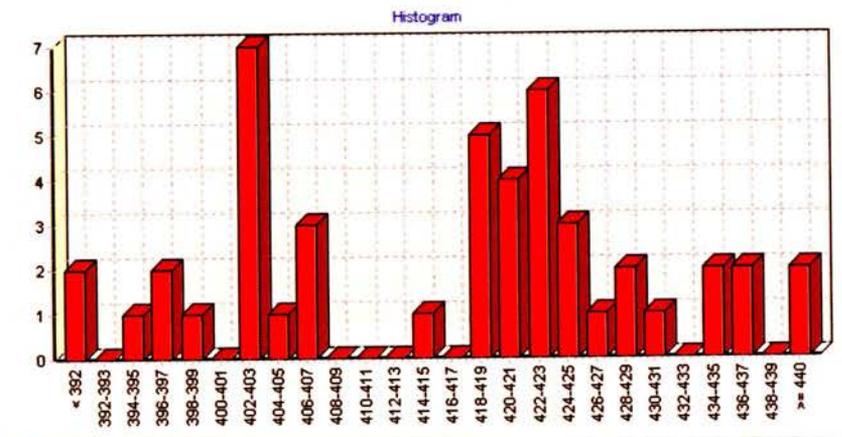


Figura 57 - Histograma de medidas de translação

Foi também efectuada uma prova para a rotação. Esta consistiu em rodar 360° o robot e comparar com o valor medido pelo robot. Neste caso verificou-se que o valor médio foi 369° com o desvio padrão de 14° . O erro percentual é de 2%. A figura a baixo ilustra as medidas obtidas:

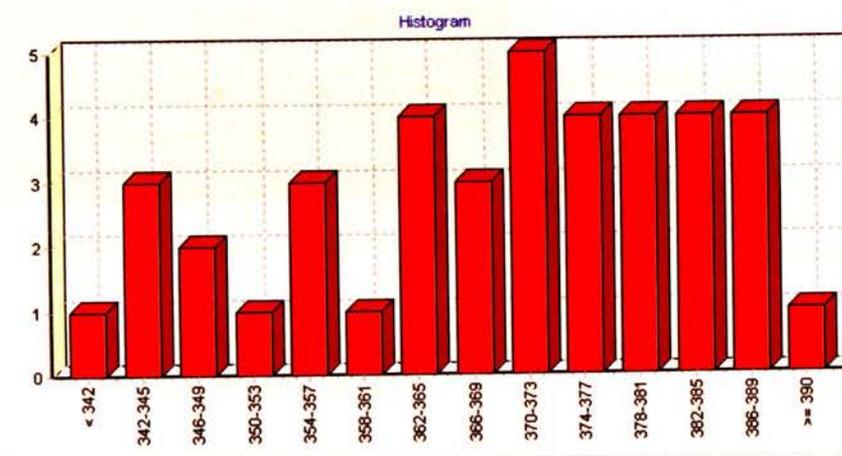


Figura 58 - Histograma de medidas de rotação

Pode-se concluir que o erro da hometria é reduzido, mas não desprezável pois é acumulativo. É de salientar o perigo do erro do ângulo do robot pois dá a informação errada quanto à direcção, originando erros consideráveis na localização do robot.



6.8.2 Algoritmo de minimização do erro de hodometria

Uma vez que a hodometria introduz no sistema um erro de posição que se acumula ao longo do tempo, foi desenvolvido um algoritmo com o objectivo de minimizar esse erro.

A minimização do erro introduzido pela hodometria, recorre à informação disponibilizada pelas linhas da passadeira uma vez que a localização e orientação da mesma é conhecida.

Quando o algoritmo de processamento detecta a presença de uma linha de passadeira é enviado um comando de velocidade tangencial nula para o robot. Nesse momento o movimento do robot é apenas angular e baseado no processamento da orientação da linha de passadeira.

O controlo envia comando para os motores até que a posição angular do robot seja perpendicular à orientação da linha de passadeira. Desta forma o erro introduzido na posição angular do robot é teoricamente anulado.

Para minimizar o erro de posição em x do robot é medida a distancia à linha de passadeira e actualizada a posição em x do robot no mundo.

Para minimizar o erro de posição em y do robot é utilizado o mesmo método descrito no parágrafo anterior, no entanto é utilizada a medida de distância à linha lateral para actualizar a posição em y do robot no mundo.

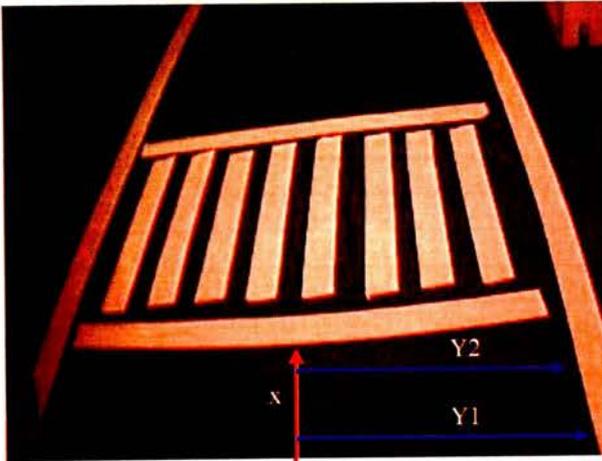


Figura 59 - Actualização do estado no robot no mundo

Na figura é possível observar-se que a posição em x do robot no mundo é actualizada pela distância à linha da passadeira (assinalado a vermelho).

A posição em y é actualizada pela distância à faixa lateral (assinalado a azul).

$$y = \frac{y_2 - y_1}{2}$$

A inclinação θ do robot é dada pela inclinação da linha da passadeira.

Ao actualizar desta forma os três parâmetros que definem o estado do robot no mundo, minimiza-se o erro de posição acumulado pela hodometria.

As distâncias (Y2, Y1 e X) resultantes do processamento de imagem, já contemplam a correcção do efeito em barril.



6.8.3 Protocolo de comunicação o entre *drivers* e sistema de controlo

A comunicação entre o controlo e os periféricos é realizada através de uma *Serial Daisy-Chain*. As tramas enviadas entre os dispositivos têm a seguinte estrutura:

< n N I O ma [dataIn1] [[dataIn2]...] [dataOut1] [[dataOut2]...] >

As tramas iniciam e acabam com “<” e “>” respectivamente. O número do nó é “n”.

O valor de “n” é zero quando enviado pelo controlo, sendo incrementado de cada vez que passa por um *driver*.

“N”, é o modo de comunicação, este pode tomar o valor de B (*Broadcast*) ou C (*compact*).

O número de *bytes* de entrada é representado com “I” sendo os de saída com “O”.

A mascara, “ma” está compreendida entre 0 e FF sendo a forma de se seleccionar os *drivers* a que o comando se destina.

Os campos “IN” e “OUT” contêm os dados de entrada/saída.

O primeiro *byte* de entrada é o “tipo de pacote”, este campo contém a informação sobre a que tipo de dado a trama se refere, encoders(09), sonares(03), valores em pwm para o controlador em malha aberta (01) ou a referência de velocidade para o controlador em malha fechada (06).

<	n	N	I	O	ma	IN	OUT	>	
<	0	C	2	1	01	[03][00]	[00]	>	Enable sonar
<	0	C	2	1	01	[03][01]	[00]	>	Disable sonar
<	0	C	2	8	01	[03][00]	[00][00][00][00]x2	>	Pede leitura
<	0	C	4	2	01	[03][01][END][VAL]	[00][00]	>	Altera ganho
<	0	C	4	2	01	[03][02][END][VAL]	[00][00]	>	Altera alcance
<	0	C	3	0	0F	[01][SEN][VAL][01][SEN] [VAL]		>	Envia PWM
<	0	C	3	0	0F	[06][ls][MS][06][LS][MS]		>	Envia Velocidade
<	0	B	1	6	0F	[09]	[00][00][00][00][00][00]x2	>	Read encoders

Legenda:

END.: endereço

VAL.: valor

SU.: sentido de rotação

LS.: bit menos significativo

MS.: bit mais significativo



6.9 Sonares

Os sonares são utilizados para a detecção e desvio de obstáculos.

O robot possui cinco sonares, dois frontais e três laterais. No caso de um dos sonares frontal detectar a presença de um obstáculo, a máquina de estados que controla o robot, evolui para estado em que impõe ao robot o seguimento de uma trajectória pré-definida até que os sonares laterais deixem de detectar a presença do obstáculo, ou por redundância, se o robot andar mais do que um numero limite de metros na faixa contrária de rodagem, uma vez que a dimensão máxima do obstáculo é conhecida.

A trajectória pré-definida consiste em acrescentar um offset à referência em y que o robot deve seguir. Desta forma o robot passa a circular na faixa contrária de rodagem até ultrapassar o obstáculo.

Quando o robot muda para a faixa de rodagem contrária só se localiza com os dados provenientes do processamento da hodometria.

A figura 60 ilustra o método descrito no parágrafo acima.

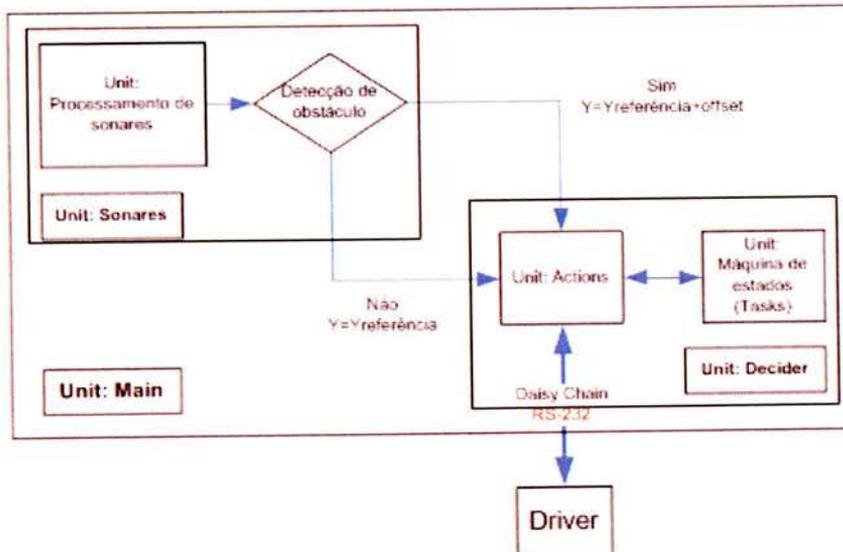


Figura 60 - Esquema alto-nível actuação em caso de detecção de obstáculos

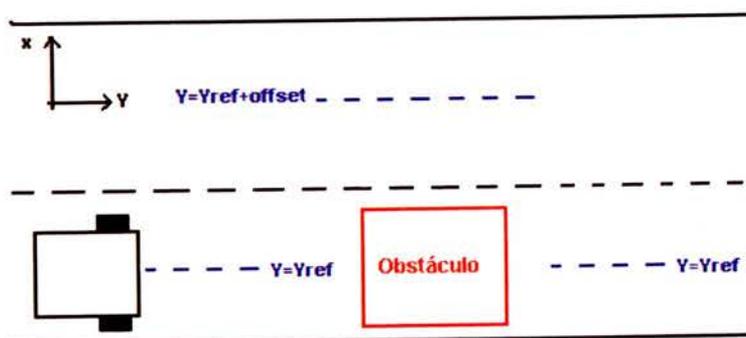


Figura 61 - Detecção e desvio de obstáculos



Capítulo 7

Fusão, filtragem de informação e Decisão

7.1 Fontes de informação

As fontes de informação que permitem calcular a posição do robot no mundo são o sistema de visão e a hodometria.

Os dados provenientes das duas fontes de informação passam por uma primeira etapa de filtragem para eliminar possíveis más medições.

No caso destes dados serem muito diferentes é dada unicamente importância a aos dados da hodometria, caso contrário a localização do robot é calculada atribuindo diferentes pesos às fontes de informação, variáveis para diferentes zonas da pista.

A folga que permite determinar se os dados provenientes das duas fontes de informação são ou não coerentes é ajustada pelo utilizador na aplicação (ver figura 62).

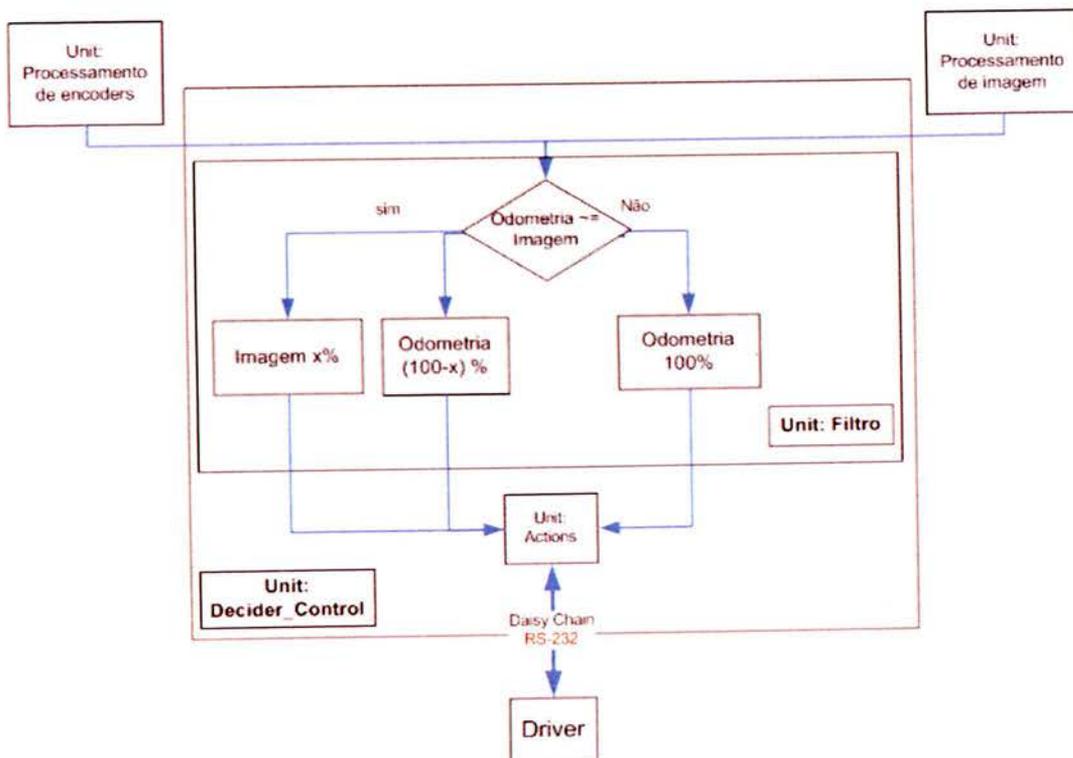


Figura 62 - Esquema da unidade de filtragem e decisão



7.2 Filtragem e decisão

O algoritmo de filtragem e decisão, atribui diferentes pesos às diversas fontes de informação, que variam consoante a posição do robot no mundo.

O peso atribuído a cada fonte de informação para as diferentes zonas do ambiente onde o robot se desloca é definido pelo utilizador.

Nas zonas de recta optou-se por atribuir um elevado peso à informação proveniente do sistema de visão, uma vez que o resultado do processamento se revelou bastante preciso.

Nas zonas de curva optamos por não efectuar um *matching* de curvas, mas antes por aproximar a curva a um conjunto de segmento de rectas.

Esta aproximação acabou por apresentar resultados que em certas zonas da curva se reflectem em erros de posição de alguns centímetros, facto pelo qual decidimos nestas zonas atribuir igual peso à informação resultante do sistema de visão e do sistema de odometria.

A figura 63 ilustra a zona da interface desenvolvida onde o utilizador tem acesso à configuração dos parâmetros do filtro de informação.

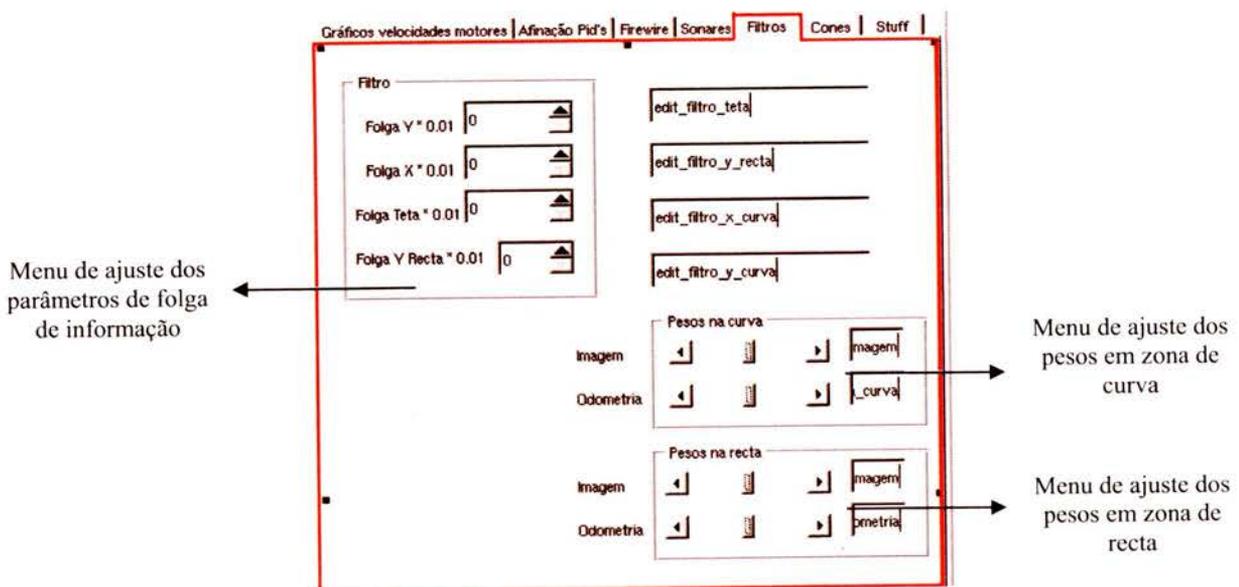


Figura 63- Menu de afinação de parâmetros do filtro



Capítulo 8

Projecto de um sistema de carga rápida

O objectivo deste projecto consiste em implementar um sistema de carga rápida para baterias nickel-metal para carregar as baterias do Robot desenvolvido, assim como as baterias dos robots das equipas de futebol robótico, 5DPO, da FEUP.

Esta necessidade surge do facto de só existir um carregador de baterias para todos os robots que existem no departamento do 5DPO. Em alturas de competição é sentida a necessidade de mais recursos pois, por vezes, as equipas de futebol estão em pavilhões separados e a partilha de um carregador torna-se um processo complicado.

8.1 Análise de requisitos

O sistema de carga deve apresentar a seguintes características:

- Carregar a bateria a carga constante de 2000 ma/h ou 3000 ma/h.
- Monitorização da tensão, corrente e temperatura da bateria, localmente (no próprio carregador) ou remotamente (PC).
- Controlo do processo de carga baseado nas medias de tensão, corrente e temperatura na bateria, localmente ou remotamente.

As curvas de carga características para este tipo de baterias são as seguintes:

(1) Corrente de carga rápida	1Cma
(2) <i>Rapid charge transition voltage restoration current</i>	0.2 - 0.3Cma
(3) Tensão inicial de carga rápida	0.8v/célula
(4) Tensão de terminação de carga	1.8v/célula
(5) $-\Delta V$	5 - 10 mv/célula
(6) $\frac{dT}{dt}$	1 - 2°C/min
(7) Máximo valor de temperatura	60°C
(8) Tempo inicial de não detecção de fim de carga	5 - 10 min
(9) <i>trickle current</i>	0.03- 0.05Cma
(10) Tempo de transferência de carga rápida	60 min
(11) Tempo de carga rápida	90 min
(12) Tempo total	10 -20 h
(13) Excursão de temperatura para cargas rápidas	0 - 40°C

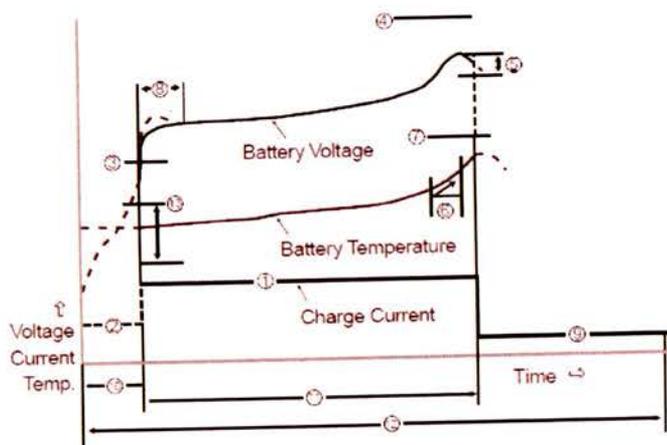


Figura 64 – Curvas características de um sistema de carga rápida



Ao observar as curvas de carga acima representadas, foram definidas as seguintes condições de paragem do processo de carga:

- Se a temperatura exceder os 45°C
- Se a tensão da bateria exceder os 1.8V/célula
- Se a carga da bateria diminuir ao ritmo de 25mv por instante de amostragem
- Se a temperatura da bateria aumentar ao ritmo de 2°C/min

8.2 Especificação do sistema de desenvolvido

Para o controlo do processo de carga das baterias é utilizado um atmega16 que apresenta as seguintes características:

- 32 canais de entrada/saída digitais
- 8 canais de conversores analógico-digital
- 4 canais de PWM
- 1 canal para comunicação 1-wire
- 1 UART para comunicação via RS-232

A escolha por este microcontrolador recaiu no facto de ser uma tecnologia já conhecida, que cumpre com os requisitos necessários para implementação do sistema.

Para a monitorização da temperatura na carga, foi escolhido o sensor DS18B20, que dispõe de uma saída para comunicação através de um barramento 1-wire.

Esta característica permite, através de um protocolo estabelecido pelo fabricante, aceder ao valor de temperatura assim como a parâmetros de configuração do próprio sensor, como por exemplo a configuração de alarmes no caso da temperatura exceder um valor limite definido pelo utilizador.

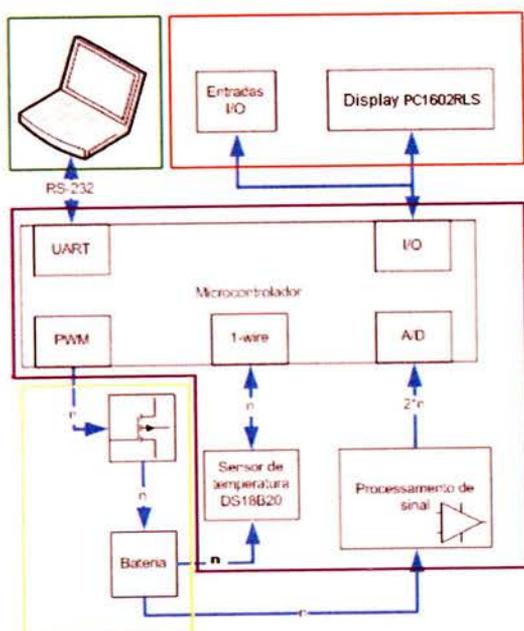


Figura 65 - Esquema alto-nível do carregador de baterias

Assinalado a vermelho e a verde estão os blocos onde são feitas, respectivamente, as interfaces local e remota com o utilizador. Assinalado a amarelo está o bloco de electrónica de potência. A roxo está assinalado o bloco de processamento de sinal e controlo do circuito de potência.

O esquema eléctrico projectado para cada um dos blocos assinalados será pormenorizado nas alíneas seguintes.



8.3 Dimensionamento e implementação

8.3.1 Circuito de monitorização e comando

8.3.1.1 Interface local

A interface local com o utilizado é realizada através de um LCD, referência PL1602LRS e um conjunto de interruptores que permitem ao utilizador visualizar o estado da carga e os dados resultantes do processo de carga de cada uma das baterias.

A figura 65 ilustra o esquemático da interface local.

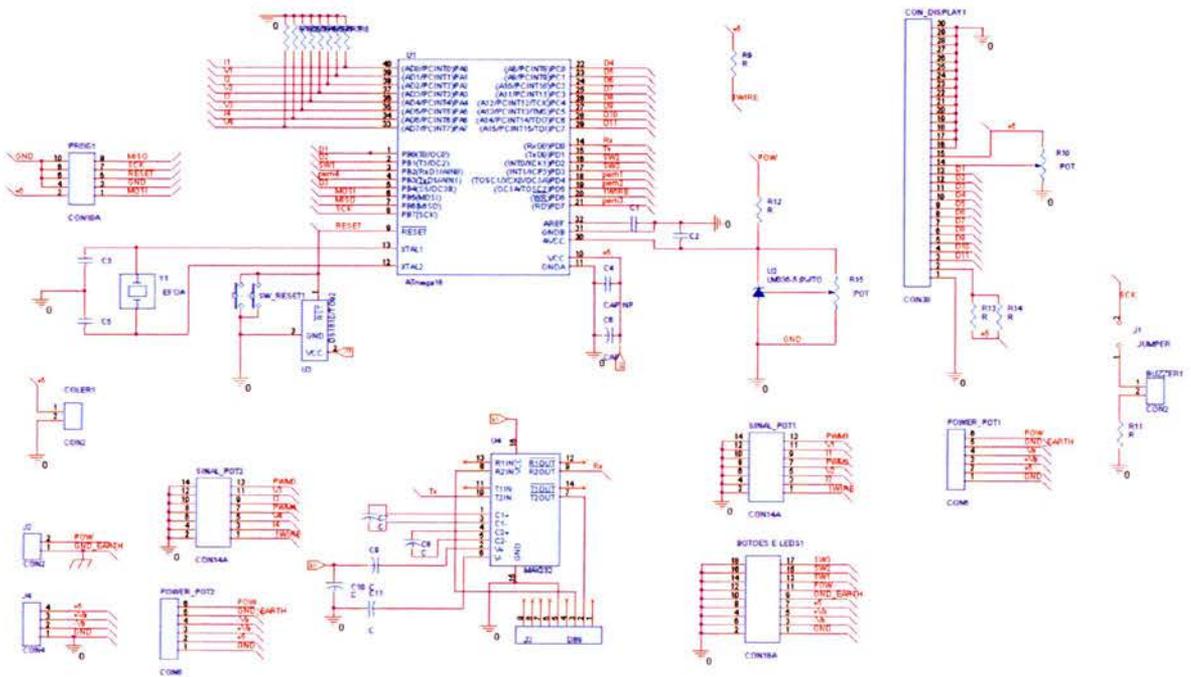


Figura 66 - Esquemático da interface local

8.3.1.2 Dimensionamento dos componentes

As resistências de *pull-down* foram dimensionadas tendo em conta a corrente de saída máxima admissível dos pinos I/O do microcontrolador.

$$R_{\min} = \frac{V_{cc}}{I_{max}} = \frac{5}{40 \cdot 10^{-3}} = 0.125k\Omega = 125\Omega$$

I_{max} foi determinado consultando no *datasheet* as características eléctricas dos pinos I/O do microcontrolador.

A montagem dos sensores de temperatura foi retirada do *datasheet* do fabricante.



8.3.1.3 Interface e controlo remoto

A interface e controlo remoto, permite ao utilizador uma visualização da evolução temporal, dos diferentes parâmetros de carga da bateria. Na aplicação desenvolvida é também possível controlar o processo de carga.

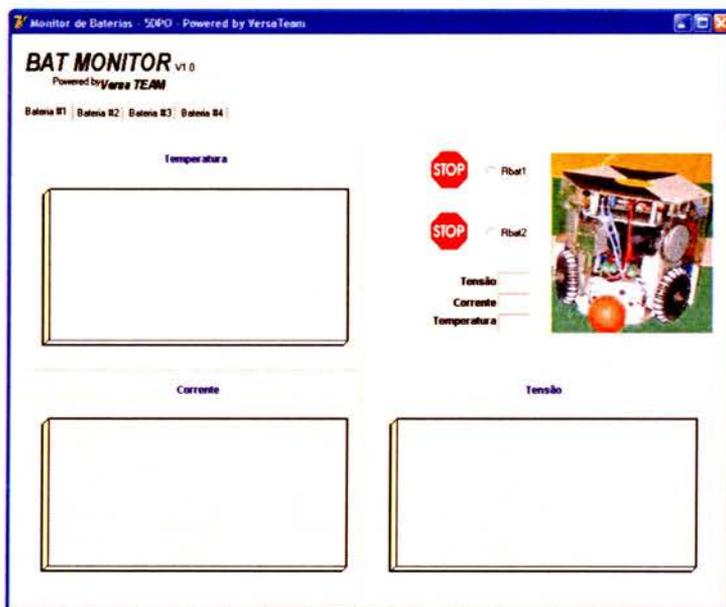


Figura 67 - Interface remota de controlo e monitorização

8.3.1.4 Protocolo de comunicação entre PC-Microcontrolador

Para a transferência de dados entre o microcontrolador e o PC foi implementado um protocolo de comunicação semelhante ao protocolo descrito na alínea 6.8.1 deste documento.

<0B060F [dataOut1] [dataOut2] [dataOut3] [dataOut4]>

Cada trama enviada para o PC contém informação relativa à temperatura, tensão, corrente, estado de carga, modo de carga e valor do PWM de controlo para cada bateria.

[dataOut n] = [temperatura] [tensão] [corrente] [estado] [PWM] [modo]

No caso do utilizador desejar desligar o processo de carga de uma bateria é enviado em broadcast uma mensagem com a seguinte estrutura:

<0B100F B[1] B[2] B[3] B[4]>

B[n] = 0	Desligar o processo de carga da bateria n
B[n] = 1	Activar o processo de carga da bateria n



8.3.3 Controlo do circuito de potência

8.3.3.1 Software de controlo

O controlo do circuito de potência é realizado recorrendo a um microcontrolador.

Através da fusão e processamento dos dados provenientes do barramento 1-wire e do porto ADC é realizado o controlo do circuito de potência através dos sinais de saída dos canais de PWM, que controlam a potência entregue à carga (neste caso a bateria).

A máquina de estados desenvolvida, que implementa o controlo desejado, apresenta a seguinte estrutura:

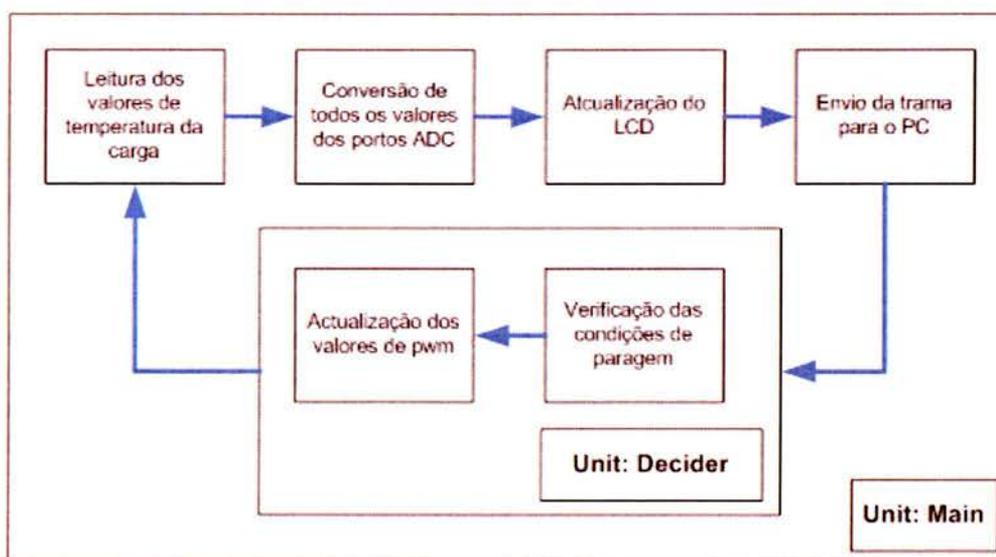


Figura 70 - Esquema da máquina de estados desenvolvida para o processo de carga

8.4 Verificação e validação experimental

Os requisitos inicialmente definidos, foram cumpridos quase na sua totalidade uma vez que até à data de entrega do relatório, apenas faltou implementar o sistema projectado em placas de circuito impresso.

A validação deste sistema foi realizada com testes reais de carga, apresentando os resultados esperados.

Foram ainda simuladas as situações para as quais o processo de carga das baterias teria que ser desligado, mais concretamente, no casos da temperatura exceder os 45°C e no caso da tensão da bateria exceder os 1.8V/célula

No anexo C deste documento estão listados todos os componentes usados na implementação deste projecto.



Capítulo 9

Verificação e validação experimental

Para que o projecto crescesse saudável, desigou-se que o processo de verificação fosse contínuo. Dessa forma, à medida que o projecto ia avançando, ia-se verificando, quase diariamente, se os requisitos inicialmente definidos estavam a ser cumpridos dentro dos prazos estabelecidos no planeamento do projecto.

Essa avaliação foi feita da forma que nos pareceu mais eficaz, ou seja, mediante a escassez de tempo disponível para que o objectivo fosse alcançado, optou-se por verificar de uma forma empírica e imediata. Para isso foi necessária a construção de uma pista à escala de um meio, tornando assim este processo extremamente rápido e eficaz.

O projecto foi validado num ambiente de competição, onde o principal indicador da eficácia do sistema desenvolvido foram os três dias de prova no Festival Nacional de Robótica, que culminaram com a obtenção do segundo lugar final.



Figura 72 - Troféu do 2ºLugar obtido

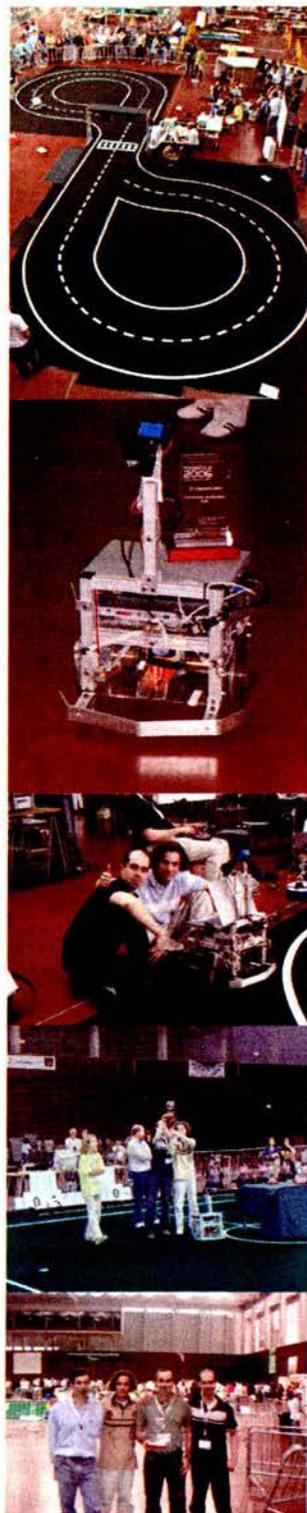


Figura 71 - Fotos do Festival de robótica 2006



Capítulo 10

Conclusões

O objectivo inicial deste projecto era conceber um sistema autónomo versátil capaz de participar em provas de robótica. Mais tarde e durante o seu desenvolvimento surgiu a oportunidade de participar na prova de condução autónoma integrada no Festival Nacional de Robótica, o que tornou o projecto muito mais ambicioso pela limitação de tempo que se disponha para o concretizar.

A maior questão que se levantou na sua execução foi o tipo de arquitectura a implementar, se se optaria por uma arquitectura do tipo *reactiva* ou do tipo *estado do mundo* não seria mais interessante pelo seu carácter tecnológico e inovador. Esta última condição foi escolhida também pela sua originalidade e pela novidade dos assuntos abordados. Esta opção trouxe-nos a sensação que optamos pelo caminho certo, pois permitiu a filtragem de informação consoante a zona do trajecto em que o robot se encontrava a dado momento, permitindo ignorar informações desnecessárias para o bom decorrer da prova. Facto esse que só uma arquitectura de tipo *estado do mundo* permite. As horas a fio dedicadas a este projecto são espelho da motivação e empenho que abraçamos até ao fim.

O sucesso deste projecto foi a reflexão do espírito de equipa que sempre existiu, as discussões ajudaram a uma reflexão e desenvolvimento mais bem estruturado, a amizade ajudou na perseverança.

Mesmo estando satisfeitos com os resultados obtidos em tão curto espaço de tempo, entendemos que existem vários aspectos a melhorar, pois o tempo de processamento de imagem pode ser optimizado. É possível também, no nosso entender melhorar o algoritmo para a afinação do ângulo do robot quando este está em curva.

Durante a competição deparamo-nos com erros na hometria que variavam com a velocidade. Atribuímos a origem a ruídos electromagnéticos causados pelos motores. Uma forma de solucionar o problema será a utilização de uma malha ligada à massa nos cabos dos *encoders*. Verificou-se também que houve alguns problemas de binário, o que pode ser facilmente resolvido com a substituição da caixa redutora ou com a diminuição do diâmetro das rodas.

A aplicação de um sistema deste tipo está condicionada a ambientes semiestruturados. Uma das aplicações possíveis seria implementar um sistema de transporte autónomo num meio fabril onde a planta é conhecida ou num trajecto definido e condicionado a normas. Em qualquer um dos casos a implementação de um sistema deste tipo está dependente do avanço das tecnologias responsáveis pela capacidade de fusão e processamento de dados dos sistemas de controlo.



Capítulo 11

Referências bibliográficas

- [1] Armando Jorge Miranda de Sousa, "Arquitecturas de Sistemas Robóticos e Localização em Tempo Real Através de Visão Aplicações no Domínio do Futebol Robótico", Tese de Doutoramento, Faculdade de Eng^a Univ. Porto (2003)
- [2] Katsuhiko Ogata, "Modern Control Engineering", Published by Prentice Hall (1997) 0 130 60907 2
- [3] Mark S. Nixon, Alberto S. Aguado, "Feature Extraction and Image Processing", Published by Newnes (2002) 0 7506 5078 8
- [4] Rafael C. Gonzales, Richard E. Woods, "Digital Image Processing", Published by Prentice Hall (2002) 0 20 118075 8
- [5] "Charge Methods for Nickel Metal Hydride Batteries"
- http://www.panasonic.com/industrial/battery/oem/images/pdf/Panasonic:NiCd_Charge_Methods.pdf
- [6] <http://www.mecatronicaatual.com.br/artigos/cnc/>
- [7] <http://www.1394imaging.com/en/resources/whitepapers/download/fwcamspecw.pdf>
- [8] <http://www.1394imaging.com/>



ANEXO A

Documentos de apoio ao capítulo 3
(estrutura mecânica)



	<table border="1" data-bbox="827 884 1241 1026"> <tr> <td colspan="3">Projecto: Versa Robot</td> </tr> <tr> <td colspan="2">Autores:</td> <td>André Carvalhosa Tiago Leite</td> </tr> <tr> <td>Data:</td> <td>Versão:</td> <td>Qtd:</td> </tr> <tr> <td>1/07/06</td> <td>1.0</td> <td>1</td> </tr> </table>	Projecto: Versa Robot			Autores:		André Carvalhosa Tiago Leite	Data:	Versão:	Qtd:	1/07/06	1.0	1
Projecto: Versa Robot													
Autores:		André Carvalhosa Tiago Leite											
Data:	Versão:	Qtd:											
1/07/06	1.0	1											

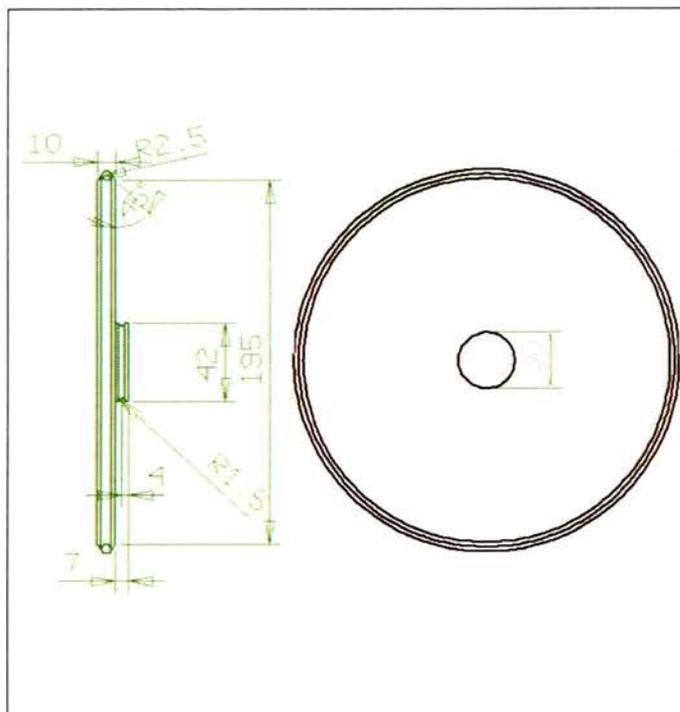
	<table border="1" data-bbox="822 1638 1237 1777"> <tr> <td colspan="3">Projecto: Versa Robot</td> </tr> <tr> <td colspan="2">Autores:</td> <td>André Carvalhosa Tiago Leite</td> </tr> <tr> <td>Data:</td> <td>Versão:</td> <td>Qtd:</td> </tr> <tr> <td>1/07/06</td> <td>1.0</td> <td>1</td> </tr> </table>	Projecto: Versa Robot			Autores:		André Carvalhosa Tiago Leite	Data:	Versão:	Qtd:	1/07/06	1.0	1
Projecto: Versa Robot													
Autores:		André Carvalhosa Tiago Leite											
Data:	Versão:	Qtd:											
1/07/06	1.0	1											



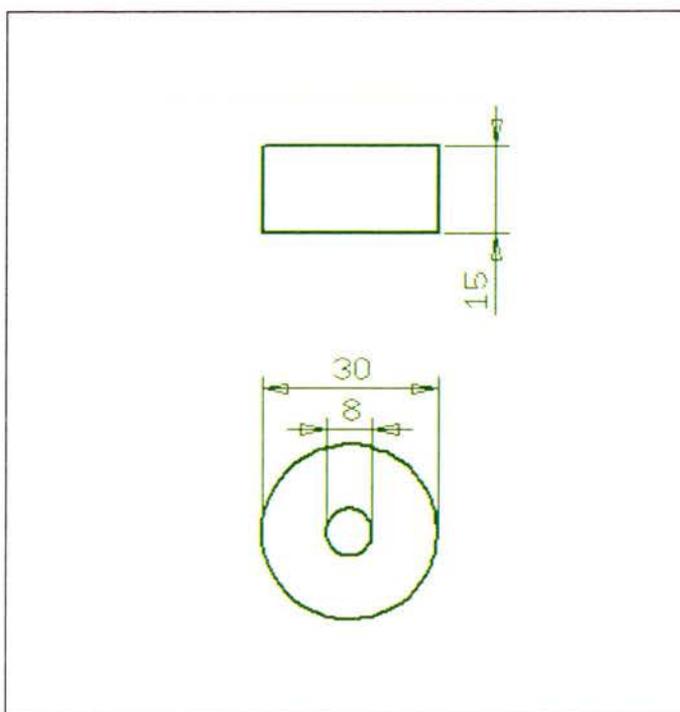
Projecto: Versa Robot		
Autores:		André Carvalhosa Tiago Leite
Data:	Versão:	Qtd:
1/07/06	1.0	1

Projecto: Versa Robot		
Autores:		André Carvalhosa Tiago Leite
Data:	Versão:	Qtd:
1/07/06	1.0	1



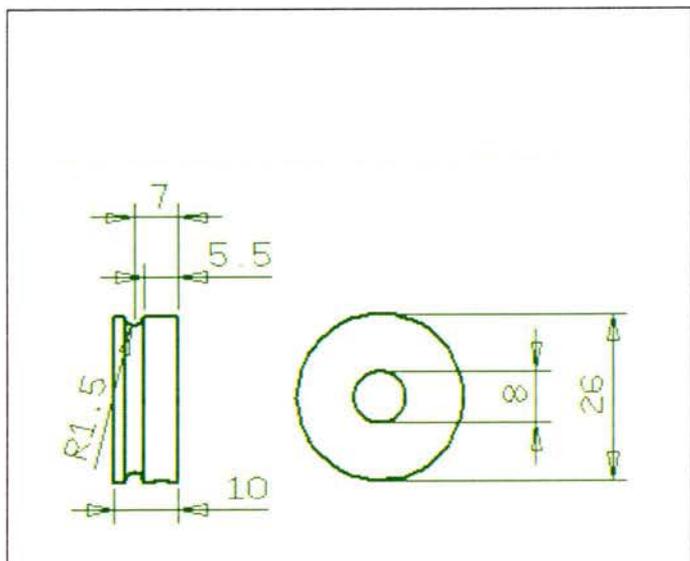


Projecto: Versa Robot		
Autores:	André Carvalhosa Tiago Leite	
Data:	Versão:	Qtd:
1/07/06	1.0	2

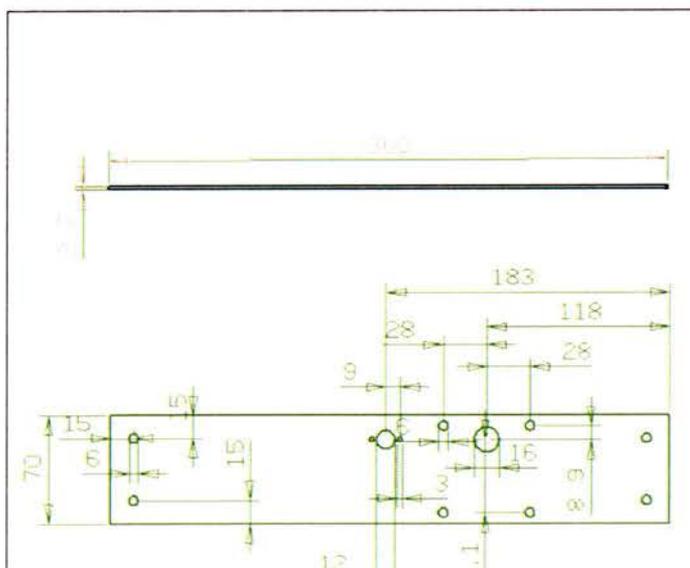


Projecto: Versa Robot		
Autores:	André Carvalhosa Tiago Leite	
Data:	Versão:	Qtd:
1/07/06	1.0	2



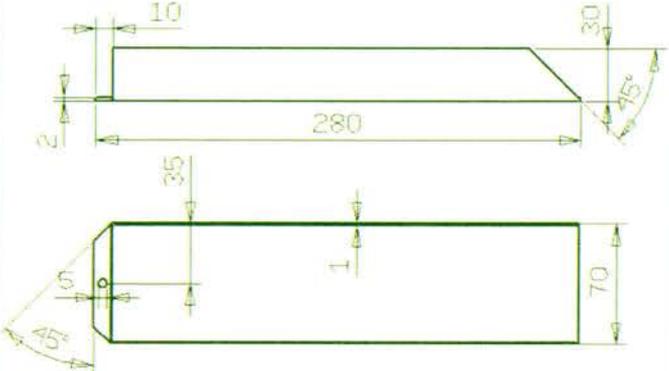
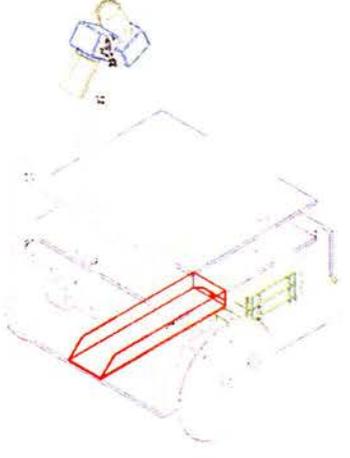


Projecto: Versa Robot		
Autores:	André Carvalhosa Tiago Leite	
Data:	Versão:	Qtd:
1/07/06	1.0	2

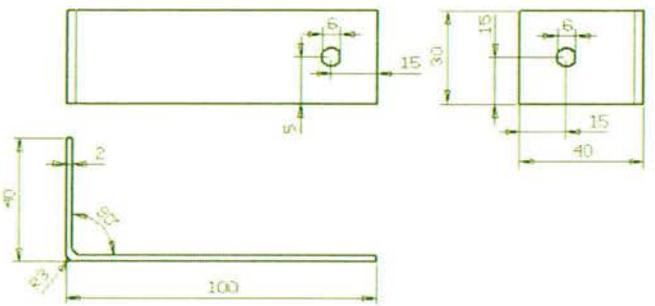
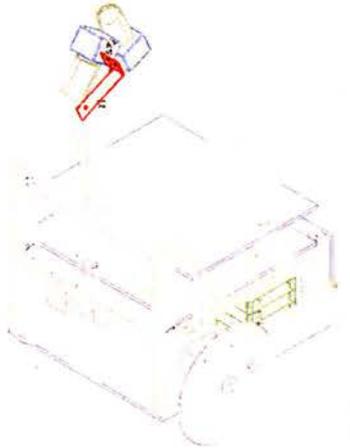


Projecto: Versa Robot		
Autores:	André Carvalhosa Tiago Leite	
Data:	Versão:	Qtd:
1/07/06	1.0	2



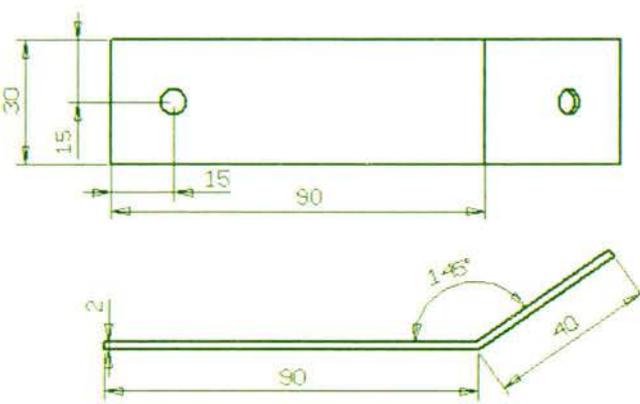
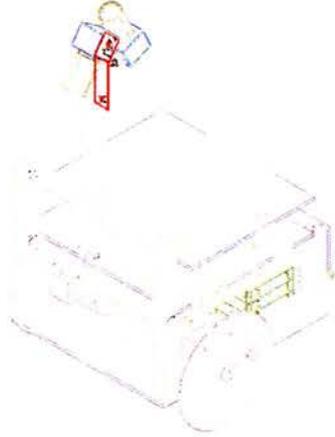



Projecto: Versa Robot		
Autores:	André Carvalhosa Tiago Leite	
Data:	Versão:	Qtd:
1/07/06	1.0	1

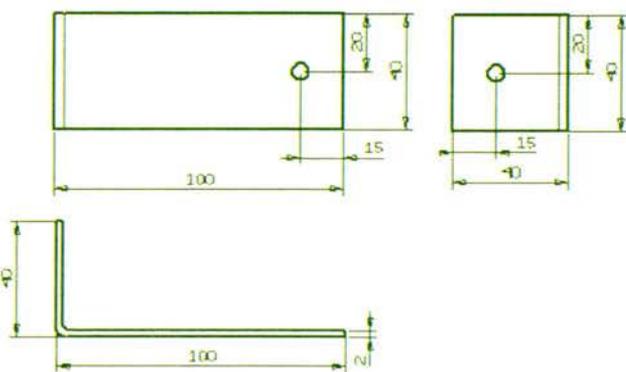
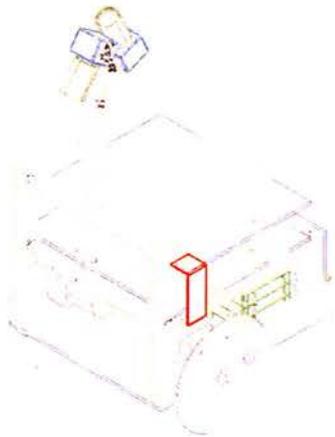



Projecto: Versa Robot		
Autores:	André Carvalhosa Tiago Leite	
Data:	Versão:	Qtd:
1/07/06	1.0	1



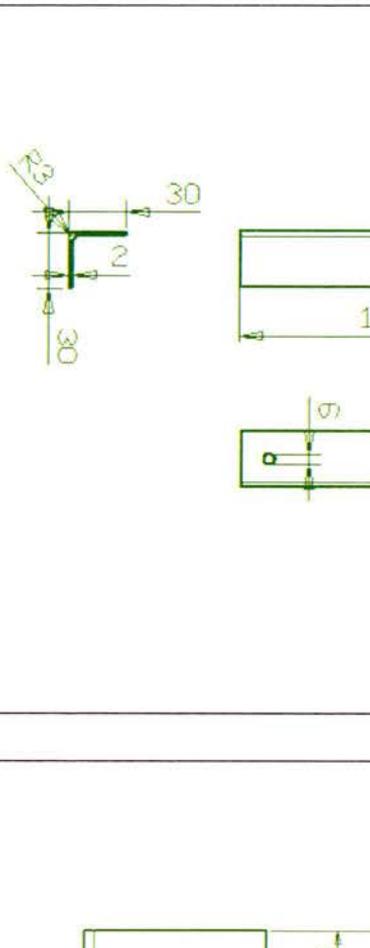



Projecto: Versa Robot		
Autores:		André Carvalhosa Tiago Leite
Data:	Versão:	Qtd:
1/07/06	1.0	1

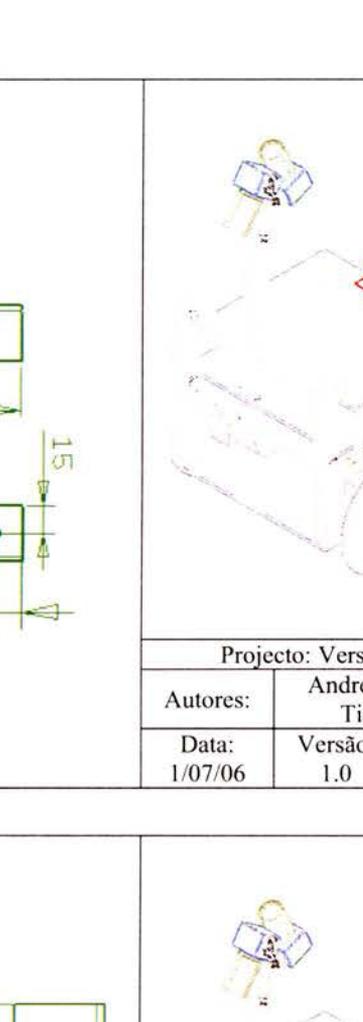



Projecto: Versa Robot		
Autores:		André Carvalhosa Tiago Leite
Data:	Versão:	Qtd:
1/07/06	1.0	4





Projecto: Versa Robot		
Autores:		André Carvalhosa Tiago Leite
Data:	Versão:	Qtd:
1/07/06	1.0	2



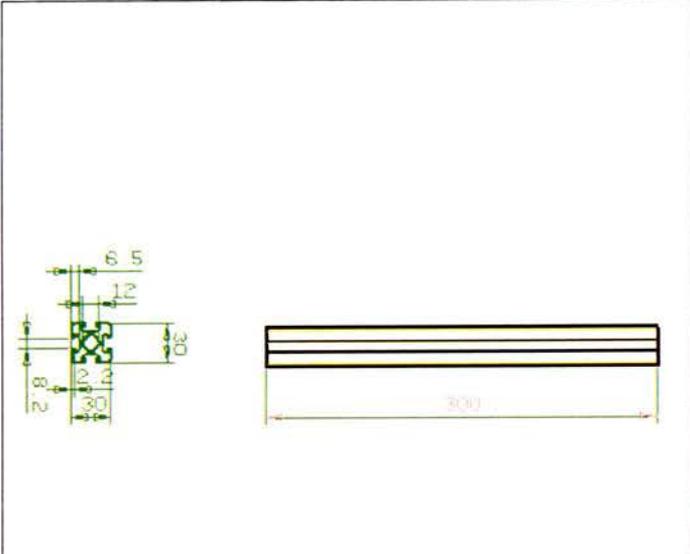
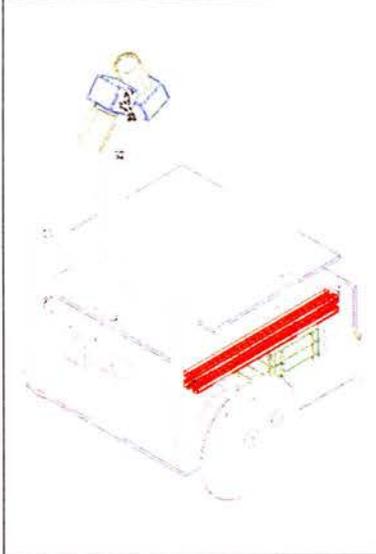
Projecto: Versa Robot		
Autores:		André Carvalhosa Tiago Leite
Data:	Versão:	Qtd:
1/07/06	1.0	2

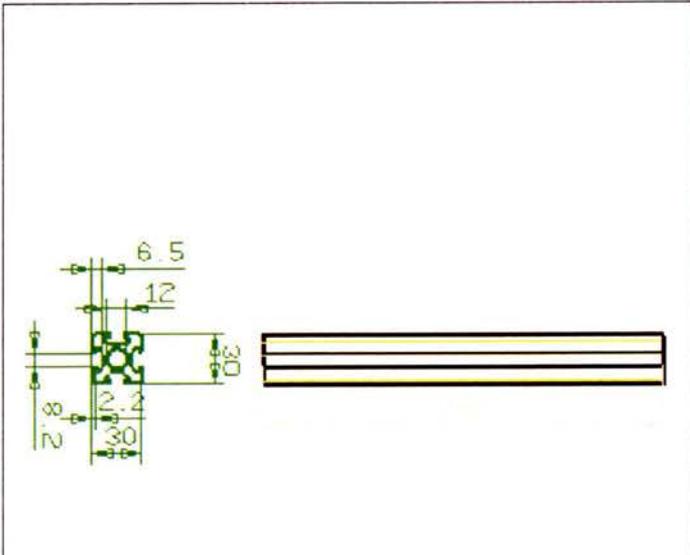
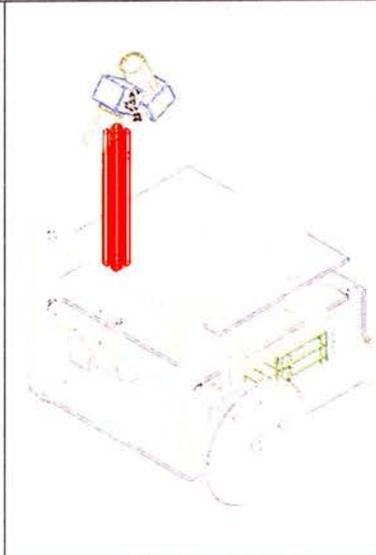


Projecto: Versa Robot		
Autores:	André Carvalhosa Tiago Leite	
Data:	Versão:	Qtd:
1/07/06	1.0	1

Projecto: Versa Robot		
Autores:	André Carvalhosa Tiago Leite	
Data:	Versão:	Qtd:
1/07/06	1.0	2



		
Projecto: Versa Robot		
Autores:	André Carvalhosa Tiago Leite	
Data: 1/07/06	Versão: 1.0	Qtd: 2

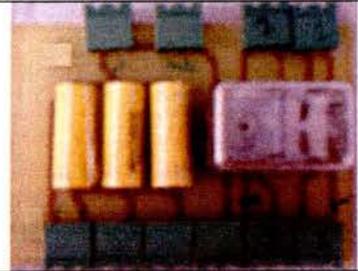
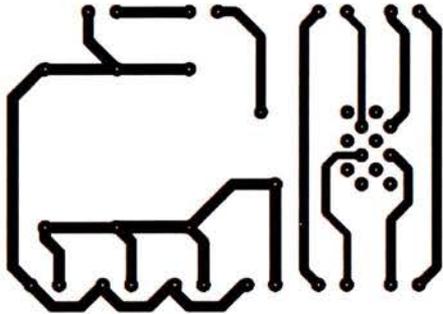
		
Projecto: Versa Robot		
Autores:	André Carvalhosa Tiago Leite	
Data: 1/07/06	Versão: 1.0	Qtd: 1



ANEXO B

Documentos de apoio ao capítulo 4
(estrutura eléctrica/electrónica)



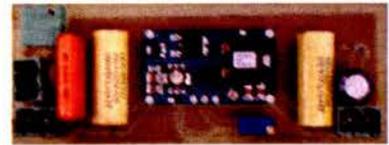
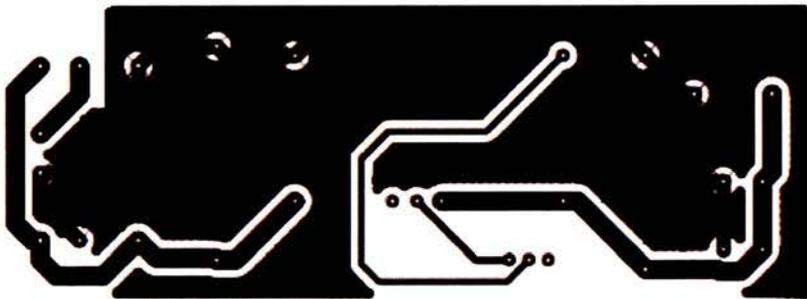


Projecto: Versa Robot

Circuito de corte de alimentação dos motores

Autores: André Carvalhosa
Tiago Leite

Data: 1/07/06 Versão: 1.0 Qtd: 1

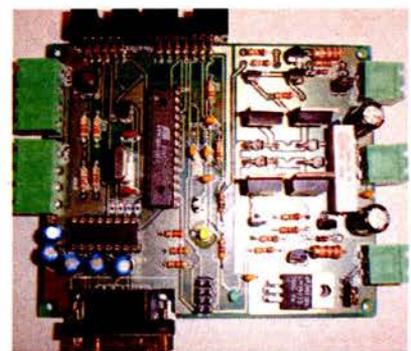
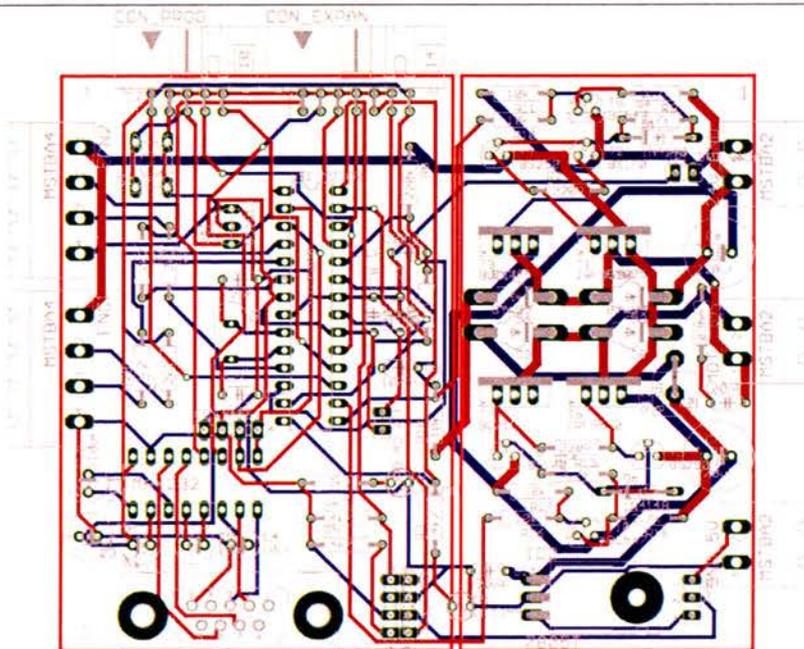


Projecto: Versa Robot

Conversor DC-DC

Autores: André Carvalhosa
Tiago Leite

Data: 1/07/06 Versão: 1.0 Qtd: 1



Projecto: Versa Robot

Drivers dos motores

Autores: André Carvalhosa
Tiago Leite

Data: 1/07/06 Versão: 1.0 Qtd: 3



ANEXO C

Documentos de apoio ao capítulo 8
(Projecto de um sistema de carga rápida de baterias)



Lista de componentes Utilizados

Referência do componente	Descrição	Qtd
DS18S20	Sensor Temperatura	4
ATMEGA 16	Microcontrolador	1
LCD (PC1602LRS-FSO-B)	Display LCD	1
MAX232		1
BS170	FET	4
IRF9540	Power MOSFET P	4
RJ45 Macho PCB		2
1447385	Bobina	4
TL074	AmpOp	4
LM336	IC, REFERENCE 5.0V	5
MBR340RL	Díodos Schottky	8
FUSE (RUEF250)	Resettable Fuse	4
Resistências – 10K		26
Resistências – 1k		4
Resistências – 4.7K		1
Condensadores – 10uf	Cap.	8
Condensadores – 22pf	Cap. NON-POL.	5
Condensadores – 100nF	Cap. NON-POL.	3
Condensadores – 2,2uf		4
Ficha DB9 - Fêmea	Conector	1
CON2 x 2	Conector	4
CON4 x 2	Conector	2
CON16 DISPLAY	Conector	1
CON18 x 2	Conector	8
CON5 x 2	Conector	2
Switch	Interruptor	3
LED RED	DiodoLed	5
LED Green	DiodoLed	5
LED Yellow	Diodo Led	5



ANEXO D

Acrónimos e Siglas

CAD - Desenho assistido por computador
CAM - Manufactura assistida por computador
CAN - *Controller area network*
COTS - *Components-Off-The-Shelf*
DC - Corrente contínua
F.E.M - Força electromotriz
FEUP - Faculdade de Engenharia da Universidade do Porto
I2C - *Two wire interface*
ISBN - *International Standard Book Number*
MCU - Microcontrolador
PC - Computador pessoal
PCB - Placa de circuito impresso
PWM - Modelação de largura de impulso
RT - Tempo real
USB - *Universal serial Bus*
V- Velocidade tangencial do robot
W- Velocidade angular do robot

5DPO – Núcleo de sistemas e robótica da FEUP



ANEXO E

Ficha técnica

Escrito em Microsoft Word versão 11.0 sob Windows XP
Fontes: Times New Roman

Páginas:	100
Parágrafos:	1259
Linhas:	3063
Palavras:	14966
Caracteres:	78754
Caracteres (incl. espaç...):	92726
Bytes:	3019264

URL da página dos Autores: <http://www.fe.up.pt/~ee00011>
<http://www.fe.up.pt/~ee99206>

URL da página da equipa de Futebol Robótico da FEUP: <http://www.fe.up.pt/~robosoc>

André Manuel Ferraz Carvalhosa
Tiago Leão Branco Leite

1 Julho de 2006





FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

BIBLIOTECA



0000104941