

Faculdade de Engenharia da Universidade do Porto



**FEUPCar: Condução Autónoma no Festival
Nacional de Robótica**

Pedro Miguel Amorim Gomes

VERSÃO FINAL

Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores
Major Automação

Orientador: Prof. Dr. António Paulo Gomes Mendes Moreira
Co-orientador: Prof. Dr. Luís Paulo Gonçalves dos Reis


Junho de 2010

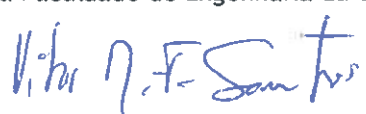
A Dissertação intitulada

“FEUPCAR: CONDUÇÃO AUTÓNOMA NO FESTIVAL NACIONAL DE ROBÓTICA”

foi aprovada em provas realizadas em 23/Julho/2010

o júri


Presidente Professor Doutor Paulo José Cerqueira Gomes da Costa
Professor Auxiliar do Departamento de Engenharia Electrotécnica e de
Computadores da Faculdade de Engenharia da Universidade do Porto


Professor Doutor Vitor Manuel Ferreira dos Santos
Professor Associado do Departamento de Engenharia Mecânica da
Universidade de Aveiro


Professor Doutor Luís Paulo Gonçalves dos Reis
Professor Auxiliar do Departamento de Engenharia Informática da
Faculdade de Engenharia da Universidade do Porto (Orientador)

O autor declara que a presente dissertação (ou relatório de projecto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extractos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são correctamente citados.


Autor - Pedro Miguel Amorim Gomes

Faculdade de Engenharia da Universidade do Porto

Resumo

A constante evolução da tecnologia leva a que hoje em dia tudo se torne possível.

Um dos avanços mais desejados é a automatização dos veículos, que para além de contribuir para maior conforto e fascínio, permitirá reduzir consumos energéticos e acidentes bem como aproveitar melhor as redes de estradas.

Por essa razão surgiram inúmeros eventos relacionados com esse campo, destacando-se assim o Festival Nacional de Robótica no que concerne ao nosso país.

Na presente dissertação sugere-se a construção de um veículo que possa vir a participar na prova de Condução Autónoma do festival mencionado, tendo como objectivo a utilização de sensores de baixo custo.

Assim efectuou-se primeiramente a análise de requisitos bem como das tecnologias de sensorização de baixo custo que poderão ser utilizadas na detecção de obstáculos, túneis, zonas de obras ou até leitura de sinais de trânsito, que poderão aparecer numa via de trânsito real.

Posteriormente desenvolveu-se todo o *hardware* que capacita o robô para a prova. Foram implementados controladores para os motores, instalado um *encoder* que visa permitir a estimação da velocidade, e a instalação e calibração de todos os sensores úteis.

Outro dos objectivos do trabalho, que se expõe, é o desenvolvimento de um software que permita criar uma interface de controlo e decisão da movimentação do robô.

No que respeita à odometria, foi estudado o modelo cinemático do respectivo veículo tendo-se recorrido à calibração da mesma com o objectivo de se melhorar a precisão deste sistema.

Por último foi modelado o robô em estudo através da plataforma *SimTwo*. Com esta modelação pretende-se verificar o comportamento do robô e possibilitar o teste de diferentes estratégias de controlo e decisão bem como aperfeiçoar o *software* desenvolvido.

Abstract

The constant evolution of technology makes everything possible nowadays.

The vehicle automation is one of the most sought after advances, which beyond contribution to a bigger comfort and allure, will make possible the decrease of energy consumption accidents as well as a better use of the road networks.

For this reason, surfaced a number of events related to this field, standing out the National Robotic Festival in regard to our country.

On this thesis, is suggested the construction of a vehicle that may be able to participate on a driverless competition on the mentioned festival, having as a goal the low cost sensors use.

Thus it was performed primarily the analyses of requirements, as well as the low budget sensing technologies that may be used in the obstacle detection, tunnels, construction zones or even traffic signs reading, that can show up in a real street.

Subsequent to this it was developed all the robot capacitating hardware for the trial. The motor drivers were implemented, an encoder installed intended to allow the speed estimation, and the installment and calibration of all the useful sensors.

Another of this thesis's goal is the development of a software which permits the creation of an interface of a control and decision of the robot motion.

On what concerns the odometry, the vehicles kinematic model was studied turning to the calibration of it, with the target to improve this system accuracy.

Finally, the studied robot was modeled thru a SimTwo platform. With this modeling it is intended to verify the robot's behavior and allow the experiment of different control and decision strategies, as well as refine the developed software.

Agradecimentos

Com a consciência de que a realização deste trabalho não teria sido possível sem a contribuição e apoio de muitas pessoas e entidades, reservo este espaço para demonstrar a minha gratidão a todas elas.

Em primeira instância quero agradecer à Faculdade de Engenharia da Universidade do Porto que me proporcionou esta oportunidade e que me ofereceu excelentes condições para a elaboração deste projecto.

Quero aqui expressar o meu agradecimento ao Prof. António Paulo Moreira, meu orientador, e ao, meu co-orientador Prof. Luís Paulo Reis por todo o tempo e apoio disponibilizado ao longo da realização deste projecto, e pela disponibilidade que sempre manifestaram.

Gostaria também de agradecer a todos os colaboradores do grupo ROBIS, especialmente ao Héber Sobreira, ao Hugo Alves e ao Miguel Pinto, que se disponibilizaram sempre para me ajudar nos mais diversos problemas que foram surgindo ao longo do projecto.

Aos meus pais que me apoiaram ao longo de todo o meu percurso académico.

E finalmente, à minha irmã, namorada e amigos, pelo apoio, encorajamento e tolerância à indisposição e indisponibilidade que demonstrei em alguns momentos.

Índice

Resumo	iii
Abstract.....	v
Agradecimentos	vii
Índice.....	ix
Lista de figuras	xi
Lista de tabelas	xiii
Abreviaturas e Símbolos	xv
Capítulo 1	1
Introdução	1
1.1. Enquadramento Geral	1
1.2. Objectivos.....	3
1.3. Organização do Documento	3
Capítulo 2	5
Estado da Arte	5
2.1. História da Condução Autónoma	5
2.2. Condução Autónoma no Festival Nacional de Robótica.....	7
2.1.1. Análise de necessidades do veículo	10
2.1.1.1. Percurso do veículo dentro da respectiva faixa de rodagem.....	10
2.1.1.2. Reconhecimento da passadeira.....	14
2.1.1.3. Circulação dentro do túnel.....	15
2.1.1.4. Detecção dos obstáculos	15
2.1.1.5. Navegação e manobras na zona de Parque de Estacionamento.....	16
2.1.1.6. Detecção e compreensão do painel de sinalização	17
2.1.1.7. Sinais de Trânsito.....	20
2.1.1.8. Identificação da zona de obras	21
2.3. Tecnologias para Condução Autónoma.....	21
2.1.2. Sensores.....	22
2.1.3. Navegação	23
2.1.4. Planeamento	23
2.1.5. Actuação	23
Capítulo 3	25
Concepção Estrutural	25

3.1. Descrição Geral	25
3.2. Hardware	26
3.2.1. Motor de Tracção.....	26
3.2.2. Motor Servo - Direcção	28
3.2.3. Encoder	29
3.2.4. Sistema de Alimentação.....	30
3.2.5. Sensores de Detecção de Linhas Brancas	30
3.2.6. Sensores de Obstáculos e de Paredes	32
3.2.7. Módulo Generic ATmega8	33
3.2.8. Esquema geral do <i>Hardware</i>	33
3.3. Software	34
3.3.1. Camada de Nível Superior.....	35
3.3.1.1. Modelização dos Sensores de Obstáculo e de Paredes.....	36
3.3.1.2. Interface Desenvolvida	38
3.3.2. Camada de Nível Inferior.....	38
3.3.2.1. Controlo do Motor de Tracção.....	38
3.3.2.2. Controlo do motor de Direcção	43
3.4. Protocolo Comunicação.....	44
3.4.1. Detalhes do Protocolo	44
3.4.2. Composição da Trama	46
3.4.2.1. Composição da Trama no Projecto	46
Capítulo 4	47
Localização e Navegação.....	47
4.1. Descrição Geral	47
4.2. Hodometria - Localização Relativa.....	47
4.2.1. Modelo Cinemático - <i>Ackerman</i>	48
4.2.2. Calibração da hodometria.....	49
Capítulo 5	51
Plataforma de Simulação.....	51
5.1. Descrição Geral do <i>SimTwo</i>	51
5.1.1. Apresentação do Ambiente de desenvolvimento <i>SimTwo</i>	52
5.2. Modelação do <i>FEUPCar</i>	54
Capítulo 6	57
Conclusões e Trabalhos Futuros.....	57
6.1. Conclusões.....	57
6.2. Trabalhos Futuros	58
Anexo A.....	61
Ensaio Realizados para Modelização dos Sensores de IV de Distância.....	61
Anexo B	65
Filtro de Mediana	65
Referências	67

Lista de figuras

Figura 1.1 - Exemplo de um Veículo Autónomo.	3
Figura 2.1 - Pista da prova de Condução Autónoma do Festival Nacional de Robótica [3].	8
Figura 2.2 - Traçado da pista de Condução Autónoma [3].	10
Figura 2.3 - Zona da passadeira (inclui também a sinalização e zona de estacionamento) [3].	14
Figura 2.4 - Túnel da pista de Condução Autónoma no FNR [3].	15
Figura 2.5 - Obstáculos (representados a verde) na pista de Condução Autónoma do FNR [3].	15
Figura 2.6 - Zona de Parque de Estacionamento da pista de Condução Autónoma (representado por P e com a presença de um obstáculo num dos lugares) [3].	16
Figura 2.7 - Painel de Sinalização da pista de Condução Autónoma [3].	17
Figura 2.8 - Sinalização do painel (respectivamente STOP, seguir em frente, fim de prova, seguir pela esquerda e seguir para zona de estacionamento).	17
Figura 2.9 - Os seis sinais de trânsito escolhidos para a edição de 2010 do FNR [3].	20
Figura 2.10 - Zona de Obras da pista de Condução Autónoma [3].	21
Figura 3.1 - Strike Short Course Truck RTR, Veículo utilizado no trabalho.	25
Figura 3.2 - Aspecto final do robô <i>FEUPCar</i>	26
Figura 3.3 - Esquema de potência do controlador do motor.	27
Figura 3.4 - Sinal de PWM à entrada da drive do <i>MOSFET</i> - <i>DutyCycle</i> de 30%.	27
Figura 3.5 - Sinal aplicado ao motor - <i>DutyCycle</i> de 30%.	28
Figura 3.6 - Driver construída para o motor de tracção.	28
Figura 3.7 - Representação do PWM de controlo do motor servo para as posições limite.	29
Figura 3.8 - Transmissão (à esquerda) e <i>encoder da IDEC</i> colocado no robô (à direita).	30
Figura 3.9 - Sistema de alimentação do robô.	30

Figura 3.10 - Esquema eléctrico dos sensores de detecção de linhas brancas.	31
Figura 3.11 - Sensor de detecção de linhas	31
Figura 3.12 - Esquema de posicionamento dos sensores.....	32
Figura 3.13 - Sensor <i>Sharp GP2D12</i>	33
Figura 3.14 - Módulo <i>Generic ATmega8</i>	33
Figura 3.15 - Diagrama de blocos do <i>Hardware</i> desenvolvido.	34
Figura 3.16 - Diagrama geral do funcionamento da camada de nível superior.	36
Figura 3.17 - Gráfico representativo da relação tensão/distância dos sensores de distâncias.	37
Figura 3.18 - Interface desenvolvida sob a plataforma <i>Lazarus</i>	38
Figura 3.19 - Diagrama de blocos do controlador de velocidade do motor de tracção.	39
Figura 3.20 - Evolução temporal da velocidade do robô para uma tensão de 3.8 V.	40
Figura 3.21 - Resultados da aproximação a um sistema de 1ª Ordem.	41
Figura 3.22 - Diagrama de blocos do sistema controlado.....	41
Figura 3.23 - Diagrama de blocos do controlador simulado em <i>MATLAB</i>	42
Figura 3.24 - Resposta do controlador dimensionado a um degrau.	42
Figura 3.25 - Resultados do controlador de velocidade do motor.....	43
Figura 3.26 - Esquema do protocolo de comunicação entre PC e controlador (<i>Generic ATmega8</i>).	44
Figura 3.27 - Diagrama de blocos do funcionamento do protocolo entre o <i>Master</i> e o <i>Slave</i>	45
Figura 4.1 - Esquema cinemático do robô.	48
Figura 4.2 - Movimento em linha recta do robô.....	50
Figura 4.3 - Movimentos do robô para vários ângulos de direcção.	50
Figura 5.1 - Janela de Ambiente 3D do <i>SimTwo</i>	52
Figura 5.2 - Janela de edição dos robôs em <i>SimTwo</i>	53
Figura 5.3 - Janela de configuração (à esquerda) e de programação em Pascal (à direita) existentes no <i>SimTwo</i>	53
Figura 5.4 - Janela de gráficos de evolução do <i>SimTwo</i>	54
Figura 5.5 - Imagem do veículo original do simulador.	55
Figura 5.6 - Aspecto final da plataforma <i>FEUPCar</i> simulada.	56

Lista de tabelas

Tabela 2.1 - Tabela sumária das penalizações em que podem incorrer os robôs [3].....	9
Tabela 2.2 - Tabela sumária de bónus [3].	9
Tabela 2.3 - Comparação de métodos utilizados para detecção do traçado da pista.....	10
Tabela 2.4 - Comparação de métodos utilizados para detecção da sinalização referente ao painel.	18
Tabela 3.1 - Constituição da trama	46
Tabela 4.1 - Parâmetros, após calibração, úteis no cálculo odométrico, do <i>FEUPCar</i>	50

Abreviaturas e Símbolos

Lista de abreviaturas

3D	Três Dimensões
CMU	<i>Carnegie Mellon University</i>
DARPA	<i>Defense Advanced Research Projects Agency</i>
ELROB	<i>The European Robot Trial</i>
FEUP	Faculdade de Engenharia da Universidade do Porto
FNR	Festival Nacional de Robótica
GPS	<i>Global Position System</i>
IV	Infra-Vermelhos
MATLAB	<i>Matrix Laboratory</i>
MOSFET	<i>Metal Oxide Semiconductor Field Effect Transistor</i>
PC	<i>Personal Computer</i>
PI	<i>Proportional-Integral</i>
PWM	<i>Pulse-Width Modulation</i>
UMB	<i>Universitat der Bundeswehr, Munchen</i>
USB	<i>Universal Serial Bus</i>

Lista de símbolos

ϕ	Ângulo relativo a uma roda imaginária situada entre as rodas dianteiras do veículo
θ	Ângulo que o veículo faz com o referencial de referência.
A	<i>Ampère</i>
cm	Centímetros

ms	Milissegundos
s	Segundos
V	<i>Volts</i>
D_{Volts}	Distância em <i>Volts</i> dos sensores <i>SHARP</i>
τ	Constante de tempo
Δd	Distância percorrida pelo veículo
L	Distância entre os eixos dianteiro e traseiro
D	Diâmetro das rodas
ϕ_E	Ângulo efectuado pela roda dianteira esquerda do robô
ϕ_D	Ângulo efectuado pela roda dianteira direita do robô
R	Raio delineado pelo movimento do robô
Δd	Distância percorrida num determinado intervalo de tempo
$\Delta\theta$	Variação de ângulo, desenhado com o referencial mundo, num determinado intervalo de tempo
NI_e	Número de impulsos do encoder num determinado intervalo de tempo
$C_{i/m}$	Factor de conversão de impulsos para distância percorrida pelo robô
P	Ponto que caracteriza a posição cartesiana no referencial mundo

Capítulo 1

Introdução

Com a constante evolução das tecnologias, a robótica móvel tem despertado elevado interesse, o que levou ao aparecimento de um sem número de projectos de investigação por todo o mundo. Com aplicações em diversas áreas, tais como indústria automóvel, transportes públicos e industriais, medicina e na procura de vida extra-terrestre, pode trazer-se maior conforto e produtividade, e ao mesmo tempo, menos risco para toda a sociedade.

No que concerne à indústria automóvel, já existem variados sistemas semi-autónomos. É disso exemplo o estacionamento automático referente ao Toyota Prius (ano 2007) - após a escolha do espaço o sistema controla a direcção, estando o controlo dos pedais reservado ao condutor - bem como o controlo activo da velocidade de cruzeiro e o controlo de estabilidade.

Tais sistemas têm vindo a evoluir, sobretudo ao nível da investigação, com o objectivo de dotar o veículo de uma maior inteligência ao ponto de este ser capaz de efectuar uma condução verdadeiramente autónoma em ambientes desconhecidos.

1.1. Enquadramento Geral

A história da condução autónoma começou há já algum tempo e tem sido alvo de estudo ao longo das três últimas décadas.

Em 1977 o primeiro veículo autónomo, feito no Japão pelo Laboratório de Engenharia Mecânica da Universidade de *Tsukuba*, era muito rudimentar e baseava-se em seguir linhas brancas, marcadas nas estradas, bem semelhantes às existentes nas ruas de hoje. Este veículo executava a tarefa a uma velocidade máxima de 30 km numa pista construída para o efeito e recorrendo a hardware especial (a capacidade de processamento dos computadores da altura era muito inferior à dos dias de hoje) [1].

Desde então, o desenvolvimento e a construção de plataformas capazes de navegar autonomamente nunca mais pararam. Por navegação autónoma entenda-se a capacidade de imitar o homem na recolha de dados sensoriais e actuação com base na análise desses mesmos dados.

Os avanços vistos actualmente nesta área têm como base o enorme investimento por parte de variadíssimas organizações, como a Comissão Europeia e o departamento de defesa

dos Estados Unidos e projectos originários de diversos países.

Daí resultaram alguns testes de sucesso em auto-estradas com condições de trânsito normal, efectuando ultrapassagens e atingindo velocidades elevadas (175 km/h) ao longo de grandes distâncias (mais de 1600 km)[1].

No campo da defesa, é de assinalar a criação de protótipos de plataformas autónomas que ajudam na procura de vida e de armadilhas.

Fruto do investimento e investigação efectuados, foram ainda concebidos sistemas auxiliares de condução. Estes sistemas proporcionam ao condutor um alívio de tarefas repetitivas que têm tendência a tornar a condução enfadonha, bem como novas formas de segurança passiva.

Tais avanços despoletaram, por todo o mundo, um maior interesse e inúmeros trabalhos de investigação, e nos últimos tempos surgiram alguns eventos que permitiram competir e evoluir nessa área.

O mais mediático desses eventos é o *DARPA Urban Challenge* [2], patrocinado pelo *DARPA*¹. Consiste em navegar num ambiente simulado de cidade, com tráfego em movimento e cumprir missões militares, evitando os obstáculos e negociando as intersecções ocupadas. Este evento nasceu em 2004.

Em Portugal existe também um evento, denominado de FNR (Festival Nacional de Robótica) [3], em que existe uma prova de condução autónoma. Com origem no ano de 2001, este evento, tem como objectivo a promoção da ciência e tecnologia através de concursos de robôs. Para além da referida prova de condução autónoma (prova realizada desde a fundação do evento), existem também as provas do *RoboCup*² e também provas de demonstração, que variam de ano para ano [5]. O Festival inclui ainda um encontro científico onde investigadores nacionais e estrangeiros da área da Robótica se reúnem para apresentar os mais recentes resultados da sua actividade.

A prova de Condução Autónoma desenrola-se em três fases, sendo em cada uma delas necessário completar duas voltas a uma pista que pretende simular uma estrada convencional. A pista tem a forma aproximada de um 8, é delimitada tanto interior como exteriormente por linhas contínuas brancas, tem duas faixas de rodagem separadas por um traço interrompido, uma passadeira e um par de painéis sinaléticos (um em cada sentido), que funcionam como semáforos. De fase para fase da competição são introduzidos novos desafios com os quais os robôs têm de lidar (obstáculos sobre a faixa de rodagem, um túnel sobre uma das curvas, uma zona de obras e uma área de estacionamento) [3].

¹ *Defense Advanced Research Projects Agency* - órgão do Departamento de Defesa dos Estados Unidos responsável pelo desenvolvimento de novas tecnologias para uso militar. A missão é manter a superioridade das forças armadas e evitar surpresas tecnológicas.

² *RoboCup* - É uma iniciativa de pesquisa educação internacional que visa promover a pesquisa em robótica e inteligência artificial, fornecendo um problema padrão onde variadas tecnologias podem ser integradas. As modalidades de Futebol Robótico - Liga Média, Futebol Robótico - Liga Júnior, Busca e salvamento e Dança júnior fazem parte desta iniciativa [4].

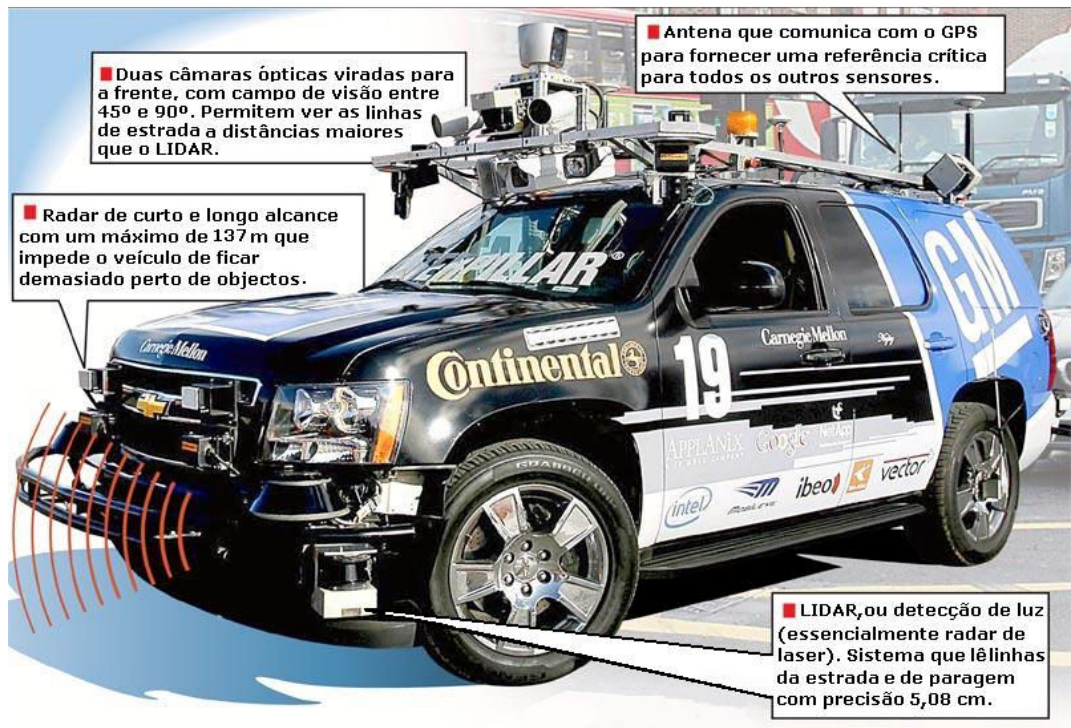


Figura 1.1 - Exemplo de um Veículo Autónomo.

1.2. Objectivos

A presente dissertação tem como objectivo geral construir um robô, a partir do zero, que possa vir a participar no Festival Nacional de Robótica. Baseado num carro telecomandado e sem a utilização de sensores muito sofisticados ou caros, este robô deverá permitir efectuar uma condução autónoma, ou por outra, permitir que o sistema inteligente possa controlá-lo sem necessitar da intervenção humana.

Assim, para a concretização deste objectivo, esta dissertação será distribuída nas seguintes etapas:

- Levantamento de estudos já realizados sobre veículos guiados autonomamente e de tecnologias similares;
- Revisão bibliográfica referente à problemática da localização;
- Projecto e implementação de hardware capaz de controlar o veículo;
- Projecto e implementação de hardware que possibilite a aquisição dos dados sensoriais do mundo;
- Desenvolvimento e implementação de software de controlo da movimentação do veículo;

1.3. Organização do Documento

Esta secção tem como objectivo descrever a estrutura e organização deste documento, explicando sucintamente o que será desenvolvido em cada capítulo.

A introdução ao tema da dissertação é feita neste mesmo capítulo, o capítulo 1 da Introdução. Logo de seguida haverá uma análise do estado da arte quanto aos métodos já existentes e que são mais utilizados na prova de condução autónoma do Festival Nacional de Robótica.

No capítulo 3 serão retratados os elementos de hardware e software desenvolvidos que capacitarão o robô para a condução autónoma, bem como do protocolo de comunicação implementado.

O capítulo responsável pela explicitação do modelo cinemático desenvolvido para a localização do robô é apresentado, seguidamente, no 4º capítulo.

O capítulo 5 apresenta a plataforma de simulação, onde se apresenta a ferramenta de simulação utilizada para testar o comportamento do robô.

Finalmente, no capítulo 6, apresentam-se conclusões gerais retiradas no decorrer do projecto e, tendo em consideração o que foi apreendido, registam-se também algumas propostas futuras.

Capítulo 2

Estado da Arte

Ao falar-se de veículos autónomos, está a falar-se de uma família de tecnologias altamente automatizadas e concebidas para tentar evitar acidentes e lesões, permitir uma melhor exploração da rede de estradas, reduzir o consumo energético, permitir uma maior e melhor mobilidade, ou mesmo reduzir o custo associado ao transporte de bens e mercadorias.

Pensa-se que o desenvolvimento deste tipo de sistemas, aliada à utilização de formas de propulsão amigas do ambiente, seja o maior um dos maiores avanços na tecnologia automóvel.

2.1. História da Condução Autónoma

A história da condução autónoma iniciou-se em 1977 quando o Laboratório de Engenharia Mecânica da Universidade de *Tsukuba*, no Japão construiu o primeiro veículo autónomo. Era ainda muito rudimentar e baseava-se em seguir linhas brancas marcadas nas estradas. Com este veículo/robô conseguiram atingir velocidades de 30km/h [1].

Em 1980, *Ernst Dickmanns*, um dos maiores intervenientes nesta área, e a sua equipa na UMB (*Universitat der Bundeswehr, Munchen*), utilizou uma carrinha Mercedes-Benz, guiada por visão, para percorrer estradas sem trânsito, atingindo os 100 km/h. Após esta experiência a Comissão Europeia iniciou um grande financiamento na área da condução autónoma, mais concretamente no Projecto EUREKA Prometheus (1987-1995) [1].

Em 1995, o mesmo Ernst Dickmanns, reinventou um Mercedes-Benz, classe S, para uma viagem de cerca de 1600Km entre Munich, Alemanha, e Copenhaga, Dinamarca. Para tal, utilizou visão sacádica³ computadorizada e transputers⁴, de forma a obter um controlo em

³ Técnica que simula o olho humano ao procurar pontos de interesse numa imagem e recolhendo informação mais detalhada acerca desses pontos em particular, de forma a construir uma representação da imagem observada e ao mesmo tempo rentabilizar os recursos disponíveis.

tempo real. Este veículo, ao contrário do que se tinha passado até então, executava manobras para ultrapassar outros veículos, e conseguiu 95% de condução autónoma, atingindo um máximo de 175 km/h [1].

No mesmo ano, um projecto na CMU (*Carnegie Mellon University*), obteve 98.5% de condução autónoma, numa viagem intitulada de *No hands across America* (cerca de 4828 Km). Utilizaram o programa de computador RALPH (*Rapidly Adapting Lateral Position Handler*) que usava imagens para determinar a localização na estrada e assim ajustar a direcção [1].

Mais tarde, em 1998, na Universidade de Parma e Pavia, Itália, um projecto denominado de *ARGO vehicle*, foi desenvolvido com a coordenação de Alberto Broggi e financiado pelo governo italiano. Com a ajuda de um *Lancia Therna* modificado e contendo duas câmaras de vídeo de baixo custo a preto e branco e algoritmos de visão estereoscópica⁵, conseguiu-se percorrer 2000 km de auto-estrada, normalmente marcada, com uma condução autónoma efectiva de 94%. Esta viagem ficou conhecida como *MilleMiglia in Automatico* [1].

Em 2001, um projecto do exército dos Estados Unidos, denominado de *Demo III*, demonstrou a capacidade de veículos terrestres autónomos percorrerem distâncias em terrenos irregulares, evitando obstáculos tais como pedras e árvores. O sistema de controlo utilizado foi um sistema hierárquico, designado de *Real-Time Control System* e fornecido por James Albus da NIST (*National Institute of Standards and Technology*). Foi ainda demonstrado que era possível não só controlar veículos individuais, mas também coordenar um conjunto de veículos de forma a atingir determinado objectivo [7].

Ainda em 2001, chega a primeira edição do maior encontro na área da robótica em Portugal, o Festival Nacional de Robótica. Este evento, actualmente uma iniciativa da Sociedade Portuguesa de Robótica, tem como objectivo contribuir para o desenvolvimento da investigação em robótica e automação e promove a ciência e a tecnologia junto dos jovens em vários níveis de ensino, bem como do público em geral, através de competições entre robôs. O Festival decorre todos os anos numa cidade distinta e inclui um encontro científico, onde investigadores nacionais e estrangeiros, da área da robótica, se reúnem para apresentar os mais recentes resultados da sua actividade. Foi neste contexto que surgiu a prova de Condução Autónoma, a grande motivação deste trabalho, com o objectivo de fomentar a investigação e consequente desenvolvimento de técnicas que pudessem ser utilizadas na construção de veículos autónomos.

Em 2004, surge uma das maiores competições a nível mundial, no que à condução autónoma diz respeito, o *DARPA Grand Challenge*. Esta competição é patrocinada pela DARPA e consiste na competição entre equipas originárias de vários países, desde que um dos elementos da equipa seja um cidadão americano, em corridas de veículos autónomos. Para promover ainda mais a competição e fomentar o interesse e empenho na investigação nesta

⁴ Um microprocessador de computação paralela de 1980 da *INMOS*, empresa inglesa.

⁵ Técnica que simula o comportamento do olho humano e usa a paralaxe para, através da fusão de duas imagens bidimensionais, conseguir uma noção de profundidade e a avaliação da distância, posição e tamanho de objectos.

área da robótica, o Congresso dos Estados Unidos, autorizou a DARPA a atribuir um prémio de um milhão de dólares ao vencedor da competição [1].

A primeira edição deste evento, em termos competitivos, não foi a melhor, já que nenhum veículo completou os 240 km de percurso (a Red Team da CMU foi a equipa que percorreu a maior distância, 11,78 km) no deserto de *Mojave* (Califórnia, Estados Unidos), ficando o prémio por reclamar. Apesar disso o evento tinha despertado um grande interesse a nível Mundial [8].

No ano seguinte realizava-se de novo o evento com uma grande quantidade de participantes. Eram noventa e cinco os participantes inicialmente mas após participação na prova *National Qualification Event* foram reduzidos para quarenta e três e posteriormente seleccionados vinte e três. Com 212 km de percurso, neste ano a prova teve melhores resultados, sendo que apenas um dos concorrentes não conseguiu ultrapassar a melhor marca do ano anterior. Houve cinco participantes a concluir o percurso, entre os quais um chegou fora do tempo limite (atingiram-se velocidades médias entre os 28.2 km/h e os 30.7 km/h). O prémio foi arrebatado por um *VW Touareg* da Universidade de *Stanford*, baptizado de Stanley, que percorreu a totalidade do percurso em seis horas e cinquenta e quatro minutos [8].

Em 2006 o departamento de defesa da Alemanha anuncia um evento similar ao *DARPA Grand Challenge*, denominado de *ELROB*⁶. O espírito deste evento não se centra na competição mas sim na demonstração de tecnologia robótica que permita resolver os problemas dos dias de hoje (salvar vidas por exemplo), além de ser uma oportunidade de mostrar o estado de arte da robótica nos dias que correm [17].

Em Novembro de 2007 tem lugar a terceira edição do *DARPA Grand Challenge*, agora apelidada de *Urban Challenge*, teve lugar num recinto fechado (uma base aérea desmantelada) e consistiu num desafio diferente em relação às duas edições anteriores, navegar num ambiente simulado de cidade, com tráfego em movimento e cumprir missões militares, evitando os obstáculos e negociando as intersecções ocupadas [8].

O *Urban Challenge* constituiu, então, num enorme avanço na área da condução autónoma, uma vez que, pela primeira vez, interagiram num meio contíguo veículos autónomos e veículos conduzidos pelo homem. O vencedor do evento foi a equipa *TARTAN Racing* da CMU, através de o seu *Chevy Tahoe*, com o tempo de 4h10m20s [8].

2.2. Condução Autónoma no Festival Nacional de Robótica

Existiram e existem neste momento diversos projectos e parcerias, que têm em vista a investigação e desenvolvimento de sistemas robotizados terrestres com a particularidade de serem completamente autónomos. Além disso existem também alguns eventos que visam promover e impulsionar o desenvolvimento das tecnologias necessárias para aqueles fins.

⁶ *ELROB: The European Robot Trial* - É um evento que possibilitou aos europeus desafios robóticos semelhantes aos *DARPA Grand Challenge*. Permite avaliar a tecnologia actual e é uma oportunidade para a reunião de investigadores e responsáveis industriais.

O Festival Nacional de Robótica é um bom exemplo disso mesmo, e tendo o objectivo de promover desenvolvimentos técnicos e científicos na área da Robótica, tal é conseguido graças à oportunidade que este oferece, nomeadamente com a existência de um encontro científico no qual são apresentados trabalhos na área da robótica móvel, em grande parte trabalhos integrados nas provas das quais o FNR é composto.

De entre todas as provas destaca-se a mais antiga das provas, e a que interessa de sobremaneira abordar, a prova de condução autónoma, tema da presente dissertação.

Esta prova representa um desafio aliciante e de média complexidade, no qual o robô autónomo deverá percorrer um percurso ao longo de uma pista fechada com a forma de um 8, e que apresenta semelhanças com uma estrada convencional de modo a simular uma condução real.

A pista é delimitada por duas linhas brancas, tem duas faixas separadas por uma linha tracejada e apresenta um cruzamento no centro, onde se apresenta uma passadeira e um par de painéis sinaléticos.



Figura 2.1 - Pista da prova de Condução Autónoma do Festival Nacional de Robótica [3].

A competição decorre em três mangas, sendo que em todas elas o objectivo principal é percorrer duas vezes a totalidade da pista no mais curto espaço de tempo e sem penalizações.

O grau de dificuldade vai aumentando em cada uma das três mangas:

- Na primeira é efectuado um contra-relógio;
- Na segunda são acrescentados semáforos que indicam variadas situações (paragem obrigatória, seguir em frente, seguir pela via da esquerda, ou efectuar um estacionamento) e são acrescentados obstáculos na pista;
- Finalmente, na terceira manga é colocado o túnel e uma zona de obras delimitada por um conjunto de cones de obras interligados por uma fita colorida.

No fim das três mangas, a equipa vencedora será aquela que apresentar o menor tempo

total. Este tempo poderá ser influenciado negativamente por penalizações provenientes de erros efectuados pelos robôs e estão relacionados com os seguintes cenários:

Tabela 2.1 - Tabela sumária das penalizações em que podem incorrer os robôs [3].

<i>Infracção</i>	<i>Efeito Relacionado</i>	<i>Penalização (em segundos)</i>
Pequena colisão com acessórios da pista		11
Colisão com acessórios da pista alterando a sua geometria	Fim de Prova	19
Saída parcial de pista		7
Saída total da pista	Fim de Prova	23
Não cumprimento da direcção indicada pelos painéis sinaléticos		25
Não imobilização perante o sinal de STOP		90
Partida prematura perante o sinal de STOP		90
Imobilização fora da zona de paragem		9
Execução prematura da manobra de estacionamento		19
Não execução do estacionamento, tempo de execução superior a 30s ou demora no seu início superior a 2s		23
Manobra de estacionamento irregular (fora da zona de estacionamento ou interferindo com o obstáculo num dos lugares)		7
Início manual dos ensaios		30

Por outro lado, os robôs em prova poderão também beneficiar de algum bónus decorrente da identificação de sinais de trânsito. Tal bónus retira tempo ao já acumulado. Veja-se na Tabela 2.2 os diferentes tipos de bónus:

Tabela 2.2 - Tabela sumária de bónus [3].

<i>Acção</i>	<i>Bónus (segundos)</i>
1 - Identificação de um sinal de trânsito e seu tipo	3
2 - Identificação do número do sinal	2
3 - Respeitar a linha reservada (*)	10
4 - Não pisar a linha contínua central (**)	5
(*) - Não é acumulada com a acção 1 e/ou 2 mas necessita delas para ser obtida	
(**) - Necessita da acção 1, 2 e 3 para ser obtida	

Tendo em conta os vários desafios, torna-se então necessário meditar e analisar os diversos sensores para aquisição correcta dos elementos constituintes do meio integrante da prova.

2.1.1. Análise de necessidades do veículo

As técnicas de seguimento de linhas e de hodometria são as opções mais fáceis e mais utilizadas, mas não são suficientes para uma prova como esta.

Havendo a necessidade de reconhecimento de objectos, como identificação de passadeira, túnel, obstáculos, semáforos e zona de obras que são essenciais à realização da prova, terão que aplicar-se técnicas mais complexas.

Analise-se então as situações que mais carecem de atenção, sob pena de haver insucesso na prova, nomeadamente na obtenção de informações do meio:

2.1.1.1. Percurso do veículo dentro da respectiva faixa de rodagem

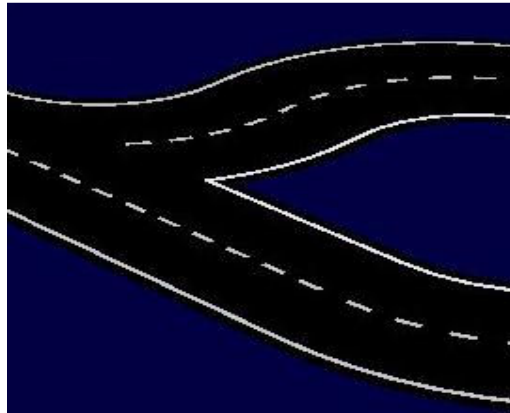


Figura 2.2 - Traçado da pista de Condução Autónoma [3].

Para a detecção do traçado da pista podem utilizar-se várias metodologias a que corresponderão os sensores necessários para o efeito. Veja-se o que já foi implementado para a prova de condução autónoma do FNR.

Tabela 2.3 - Comparação de métodos utilizados para detecção do traçado da pista.

<i>Robô</i>	<i>Metodologia Adoptada</i>	<i>Descrição</i>	<i>Sensores Necessários</i>	<i>Classific ação no FNR</i>	<i>Referênc as</i>
ATLAS III (2005)	Obtenção de um quadrilátero com os extremos da pista dentro da imagem.	Consoante os cantos do quadrilátero encontrados é tomada uma decisão diferente e se não forem encontrados os cantos é tomada a decisão anterior do processo.	Câmara	2º Lugar	[16]
Versa (2006)	Identificação e balizamento das linhas da	Inicialmente é feita a binarização da imagem. Depois é	Câmara e encoder	2º Lugar	[13]

	pista, baseado no centro de massa e orientação do robô.	realizado um threshold dinâmico seguido de um algoritmo de detecção de objectos que consiste em balizar os objectos, filtrar o ruído e cálculo dos momentos centrados da imagem (centro de massa, área, orientação). O parâmetro que distingue as várias faixas de rodagem é a inclinação e o centro de massa do objecto em relação ao robô. É calculada, depois de calibrada a câmara, a posição absoluta do robô através das medições do mesmo às linhas da faixa de rodagem e de seguida é ajustada a posição através de rotações e translações.			
ROTA (2007)	Linearização da pista. Seguimento da linha direita da faixa de rodagem.	Dependendo da distância à linha é ajustada a direcção do veículo. Recorre a um mapa de distâncias de pontos da imagem em relação à frente do carro. A um píxel (px,py) faz-se corresponder um ponto em coordenadas cartesianas (x,y).	Câmara e encoder	2º Lugar	[14]
ROTA	Linearização da pista.	Dependendo da distância à linha é	Câmara e encoder	6º Lugar	[15]

(2008)	Seguimento da linha direita da pista.	ajustada a direcção do veículo. Não é utilizado o mapa de distâncias de 2007 mas sim as distâncias em píxeis, apoiando-se na hometria para ter a noção das distâncias.			
ISRobot (2008)	Seguimento da linha central a tracejado.	Faz-se o varrimento da imagem do centro para as extremidades seguindo-se a linha a tracejado, pela distância à mesma.	Câmara	10º Lugar	[16]
FORMULA UM.2 (2010)	Detecção das linhas da pista através de threshold e a utilização de PID para corrigir trajectória.	Faz-se o varrimento do centro inferior para extremidades para detectar as rectas que sobrepõe as linhas, contínua e descontínua. Dessas linhas calcula-se o diferencial que origina uma recta resultante com ângulo e deslocamento em relação ao centro que serve para corrigir a trajectória através de um PID.	Câmara	?º Lugar ⁷	Segundo o Professor Gil Lopes pertencente ao Departamento de Electrónica Industrial da Universidade do Minho, de onde provêm os robôs, e responsável pela

⁷ Classificação não disponibilizada pela organização.

					equipa MAG'n'UM
MAG'n'UM e BIGMAG (2010)	Algoritmo de Canny para determinar linhas da pista seguido de PID para correcção de trajectória em relação à linha contínua da direita.	Utiliza-se o algoritmo de Canny para neutralizar a imagem e desenhar as linhas, e de seguida efectua-se uma procura radial, do centro para a extremidade, por pontos a branco. Por fim, através de um PID, faz-se uma correcção da trajectória pela linha da direita.	Câmara	Respectivamente ?º Lugar ⁷ e ?º Lugar ⁷	Segundo o Professor Gil Lopes pertencente ao Departamento de Electrónica Industrial da Universidade do Minho, de onde provêm os robôs, e responsável pela equipa MAG'n'UM
Corisco e Os Beta (2010)	Binarização da imagem e análise de pixel, procurando cor branca. Utiliza um PD para corrigir o posicionamento do veículo.	Inicialmente utiliza-se um algoritmo de binarização para detecção das linhas da estrada e de seguida faz-se uma pesquisa (em coordenadas) dos pixéis a branco para detectar a posição do veículo em relação às linhas. A posição é actuada com recurso	Câmara	?º Lugar ⁷	Segundo o Professor Diogo Veríssimo Correia responsável pelas equipas Corisco e Os Beta.

a um PD.

Como se pode ver as câmaras têm um papel fundamental pois permitem, nomeadamente, ter noção da posição do veículo em relação às linhas que delimitam pista, contudo a hodometria (com utilização de encoders) é também utilizada por vezes. Pode pensar-se que se poderia percorrer toda a pista só com a hodometria, mas esta acumula bastante erro pelo que é acompanhada de outros métodos que permitem corrigi-la. Seria necessário também ter um conhecimento prévio da pista, para que fosse conhecida a posição do veículo e assim reagir conforme as necessidades (por exemplo curvas). Assim ela serve apenas para informar acerca da posição longitudinal, sendo corrigida, nestes casos abordados na tabela, pela câmara.

2.1.1.2.Reconhecimento da passadeira

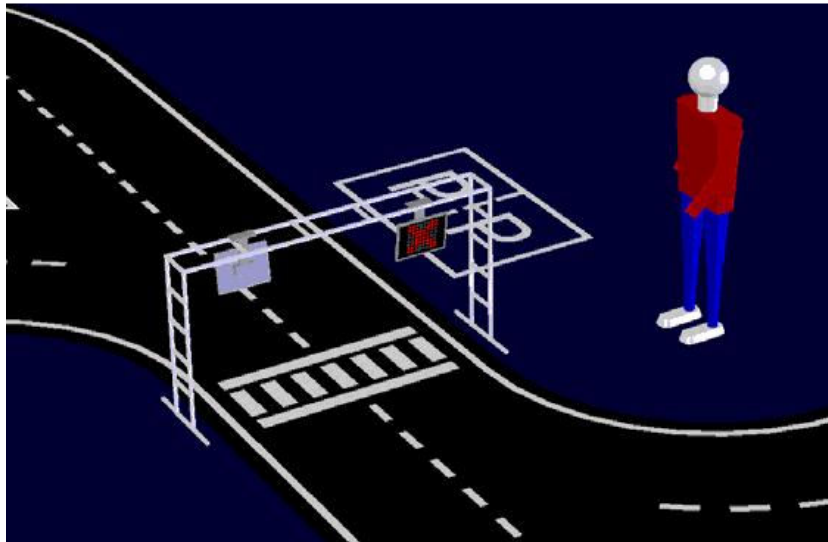


Figura 2.3 - Zona da passadeira (inclui também a sinalização e zona de estacionamento) [3].

O reconhecimento da passadeira é uma tarefa normalmente ligada à análise de imagem. Normalmente trata-se de uma busca por uma linha transversal à pista, linha essa que delimita, em ambos os lados, a passadeira. Após essa busca ter sucesso, procura-se então uma sucessão de linhas compreendidas entre os limites da faixa de rodagem.

O objectivo, nesta zona da pista, é imobilizar o veículo e também corrigir a hodometria. Pode-se também utilizar um sensor baseado em infra-vermelhos (em complemento à câmara), que permita detectar a transição de preto para branco, partindo do pressuposto que a cor preta absorve a luz e a branca reflecte totalmente. Desta forma, quando o sensor detectar a linha pode ser cumprido o objectivo referido anteriormente.

2.1.1.3. Circulação dentro do túnel

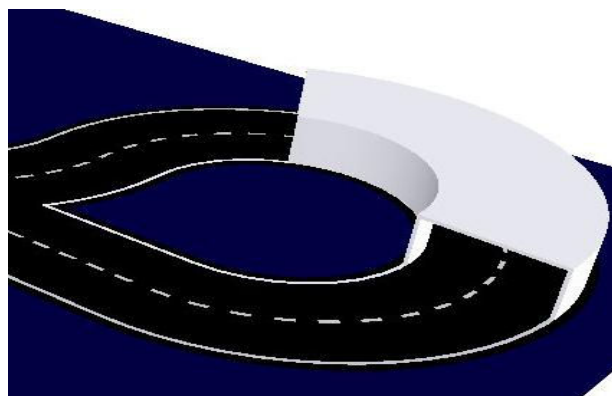


Figura 2.4 - Túnel da pista de Condução Autônoma no FNR [3].

Numa das mangas da prova de condução autônoma é colocado um túnel. Esta alteração pode deitar por terra algumas das metodologias referidas para o seguimento da pista, pois dentro do túnel as linhas delimitadoras da faixa de rodagem desaparecem, dando lugar a paredes. Do mesmo modo é necessário contornar as adversidades.

A maior parte das técnicas passam pela leitura da distância de cada um dos lados do robô, através de sensores de IV, laser ou ultra-sons e com base nisso manter-se o robô, no respectivo lado da faixa de rodagem. Isto traduz-se na quebra da filosofia aplicada para o seguimento da pista. No caso de se querer implementar essa filosofia que passa pela utilização da câmara, haverá então necessidade de iluminação auxiliar, atendendo a que esta é limitada dentro do túnel.

2.1.1.4. Detecção dos obstáculos

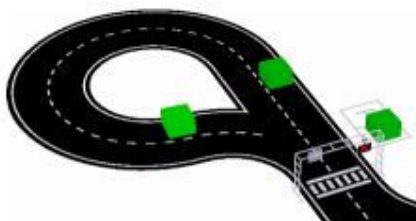


Figura 2.5 - Obstáculos (representados a verde) na pista de Condução Autônoma do FNR [3].

Durante a realização da prova, poderão surgir obstáculos (representativos por exemplo de outros veículos ou mesmo pessoas ou objectos) na via de trânsito, bem como no parque de estacionamento. Tais obstáculos apresentam a cor verde e são de dimensões tais, que permitem a sua fácil identificação. Sempre que sejam identificados na via de trânsito, o robô terá que ser capaz de contorná-lo e seguir pela metade esquerda da faixa de rodagem, até que este seja ultrapassado. Para o efeito terá que identificá-lo.

Pode fazê-lo através do sistema de visão e mediante uma segmentação, ou então pode, mais uma vez, recorrer a sensores de distância por IV ou sonares que permitam saber a que distância se encontram e se por ventura se encontram na frente do veículo. Uma metodologia

interessante e que já foi utilizada na prova de condução autónoma (segundo o professor Gil Lopes da Universidade do Minho e responsável por uma equipa no FNR) passa pela incidência de um foco de luz em frente. Desta forma, e com um algoritmo de detecção de formas é possível detectar um círculo formado pela luz quando embate no obstáculo. O círculo aumenta à medida que o veículo se aproxima do obstáculo.

Para o caso dos obstáculos surgirem no parque de estacionamento, o veículo terá que procurar um lugar livre para estacionar.

2.1.1.5. Navegação e manobras na zona de Parque de Estacionamento

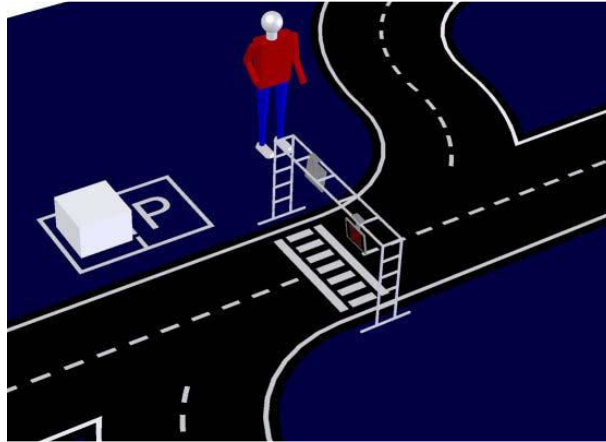


Figura 2.6 - Zona de Parque de Estacionamento da pista de Condução Autónoma (representado por P e com a presença de um obstáculo num dos lugares) [3].

Um dos sinais que podem surgir no painel de sinalização é o que indica ao robô para se dirigir para o parque de estacionamento, como pode ser visto no ponto 6. Desta forma, o robô terá de proceder à procura de um rectângulo fechado com duas zonas distintas (dois quadrados com a letra P), estando um deles, possivelmente, ocupado por um obstáculo. As técnicas utilizadas para o efeito recorrem mais uma vez à imagem da câmara, o que comprova a mais-valia deste sensor para a condução autónoma. Poderia utilizar-se apenas este sensor, para se proceder ao estacionamento, mas apesar de tudo é um sensor que provoca alguma complexidade nos algoritmos, o que leva a uma maior perda de tempo ao extrair-se informação. Assim a colocação de sensores de IV, sonares ou laser facilitarão, tanto a nível temporal, como ao nível de complexidade de algoritmo, a detecção de parque ocupado.

Para além disso, através de sensores de IV de detecção de linhas poderá proceder-se à imobilização do veículo no limite do lugar de estacionamento.

2.1.1.6. Detecção e compreensão do painel de sinalização

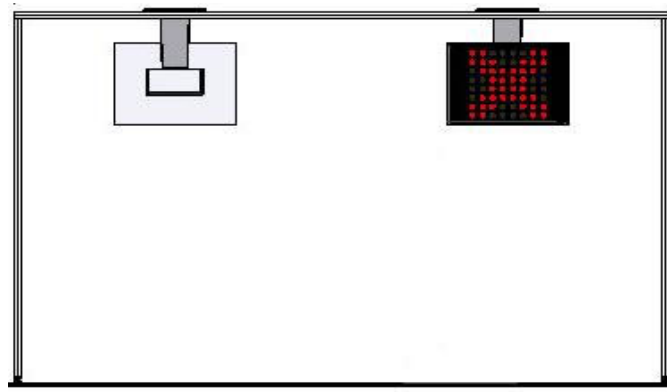


Figura 2.7 - Painel de Sinalização da pista de Condução Autónoma [3].

No centro da pista, junto das passadeiras, surgem os dois painéis de sinais, cada um orientado para uma das metades da pista. Nestes painéis um dos símbolos representados na figura seguinte podem surgir, indicando respectivamente STOP, seguir em frente, fim de prova, seguir pela esquerda e seguir para parque de estacionamento.

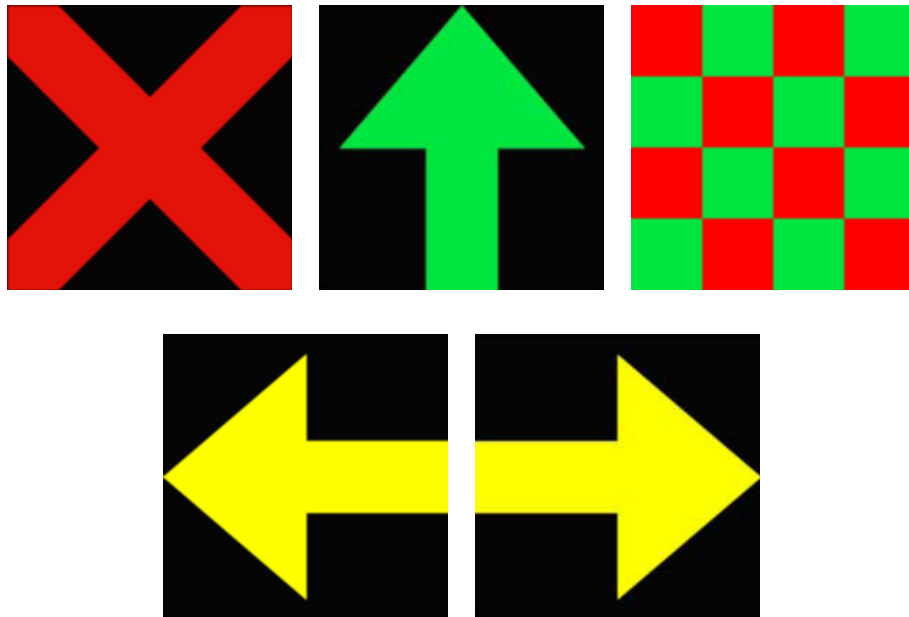


Figura 2.8 - Sinalização do painel (respectivamente STOP, seguir em frente, fim de prova, seguir pela esquerda e seguir para zona de estacionamento).

Após análise de alguns projectos já realizados, quanto à prova de condução autónoma do FNR, pode ver-se as várias metodologias implementadas na tabela seguinte.

Tabela 2.4 - Comparação de métodos utilizados para detecção da sinalização referente ao painel.

<i>Robô</i>	<i>Metodologia Adoptada</i>	<i>Descrição</i>	<i>Sensores Necessários</i>	<i>Classificação no FNR</i>	<i>Referências</i>
ATLAS III (2005)	Identificação da cor e forma dos sinais através das componentes HSV.	O fundo é anulado com base na saturação (S), pois o semáforo apresenta cores mais saturadas, e depois é identificada a cor através da componente de tonalidade (H) com base em filtros de cor (vermelho e verde), sendo que a cor amarela passa nos dois filtros. De seguida é avaliada a forma com base na comparação do centróide da forma e o correspondente ao rectângulo mínimo que envolva a mesma.	Câmara	2º Lugar	[16]
Versa (2006)	Segmentação do sinal de trânsito e análise da cor coincidente com o centro	É utilizada a mesma metodologia que foi utilizada para percorrer a faixa de rodagem, mas neste caso após o	Câmara	2º Lugar	[13]

	de massa.	cálculo dos momentos centrados da imagem é identificada a cor dos pixéis do objecto. A escolha da cor é feita dentro das esferas R, G e B.			
ROTA (2007) e ROTA (2008)	Identificação da cor e forma.	É feita uma separação por cor e posteriormente é analisada a forma com base em características morfológicas. A imagem capturada, no formato YUV22 é segmentada e só no final transformada em HSV através de uma correspondência feita por uma tabela.	Câmara	2º Lugar e 6º Lugar respectivamente	[14] e [15]
FORMULA UM.2, MAG'n'UM e BIGMAG (2010)	Detecção de padrões com cor a partir de um matching no espaço de cores RGB.	Utilizam-se alguns pontos pré-definidos que se procuram por toda a imagem, por varrimento de cor respectiva e posição relativa. Por cada ponto coincidente	Câmara	?º Lugar ⁷	Segundo o Professor Gil Lopes pertencente ao Departamento de Electrónica Industrial da Universidade

		aumenta o número de pontos válidos. Se for atingida a quase totalidade dos pontos é considerado um match.			do Minho, de onde provêm os robôs, e responsável pela equipa MAG'n'UM.
Corisco e "Os Beta" (2010)	Detecção por cor e forma.	É feita a separação por cor e seguidamente analisada a forma, no formato HSV e com utilização do algoritmo HAAR.	Câmara	?º Lugar ⁷	Segundo o Prof. Diogo Veríssimo Correia responsável pelas equipas Corisco e "Os Beta".

Era de esperar a utilização da câmara, visto os sinais serem de várias formas e cores, daí que todas as metodologias sigam técnicas de identificação dessas mesmas características.

No caso de todos os sinais serem de diferentes cores, poderia utilizar-se apenas um dos métodos (análise da cor), mas neste caso temos dois sinais da mesma cor (amarela), logo um despiste de cor seguido de uma análise da forma é imprescindível.

Uma vantagem dada nesta prova é que os sinais não possuem a componente azul da cor, ao contrário do fundo da imagem onde eles estão inseridos. Assim torna-se uma tarefa menos complicada isolar os símbolos para posterior análise das características.

2.1.1.7. Sinais de Trânsito



Figura 2.9 - Os seis sinais de trânsito escolhidos para a edição de 2010 do FNR [3].

Na edição de 2010 do Festival Nacional de Robótica surgiram novos desafios, e a identificação dos sinais de trânsito da Figura 2.9 foi um deles. Este facto deverá ser explorado da melhor forma pois garante, em caso de sucesso, algum bônus temporal.

Com tamanhos equivalentes aos sinais do painel referido no ponto 6, estes poderão aproveitar algumas das metodologias utilizadas para o efeito, isto é a identificação da cor e

forma dos sinais.

Outro aspecto que poderá facilitar a identificação dos sinais é o facto dos mesmos terem um rebordo de cor branca, e isso poderá ajudar a fazer a separação do mesmo com o meio.

2.1.1.8. Identificação da zona de obras

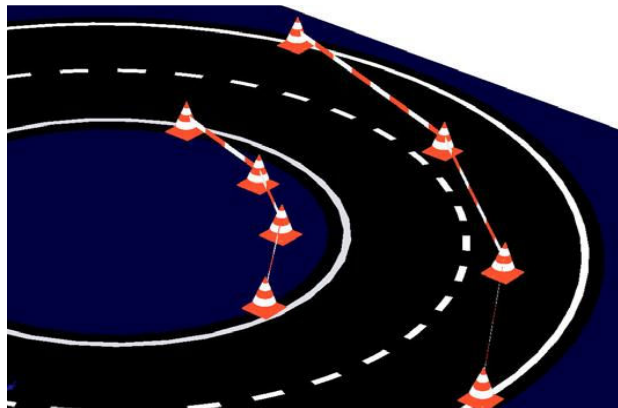


Figura 2.10 - Zona de Obras da pista de Condução Autónoma [3].

A zona de obras, ilustrada na Figura 2.10, é delimitada por cones de cor laranja e branca, espaçados aproximadamente um metro, que estão ligados por uma fita vermelha e branca cuja altura em relação ao chão é fixa (10 cm). Esta zona, muito semelhante à das ruas reais, pode aparecer em qualquer região da pista, portanto terá de ser descortinada a qualquer altura da prova.

Pela razão de ter medidas predefinidas, pode ser percorrida através de um sensor de distâncias da mesma forma que se indicou para a zona do túnel.

Outra técnica para contornar este problema é mais uma vez recorrendo à visão, através da identificação dos cones pela transição da cor branca ou preta para laranja.

A identificação da fita poderá não ser tão trivial dadas as condições de aquisição da câmara (ângulo).

2.3. Tecnologias para Condução Autónoma

Ao longo dos dias, na tentativa de criar um veículo totalmente autónomo, um vasto leque de soluções tem sido criado e explorado a nível tecnológico.

O trabalho desenvolvido, nesta progressiva evolução, caracterizou-se ainda pelo reaproveitamento de tecnologias para integração em sistemas usados em segurança activa, segurança passiva e no alívio do esforço do condutor em situações específicas

Os desafios relacionados com a condução autónoma passam pela automatização dos processos de condução, que requerem algumas especificações cumpridas. São elas:

- Sensores, que capacitam o veículo de perceber o meio em que está inserido;
- Navegação, que capacita o veículo de saber onde está e para onde deve dirigir-se;
- Planeamento, que possibilita ao robô definir uma trajectória e uma velocidade, recorrendo à informação recolhida.
- Actuação, que dá vida ao robô, e que o possibilita de agir conforme o necessário.

O preenchimento de cada um dos requisitos anteriores encontra-se em estágios de desenvolvimento diferentes. Assim, as áreas da navegação e actuação estarão quase conseguidas, sendo que a área dos sensores está também bastante desenvolvida. A área que requer ainda um maior desenvolvimento é a área do planeamento.

2.1.2. Sensores

Para que seja possível recolher informação do meio que rodeia os veículos é necessária a utilização de sensores.

A escolha do tipo de sensores a usar é determinada por vários factores, que devem ser bem ponderados. Está relacionada com factores económicos, com as especificações a que cada veículo tem de obedecer, ou mesmo, com o meio onde o robô é inserido.

Nesta área (condução autónoma), a recolha de informação normalmente recorre à captação de imagens por intermédio de câmaras de maior ou menor custo, ou então a sensores como radares laser, sonares, encoders ou infra-vermelhos.

A utilização de radares laser é uma prática bastante comum, noutros robôs autónomos. O seu funcionamento baseia-se na incidência de um feixe laser num espelho rotativo que reflecte esse feixe em várias direcções. Posto isto é medido o tempo de cada feixe, obtendo-se uma medida de distância a que se encontra um determinado objecto em relação ao aparelho.

Como resultado é obtido um conjunto de pontos, representando uma imagem virtual 2D com a topologia do terreno que se encontra em redor do aparelho.

Apesar deste tipo de aparelho ter a capacidade de examinar uma larga área com boa precisão, tem também algumas desvantagens, entre as quais, a perda de rigor nas medidas em ambientes com poeira e com variações posicionais do sensor causadas por vibração, a não detecção de zonas com água e também o custo do mesmo.

Outro sistema usado para construir uma imagem virtual do meio integrante do robô é um sistema baseado em câmaras. Através da técnica de visão estereoscópica, pode saber-se a distância do robô a obstáculos ou objectos, bem como a posição e o tamanho.

Esta técnica simula o comportamento do olho humano e consiste na recolha de imagens a partir de duas câmaras (como se fossem dois olhos). Assim com a diferença entre frames recolhidos para o mesmo instante de tempo e usando o erro de paralaxe⁸, consegue-se obter uma noção de profundidade.

Os sonares, muito empregues na navegação marítima, permitem saber o posicionamento relativo de obstáculos. O seu funcionamento baseia-se na emissão e recepção de ondas sonoras, medindo-se a variação de tempo entre ambas.

Quanto aos sensores de infra-vermelhos, o seu funcionamento baseia-se na reflexão da luz. Muito úteis para distinguir por exemplo linhas brancas em estradas de asfalto, permitem também saber a distância em relação a objectos e tem a vantagem de ser um sensor relativamente barato.

⁸ É a mudança aparente da posição de um objecto observado, causada por uma mudança da posição do observador.

A utilização dos *encoders* é um método de localização absoluta que não nos permite saber a posição, a distância, bem como as dimensões de obstáculos, se não for conhecida previamente. Por outro lado permitem saber qual a posição absoluta no mundo.

De notar que no caso de sistemas que incorporem vários sensores é fundamental entender bem quais devem ser utilizados e quando devem ser utilizados, sob pena de não ser bem interpretada a informação recolhida e para ser poupada energia do veículo. Dependendo da situação a utilização de um sensor pode ser mais útil ou menos útil.

2.1.3. Navegação

A noção de navegação passa pela identificação, com base em informação disponibilizada pelos sensores, de eventos que requeiram uma alteração de velocidade ou a realização de uma determinada manobra.

A navegação para veículos comerciais já se encontra disponível há vários anos, através do sistema de GPS. Este sistema, de posicionamento geográfico, dá as coordenadas de um lugar na Terra, desde que tenha haja um receptor de sinais de GPS. Foi desenvolvido pelo departamento de Defesa Americano para fins civis e militares.

Porém não se adequa a prova de condução autónoma, visto ser um sistema com erros de alguns metros e que tem a desvantagem de não funcionar em interiores. Será muito útil para navegações em áreas exteriores.

Neste caso, é necessário desenvolver outras estratégias para localização dos robôs, podendo essas soluções passar pela utilização de pontos característicos do circuito ou recorrendo à hodometria.

A hodometria permite, através da medição da rotação das rodas, contabilizar os movimentos do robô e determinar, através do tal, o local onde o robô se encontra. Os sensores mais utilizados para essa tarefa são os encoders.

2.1.4. Planeamento

A noção de planeamento passa pela definição dos perfis de velocidade a aplicar ou qual a trajectória a seguir ao longo de uma determinada manobra após o robô identificar a necessidade para que isso aconteça.

Em termos de software, existe uma grande variedade de algoritmos desenvolvidos que permitem a determinação das zonas da estrada pelas quais deverá circular o robô de forma a cumprir com os requisitos e otimizar a trajectória.

Denota-se, portanto que as técnicas de análise de imagem assumem um papel fundamental na correcta condução, além de se assegurar a segurança de elementos terceiros.

É portanto fundamental dar atenção ao planeamento de trajectórias, sendo o ponto-chave de uma boa condução autónoma.

2.1.5. Actuação

A área de actuação do veículo é, nos dias de hoje, uma área em grande desenvolvimento.

Existem já bastantes sistemas de *drive-by-wire*⁹ que fazem com que o controlo do veículo deixe de estar ligado à parte mecânica. Assim as ordens passam como parâmetros a sistemas electrónicos que irão controlar vários módulos, que no seu conjunto permitem a operação do veículo.

⁹ Tecnologia que na industria automóvel substitui sistemas hidráulicos e mecânicos com sistemas de controlo electrónico, usando actuadores electrónicos e interfaces homem-máquina como sensores emuladores de pedal e volante.

Capítulo 3

Concepção Estrutural

Neste capítulo apresenta-se, de uma forma geral, a infra-estrutura utilizada para o desenvolvimento deste trabalho, bem como do material utilizado para dotar a infra-estrutura de meios que possibilitem atingir o objectivo final de forma segura e objectiva.

3.1. Descrição Geral

A plataforma utilizada para o desenvolvimento do trabalho foi um carro telecomandado eléctrico, o *STRIKE Short Course Truck RTR* da *LOSI*, presente na Figura 3.1. Este porém, teve de ser adaptado de modo a poder ser controlado automaticamente e assim se poder efectuar uma condução autónoma.



Figura 3.1 - Strike Short Course Truck RTR, Veículo utilizado no trabalho.

A escolha deste veículo é explicada pela sua semelhança com um automóvel convencional, visto ser uma escala aproximada de 1/10. Assim assemelha-se à condução autónoma em veículos reais, o que é um passo em frente rumo à aplicação dos algoritmos à realidade.

Apesar de tudo, este veículo não possui todas as “armas” necessárias à absorção dos dados do mundo, bem como para reagir às situações com que venha a deparar-se.

Houve portanto que alterar o veículo, bem como de dotá-lo de novos recursos, entre os

quais o fortalecimento da suspensão. Pode não parecer muito importante, mas apesar de tudo sem uma suspensão equilibrada irão surgir problemas ao nível da hometria. Apresenta-se então o aspecto final do robô.



Figura 3.2 - Aspecto final do robô FEUPCar.

3.2. Hardware

O carro telecomandado já continha algum hardware fundamental para que se procedesse ao seu controlo, sendo que era comandado manualmente à distância através de um par receptor/emissor de rádio frequência que por sua vez controlava a velocidade e direcção electronicamente. Assim, e como não foram disponibilizados quaisquer dados pelo fabricante, foi utilizado tal controlo para que se pudesse entender que tipos de comunicações eram responsáveis pelo funcionamento do equipamento existente.

3.2.1. Motor de Tracção

O carro à disposição contempla apenas um motor de tracção que está acoplado às rodas traseiras através de uma transmissão.

Este motor, um motor DC, tem variações de velocidade dentro de uma gama de tensões de $[-9; +9]$ V, sendo que à tensão zero está desligado, e às tensões de +9V e -9V é impressa a velocidade máxima positiva e negativa, respectivamente. Sabe-se também que o motor consome correntes de pico muito altas (na ordem dos 60A) no momento de arranque.

Tomando em atenção os dados referidos, e sabendo da necessidade de controlar a velocidade do robô houve então que desenvolver o hardware capaz de tornar isso possível.

Atente-se na Figura 3.3.

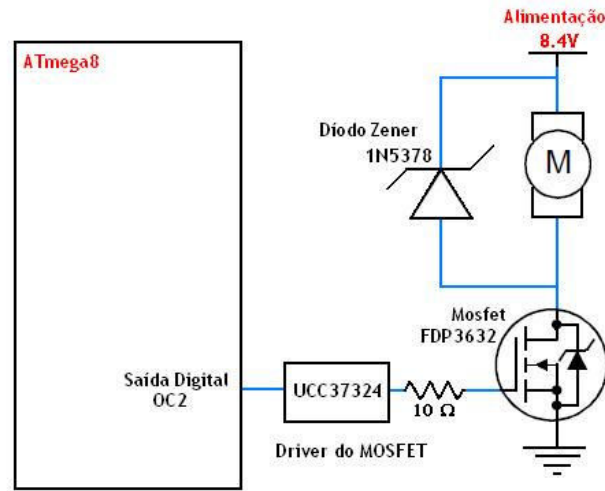


Figura 3.3 - Esquema de potência do controlador do motor.

Pode ver-se que quem será o responsável pelo controlo da velocidade, será o microcontrolador *ATmega8*. Este vai aplicar um sinal em *PWM*, com um *Duty-Cycle* variável no *driver* do *MOSFET* (Figura 3.4) que por sua vez se encarrega de comutar o *MOSFET* (suporta correntes até 80A). Assim, o motor é alimentado pela bateria e controlado pelo *PWM* aplicado no *MOSFET* (Figura 3.5).

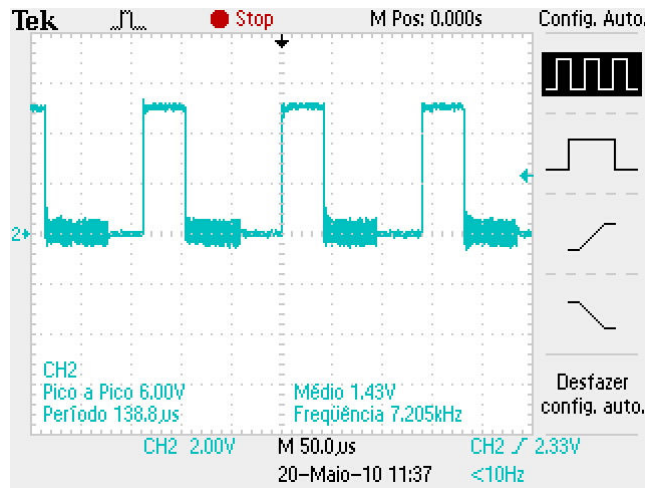


Figura 3.4 - Sinal de PWM à entrada da drive do *MOSFET* - *DutyCycle* de 30%.

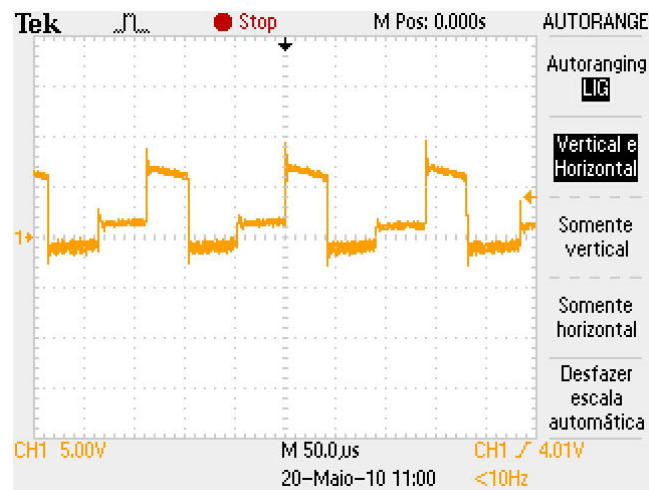


Figura 3.5 - Sinal aplicado ao motor - *DutyCycle* de 30%.

Este controlador para além de ser simples não é caro, cumprindo então com os pressupostos da dissertação (para já não foi abordado o problema de inverter o sentido de marcha). Foi aplicado também um dissipador ao *MOSFET* para contrariar o aquecimento causado pelas altas correntes comutadas. De salientar que na concepção do controlador foram utilizados componentes existentes no laboratório que foram reutilizados.

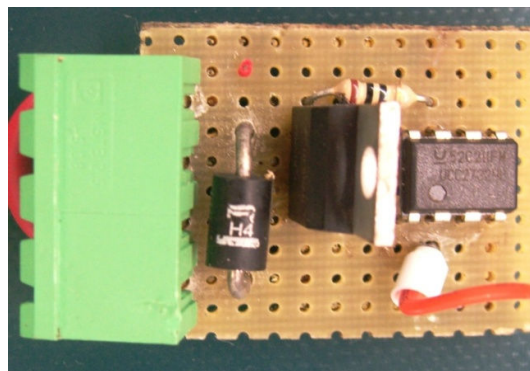


Figura 3.6 - Driver construída para o motor de tracção.

3.2.2. Motor Servo - Direcção

As rodas dianteiras do robô são controladas por um motor servo. Este tipo de motores funciona em malha fechada. Eles sabem a sua posição actual, e quando recebem um sinal de controlo reagem, movendo-se para a posição imposta. Normalmente utilizam um potenciómetro que serve como sensor para saber qual a posição actual. O motor é tão mais preciso quanto maior a qualidade do sinal no potenciómetro.

O sinal de controlo do motor é um sinal PWM, comumente com o período de 20ms, em que a largura de impulso a cada período representa a posição que será imposta.

Para o controlo deste motor não foram necessárias alterações de maior. Inicialmente

foram estudadas as larguras de impulso das posições limite do robô, bem como a posição condizente com a orientação do robô seguir em frente. Pode ver-se o seu funcionamento nessas mesmas posições, na figura seguinte:

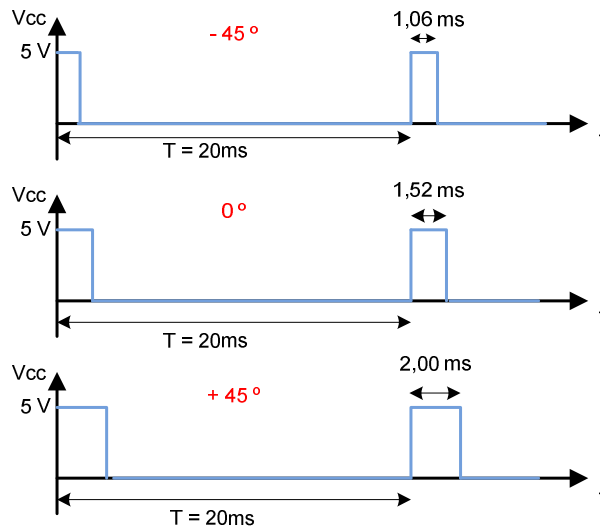


Figura 3.7 - Representação do PWM de controlo do motor servo para as posições limite.

A alimentação do motor é de 5V, e é fornecida por um módulo *Generic ATmega8* utilizado no laboratório (que será descortinado mais adiante), enquanto o *PWM* é concebido por *software* pelo *ATmega8*.

3.2.3. Encoder

Com o objectivo de quantificar a distância percorrida pelo robô e a velocidade do mesmo, acoplou-se um *encoder* directamente ao veio do motor. Foi uma tarefa um pouco morosa olhado ao tempo empregue no desenvolvimento de um suporte que mantivesse o *encoder* bem fixo. Este *encoder* possui uma resolução de 60 impulsos por revolução (360°) e apenas um sinal de saída. Assim sendo não permite que seja detectado o sentido do movimento, o que poderia ser problemático. Pode afastar-se esse problema, de uma forma simples. Sabendo a velocidade aplicada ao motor de tracção (positiva ou negativa) pode saber-se a direcção do robô, logo com esta solução e se se controlar a aceleração e desaceleração de modo a que a inversão de sentido seja feito a muito baixa velocidade, podemos desprezar a dinâmica do veículo. Desta forma obtemos, mais uma vez, uma solução menos dispendiosa que um *encoder* de duas fases.

Apesar de o *encoder* possuir uma resolução de 60 impulsos por revolução, a cada volta completa das rodas são contabilizados 600 impulsos. O responsável pela diferença é a transmissão do veículo. Possui um conjunto de engrenagens que transformam a velocidade de rotação do motor numa velocidade dez vezes superior fornecida às rodas traseiras.

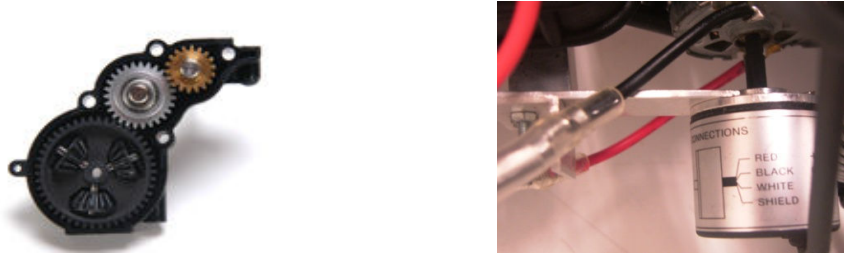


Figura 3.8 - Transmissão (à esquerda) e *encoder da IDEC* colocado no robô (à direita).

3.2.4. Sistema de Alimentação

A alimentação do robô é feita com a bateria pertencente ao veículo telecomandado fornecido para o projecto. Esta é uma bateria própria para veículos deste género, sustentando portanto todos os dispositivos (sensores) aplicados, pois consomem uma energia quase residual quando comparadas com os motores (actuadores).

Assim, o sistema de alimentação do FEUPCar é composto por uma bateria de níquel-hidreto metálico (Ni-MH) de 8.4 Volts.

Estas baterias assumem a responsabilidade de alimentar todos os dispositivos que incorporam o robô.



Figura 3.9 - Sistema de alimentação do robô.

3.2.5. Sensores de Detecção de Linhas Brancas

Os sensores de detecção de linhas têm o objectivo principal de detectar as linhas delimitadoras de estrada como situações limite. Outra das grandes utilidades destes sensores é a detecção das linhas da passadeiras, linhas essas que podem ser utilizadas para se efectuar o acerto da hodometria (principalmente da distância percorrida), bem como para a imobilização do robô junto da mesma. Também pode ser muito útil para o estacionamento, efectuando-se a imobilização do robô antes de este ultrapassar o limite.

A solução implementada consiste num conjunto de dois reflectores ópticos instalados na frente do robô, apontados para o chão e ligeiramente inclinados de forma que estes nos informem da detecção das linhas antes mesmo de estar sobre elas. Com isto o robô tem possibilidade de reagir em tempo útil, evitando penalizações na prova.

Os infra-vermelhos emitidos por estes dispositivos são reflectidos no chão e detectados por um fotodetector. A razão da escolha dos reflectores ópticos deve-se à curta distância que estes têm de medir.

Houve então que se efectuar uma escolha do par emissor/receptor, e passar então para a implementação do circuito devido.

Em relação ao emissor de infravermelhos, foi escolhido um díodo emissor de infravermelhos com pequena abertura de feixe de emissão e com grande intensidade. Já quanto ao fotodetector escolhido, optou-se por um fototransistor com grande sensibilidade, devido ao seu ganho interno. Deste modo poupa-se no desenvolvimento de um circuito amplificador.

Com esta solução, possuímos condições necessárias para que a corrente de polarização do emissor seja menor e assim conseguimos diminuir o consumo de energia.

O circuito desenvolvido está representado na Figura 3.10. O fototransistor encontra-se montado em emissor comum, e R3 é ajustado de modo a que se obtenha um valor acima de 2.5V para detecção das linhas e abaixo desse valor para reflexão noutras superfícies (estrada na prova de condução autónoma). A presença do potenciómetro tem a vantagem de possibilitar a regulação dos valores aquando da mudança de cores de linhas e pista.

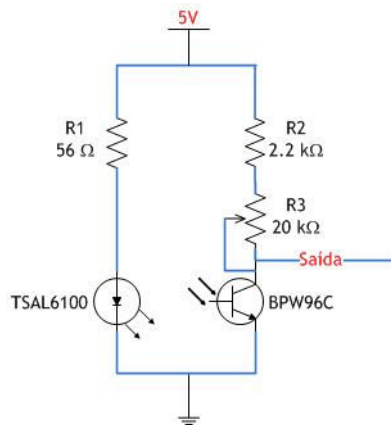


Figura 3.10 - Esquema eléctrico dos sensores de detecção de linhas brancas.

A alimentação do sensor é feita mais uma vez pelo módulo *Generic ATmega8*, enquanto a saída é lida por uma porta digital do micro-controlador. Os valores acima mencionados acerca dos valores para detecção de linha (ou não) estão relacionados com a leitura das portas digitais do micro-controlador.

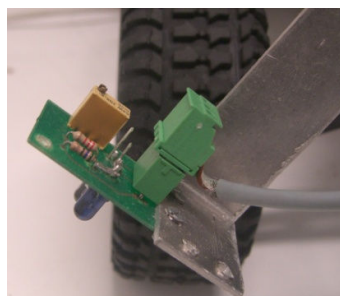


Figura 3.11 - Sensor de detecção de linhas

3.2.6. Sensores de Obstáculos e de Paredes

De forma a detectar os obstáculos e também para averiguar a distância às paredes do túnel, que poderão surgir no percurso, recorreu-se aos sensores de infra-vermelhos, *Sharp GP2D12*. Estes sensores, que estavam já disponíveis no laboratório, possuem um alcance compreendido entre os 10 e 80 cm e devolvem tensões entre os 0 e os 2.6 Volts [20].

Para aumentar a sua utilidade e para uma correcta leitura de distâncias, foram colocados de forma estratégica no robô.

Veja-se a Figura 3.12.

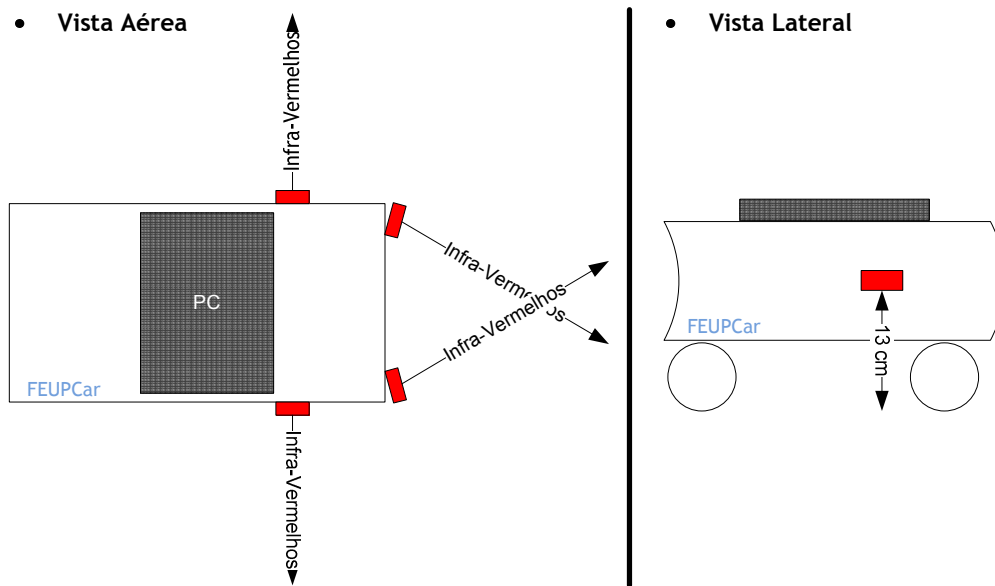


Figura 3.12 - Esquema de posicionamento dos sensores.

Pode verificar-se que os sensores frontais (sensores de obstáculos) estão colocados de forma que possam identificar qualquer tipo de objectos na frente do robô, ao contrário do que aconteceria se eles estivessem orientados perpendicularmente ao mesmo (neste caso estão orientados diagonalmente).

Quanto aos sensores de detecção de paredes (sensores laterais) optou-se por colocá-los a 13 cm do chão para que fosse acrescentado um novo apoio à sua navegação autónoma, a tentativa de detecção das fitas que unem os cones da zona de obras da pista de condução autónoma do FNR.

De modo a obter-se uma distância a partir da tensão de saída do sensor terá que se determinar a curva característica. Tal será explicado mais à frente.



Figura 3.13 - Sensor Sharp GP2D12.

3.2.7. Módulo Generic ATmega8

Para desenvolver a *placa central* do FEUPCar, responsável pelo controlo dos actuadores e leitura dos sensores, bem como da interligação entre as camadas de nível superior e inferior, foi utilizado o módulo *Generic ATmega8*, desenvolvido na FEUP, já mencionado anteriormente. Este módulo incorpora um microprocessador *ATmega8*, uma interface de comunicação RS-232, e também um regulador de tensão que faz a conversão da tensão da bateria para os 5 Volts pretendidos por alguns dos componentes.



Figura 3.14 - Módulo *Generic ATmega8*.

3.2.8. Esquema geral do *Hardware*

Para que se perceba melhor o hardware desenvolvido bem como a sua interacção, apresenta-se de seguida um diagrama geral de blocos.

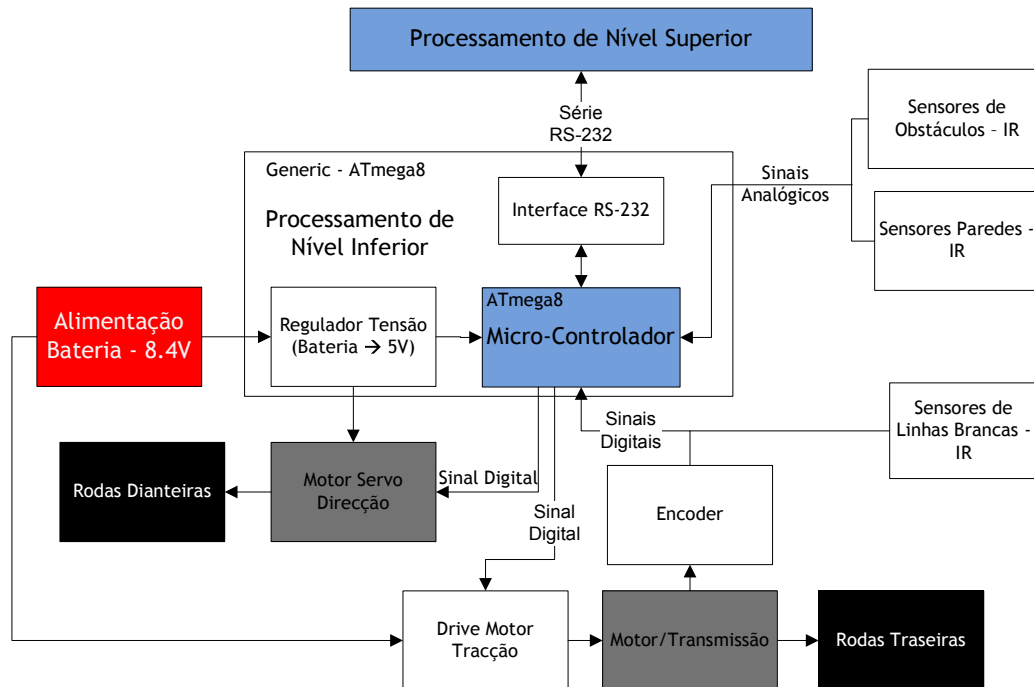


Figura 3.15 - Diagrama de blocos do *Hardware* desenvolvido.

Pode ver-se para além de todo o *hardware* já referido, uma referência ao processamento de nível superior e inferior. Isto porque neste projecto houve o cuidado de efectuar a divisão entre as acções que necessitam de menor processamento - processamento de nível inferior (como acontece no accionamento e leitura dos sensores) e as acções de maior exigência - processamento de nível superior (algoritmos de controlo e navegação).

A colocação de câmaras, elemento muito útil na prova de condução autónoma, foi acautelada, podendo coexistir nesta estrutura. Sabendo da exigência que é a análise de imagens, elas surgirão sempre ligadas ao processamento de nível superior ligadas por *Firewire* ou *USB* até fazendo uso da câmara incorporada no computador portátil.

3.3. Software

A organização da arquitectura de *software* de um sistema afecta totalmente o desempenho de um robô móvel. A integração de múltiplos sistemas de sensores, sem uma aquisição e pré-processamento de dados adequada e eficiente, reflecte-se numa enorme ineficácia de desempenho por parte do robô.

Nesta secção apresentam-se os vários componentes que constituem a plataforma de software, assim como as suas principais características.

Será ainda objecto de estudo nesta secção, o desenvolvimento e a calibração do controlador de velocidade do motor responsável pela locomoção do *FEUPCar* bem como a modelização dos sensores de IV para detecção de paredes e obstáculos.

O *software* de decisão do robô foi realizado numa aplicação desenvolvida na camada de processamento superior, usando a plataforma de desenvolvimento *Lazarus* que incorpora a linguagem *Free Pascal* sobre o sistema operativo *Windows*. Salienta-se também que a utilização desta plataforma permite transpor facilmente o *software* de decisão e controlo do

sistema operativo *Windows* para o sistema operativo *Linux*, necessitando apenas de ser novamente recompilado. Pelo facto da linguagem *Free Pascal* ser orientada a objectos, permite o fácil acesso aos dispositivos e sistemas periféricos, assim como a fácil construção de interfaces para o utilizador.

Já o desenvolvimento dos PWM's dos actuadores, bem como a aquisição do valor dos sensores estará ao cargo do microprocessador *ATmega8*.

3.3.1. Camada de Nível Superior

A camada de processamento superior é responsável pelo controlo e decisão do comportamento do robô.

Como se refere anteriormente, a plataforma *Lazarus* utilizada para implementar o processamento da camada superior é um ambiente de desenvolvimento integrado desenvolvido para o compilador *Free Pascal* sendo este um compilador de *Object Pascal*. Desta forma, o software de controlo e decisão foi desenvolvido baseado na programação orientada a eventos. Estes eventos são definidos como internos ou externos, despoletados por acções externas, como recepção de tramas via RS-232, ou por acções internas, como acções de botões.

Nesta camada o ciclo de controlo é apenas despoletado a partir do momento em que é feita a ligação por porta série. Interiormente, no ciclo de controlo - ciclo principal - é feita a divisão entre o modo manual, em que pode comandar-se o robô manualmente, e o modo automático, controlo esse que servirá para implementar a autonomia do robô.

O modo manual foi muito útil para testes de hodometria, bem como para testar o funcionamento dos motores.

Pode ver-se o funcionamento geral desta camada no diagrama a seguir representado.

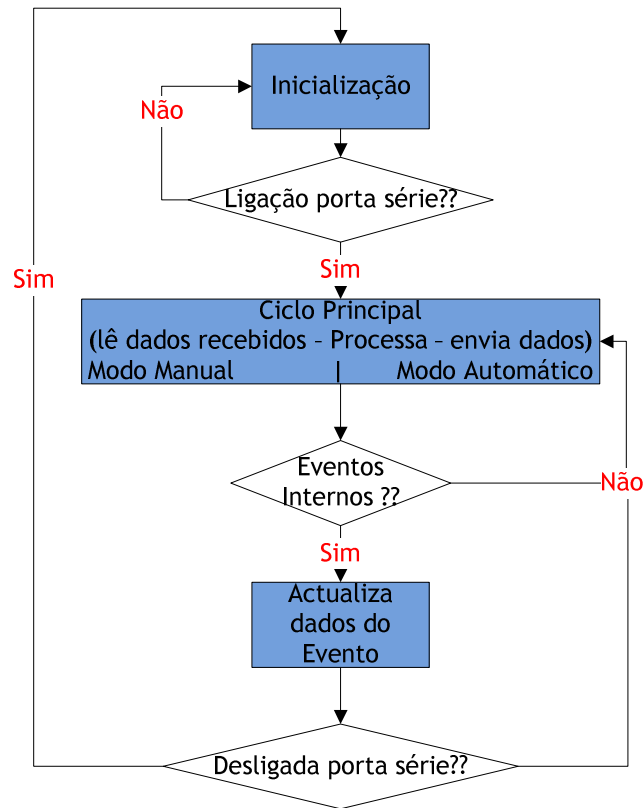


Figura 3.16 - Diagrama geral do funcionamento da camada de nível superior.

Já foi referido que é nesta camada que é realizado o algoritmo mais complexo. Pode acrescentar-se também que é desenvolvido boa parte da modelização dos sensores de infravermelhos assim como o desenvolvimento da interface de controlo e visualização dos dados do *FEUPCar*.

3.3.1.1. Modelização dos Sensores de Obstáculo e de Paredes

A. Primeiro Passo:

Após o desenvolvimento do *hardware*, é possível retirar a informação útil (no caso a distância) destes mesmos sensores.

Os dados fornecidos são valores de 10 bits originados pela leitura do ADC do microprocessador, assim teve que se efectuar uma conversão destes valores para outros em tensão, da seguinte forma:

$$D_{Volts} = \frac{\text{Valor}_{\text{enviado pelo nível inferior}} \times V_{\text{ref}}}{2^{10}} \text{ (Volts)}, \quad (3.1)$$

onde V_{ref} representa o valor de referência do ATmega8, 5V neste caso.

B. Segundo Passo:

Perante este avanço pode passar-se à determinação da distância propriamente dita.

Na *datasheet* referente ao sensor utilizado [20] é fornecida uma curva característica, mas apesar de tudo não é disponibilizada a sua fórmula matemática. Houve portanto que

efectuam-se vários ensaios para estabelecer uma relação tensão/distância.

Como resultado de tais ensaios obtém-se o gráfico da Figura 3.17.

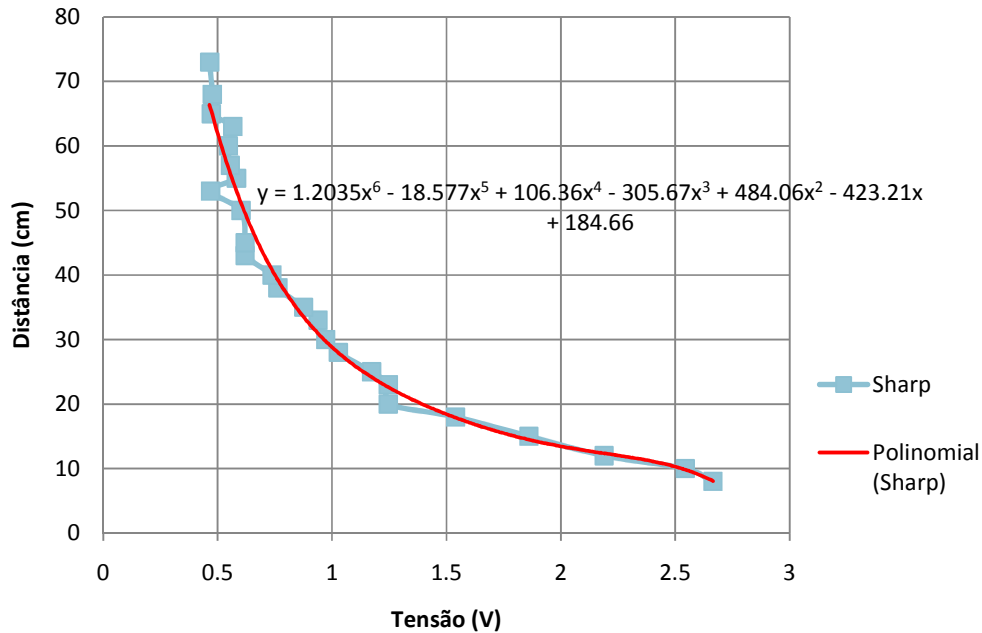


Figura 3.17 - Gráfico representativo da relação tensão/distância dos sensores de distâncias.

Da análise do gráfico pode retirar-se que a curva característica é uma aproximação polinomial de 6º grau, que se aproxima muito dos valores medidos no ensaio.

Uma aproximação de 5º grau já se aproximava muito desta curva, mas apesar de tudo, como é efectuado na camada de nível superior, onde a capacidade de processamento não é prejudicada por esta alteração, optou-se pelo grau mais alto.

De referir que apesar de o ensaio ter sido efectuado para valores desde 1cm até 125 cm, a curva característica foi calculada para valores compreendidos entre os 8 e os 73 cm. Tal tem uma explicação. Para além de alguns dos valores fora dos limites impostos serem muito variantes e pouco credíveis, esta foi a curva característica mais semelhante à curva definida pelos pontos do ensaio.

C. Terceiro Passo:

Pelos testes efectuados com os sensores, notou-se, mesmo assim, alguma variância de valores em volta do valor correcto da distância. Para corrigir esse aspecto foi aplicado um filtro digital usando a plataforma de desenvolvimento *Lazarus*. O filtro escolhido para o efeito foi um filtro de mediana pela razão de este ser muito útil para caracterizar observações que tenham uma tendência central, como é o caso. Pode ver-se o algoritmo de mediana utilizado no Anexo B, que se baseia no cálculo da mediana de um vector circular de dez medidas de cada sensor.

Finalmente o robô está preparado para a aquisição das distâncias dos sensores.

3.3.1.2. Interface Desenvolvida

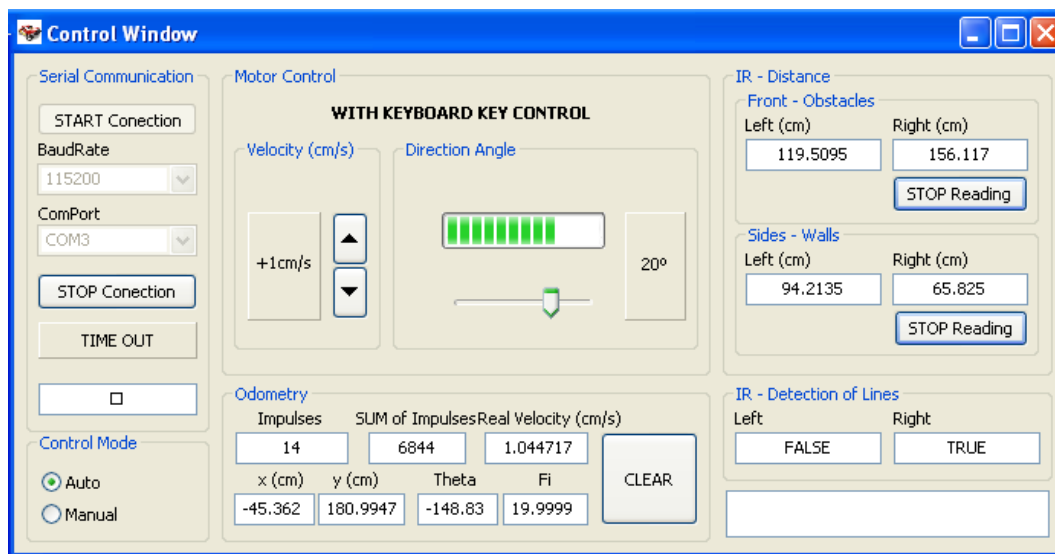


Figura 3.18 - Interface desenvolvida sob a plataforma *Lazarus*.

A interface foi desenvolvida para visualização de todos os dados respeitantes ao robô, bem como para controlo do mesmo.

Contempla os modos Manual e Automático, e a visualização de dados como velocidades actuais, velocidades de referência, ângulos de direcção, estado dos sensores e posicionamento relativo do robô.

O modo Manual permite controlar o robô como se fosse um carro telecomandado, mas para isso terá que aceder-se remotamente ao *PC* portátil localizado no robô.

Já no modo Automático, o controlo estará ao cargo dos algoritmos implementados para o efeito.

3.3.2. Camada de Nível Inferior

A camada de nível inferior está assente num microprocessador *ATmega8*. Aqui são realizadas acções de menor exigência a nível de processamento, como é o caso da leitura dos sensores e a geração dos *PWM*'s responsáveis pelo controlo dos actuadores.

A programação está ao cargo do compilador *Win-AVR* assente na linguagem C.

Nesta secção são referidos os controlos que carecem de maior esclarecimento, como é o caso dos controladores dos motores.

3.3.2.1. Controlo do Motor de Tracção

Na secção de hardware foi explicado o driver implementado para fazer o controlo do motor. Foi também referido que seria decisivo a aplicação de um *PWM* com um *Duty-Cycle* variável.

Como o microprocessador já disponibiliza três temporizadores para esse mesmo objectivo e contém algumas predefinições para o tal, a tarefa foi facilitada. Foi utilizado o *TIMER2*, de oito bits, configurando o mesmo para funcionar no modo *FAST-PWM*. Perante isto só haverá necessidade de actualizar o registo responsável pelo *Duty-Cycle* - *OCR2* (*Duty-Cycle*=100%

equivale a um OCR2=255).

3.3.2.1.1. Controlador de velocidade do Motor

O controlador de velocidade é responsável pela actuação no motor acoplado às rodas, garantindo que o robô se encontra a rodar à velocidade de referência estabelecida. Deste modo, exige o conhecimento prévio do modelo dinâmico aproximado do robô.

O controlo da velocidade foi também implementado no microprocessador. Sendo que para o robô circular a uma certa velocidade é dada uma referência em número de impulsos por tempo de amostragem.

Registe-se que a referência é fornecida pela camada de nível superior.

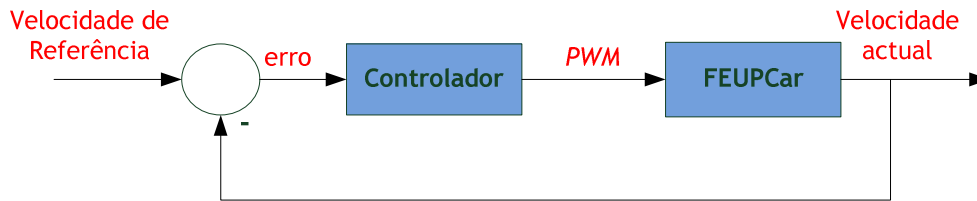


Figura 3.19 - Diagrama de blocos do controlador de velocidade do motor de tracção.

De seguida são descritos os vários procedimentos levados a cabo para obtenção do controlador de velocidades do motor.

A. Determinação do Modelo Dinâmico do FEUPCar:

Os movimentos do robô surgem como resultado da aplicação de tensões DC no motor, sendo criado um determinado binário. Estes binários são convertidos pelas próprias rodas em força de tracção paralelas ao chão que obrigam o robô a descrever uma trajectória.

Como anteriormente mencionado, a calibração de um controlador de velocidade para o motor requer o conhecimento do comportamento do robô. Este pode ser descrito por um modelo dinâmico aproximado.

Através da aplicação de um degrau de tensão no motor e sabendo o desenrolar da velocidade no robô podemos encontrar um modelo dinâmico aproximado.

Para isso adaptou-se a aplicação feita em Lazarus para recolher os dados referentes à velocidade (em impulsos por período de amostragem) do robô e posteriormente foram aproveitados os dados para a construção gráfica da evolução temporal da velocidade.

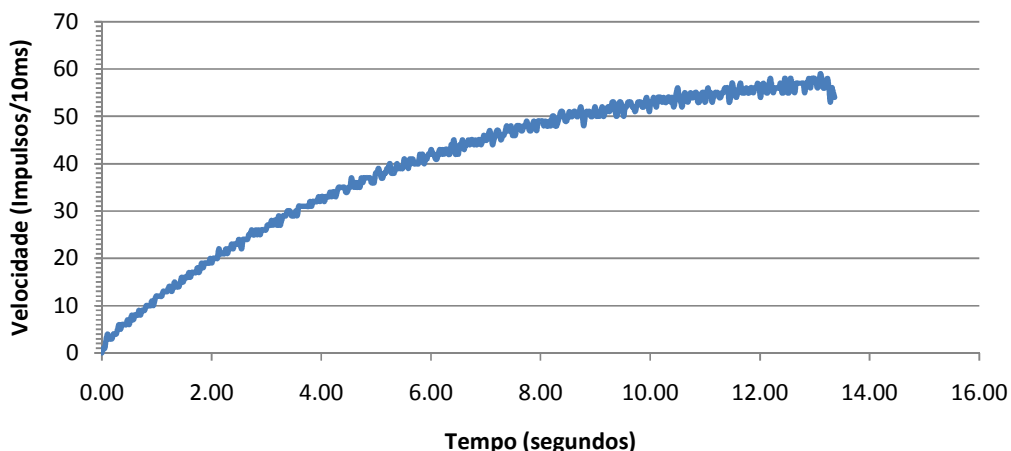


Figura 3.20 - Evolução temporal da velocidade do robô para uma tensão de 3.8 V.

Pode verificar-se que a resposta é semelhante a um sistema de 1ª ordem. É possível verificar também que o motor precisaria de maior tempo (ou seja uma maior distância) para estabilizar. Apesar de tudo, e porque estamos já perto do valor da fase de estabilização, o gráfico apresentado é viável para a determinação do modelo.

O tempo de estabelecimento do sistema é 63% da velocidade final o que equivale a dizer que é de aproximadamente 5 s.

A função de transferência de um sistema de 1ª ordem é representado por:

$$G(s) = \frac{Y(s)}{X(s)} = \frac{G_{DC}}{\tau s + 1} \quad (3.2)$$

Desta forma, a correspondente equação diferencial é:

$$\tau \cdot \dot{y}(t) + y(t) = G_{DC} \cdot x(t) \quad (3.3)$$

Logo a expressão da resposta do sistema a um degrau na é dada pela equação:

$$v(t) = G_{DC} \cdot \left(1 - e^{-\frac{t}{\tau}}\right) \cdot u(t), \quad (3.4)$$

onde $v(t)$ é o valor de saída, neste caso a velocidade, G_{DC} é o ganho DC, τ a constante de tempo para regime permanente e $u(t)$ é o degrau aplicado no motor.

Uma vez definida a equação que modela o robô, é possível determinar os parâmetros da equação, nomeadamente G_{DC} e τ .

Para isso utilizou-se o método dos Mínimos Quadrados, usando os dados que originaram o gráfico da Figura 3.20, estimando assim os parâmetros do robô.

Obteve-se o valor de 16.423 para G_{DC} e 5,373 segundos para τ .

De seguida é mostrado o gráfico resultante da aproximação a um sistema de 1ª Ordem, onde a curva a vermelho representa a aplicação do modelo de 1ª Ordem.

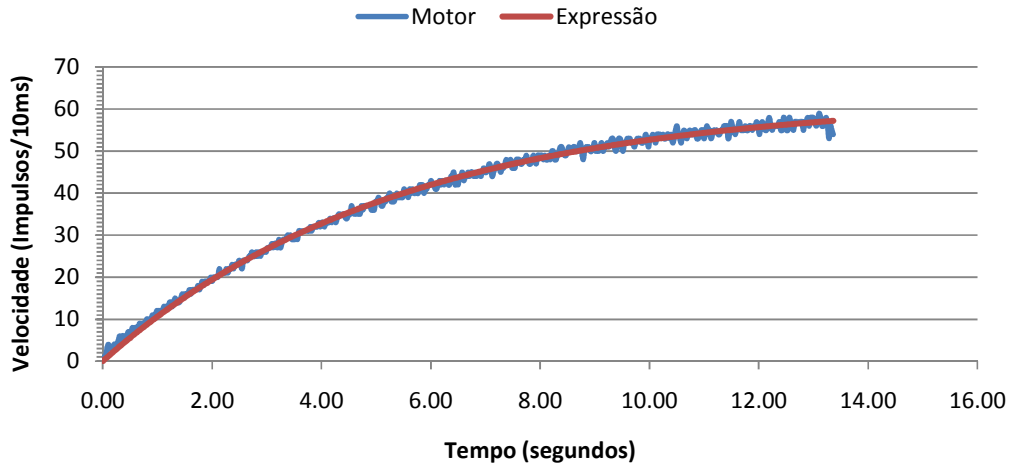


Figura 3.21 - Resultados da aproximação a um sistema de 1ª Ordem.

B. Dimensionamento do Controlador:

O controlo da velocidade foi feito a partir de um controlador em malha fechada do tipo PI, com filtro de entrada e *feed-forward*. Pode ver-se o diagrama de blocos do controlador na figura seguinte.

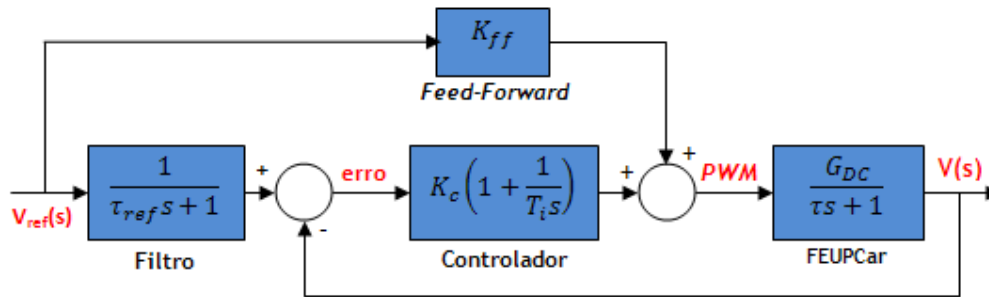


Figura 3.22 - Diagrama de blocos do sistema controlado.

O controlador PI introduz um zero na função de transferência em malha fechada. Esse zero pode provocar sobre-elongações indesejadas. O filtro de entrada permite anular essa sobreelongação através da anulação do zero referido e o *feed-forward* tem o objectivo de aumentar a velocidade de resposta do sistema, sem afectar a estabilidade, o que poderia ser vantajoso porque o robô tem uma resposta pouco rápida.

Com este controlador obtemos a seguinte função de transferência, com “ τ_{ref} ” ajustado de modo a cancelar o zero, e sem o ramo de “*feed-forward*”:

$$\frac{V(s)}{V_{ref}(s)} = \frac{\frac{1}{\tau_{ref}s + 1} \cdot K_C \left(1 + \frac{1}{T_i s}\right) \cdot \frac{G_{DC}}{\tau s + 1}}{1 + \frac{K_C \cdot \left(1 + \frac{1}{T_i s}\right) \cdot G_{DC}}{\tau s + 1}} = \frac{\left(\frac{K_C \cdot G_{DC}}{T_i \tau}\right)}{s^2 + \left(\frac{1 + K_C \cdot G_{DC}}{\tau}\right)s + \left(\frac{K_C \cdot G_{DC}}{T_i \tau}\right)} \quad (3.5)$$

Obtida a função de transferência, terá que calcular-se os parâmetros K_C e T_i . Para isso recorreu-se ao protótipo de *Bessel* [19], fixando-se um tempo de estabelecimento de 1 segundo.

Calcularam-se então os pólos da função de transferência através dos pólos de *Bessel* resultando assim, para o valor do ganho do controlador (K_C), 2.59 e para o valor do tempo integral (T_i) e constante de tempo do filtro (τ_{ref}), 0.36 s.

Temos portanto o controlador dimensionado, resta agora testar o controlador e aplicá-lo no robô.

Para testá-lo, utilizou-se o simulador do *MATLAB*, com o circuito de simulação que se apresenta Figura 3.23, donde se poderia retirar o gráfico da resposta do sistema.

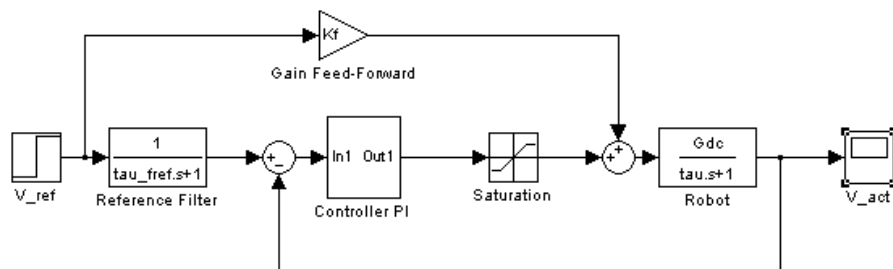


Figura 3.23 - Diagrama de blocos do controlador simulado em *MATLAB*.

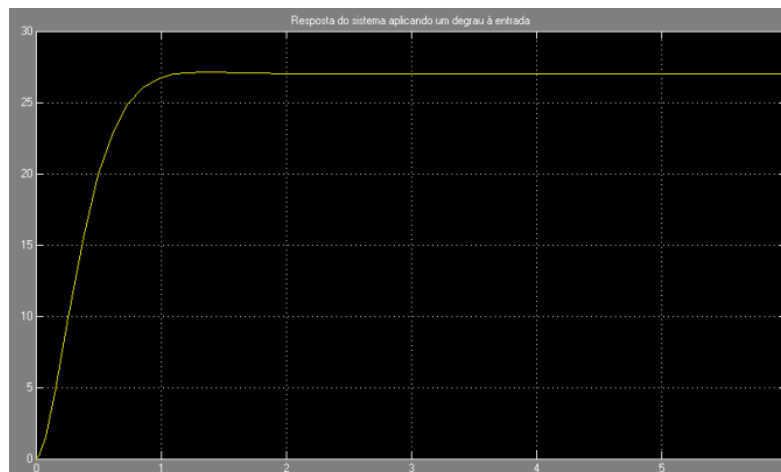


Figura 3.24 - Resposta do controlador dimensionado a um degrau.

Com isto pode dizer-se que a resposta do sistema está dentro do esperado, visto o sistema estabilizar ao fim de 1s e com o valor do degrau aplicado.

Com o simulador pôde-se também chegar à conclusão que não foi necessária a utilização do *Feed-Forward*.

C. Aplicação da Controlador simulado ao FEUPCar:

Testado que está o controlador, havia necessidade de comprovar a sua eficácia. Assim, foi aplicado, ao robô, o controlador mas no seu equivalente discreto com uma velocidade de referência, surgindo o resultado apresentado na Figura 3.25.

Saliente-se que apesar de o *driver* do robô estar calibrado para enviar dados para o computador com um período de 40ms, o controlo de velocidade é realizado internamente com período igual a 10ms.

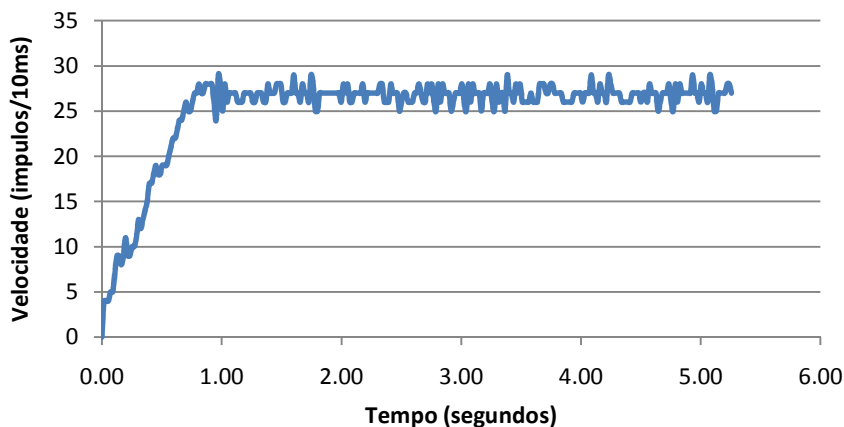


Figura 3.25 - Resultados do controlador de velocidade do motor

Pode registar-se que o controlador de velocidade está capacitado para corrigir e estabilizar a velocidade pretendida.

3.3.2.2. Controlo do motor de Direcção

Pôde ver-se na secção de *hardware* como teria de ser tratado o funcionamento do motor servo. Pôde ver-se também entre que valores variavam os impulsos limite do *PWM* a aplicar.

Desta forma, o que foi feito, passou pela divisão do intervalo [-45;45]° em noventa e uma posições diferentes e assim obtêm-se os impulsos pretendidos para os vários ângulos da direcção do robô.

O *PWM* foi implementado no microprocessador, mais uma vez utilizando as potencialidades do *ATmega8* no que toca à geração de *PWM*'s. Recorreu-se, desta vez, ao *TIMER1* de 16 bits de forma a obter mais precisão, e, programando para o modo de *FAST-PWM*, resta só definir o período para 20ms. Falta agora actualizar o valor do registo *OCR1A*, registo este que é responsável pela definição da largura do impulso a aplicar ao motor.

3.4. Protocolo Comunicação

Para que fosse possível o veículo transmitir informação desde o *PC* até ao controlador do mesmo (e vice-versa) sem perder informação e de modo a que esta não seja enviada de forma errada foi implementado um protocolo de comunicação via RS-232 do tipo *Master-Slave*.

Na Figura 3.26 é revelado em traços muito gerais o seu funcionamento.

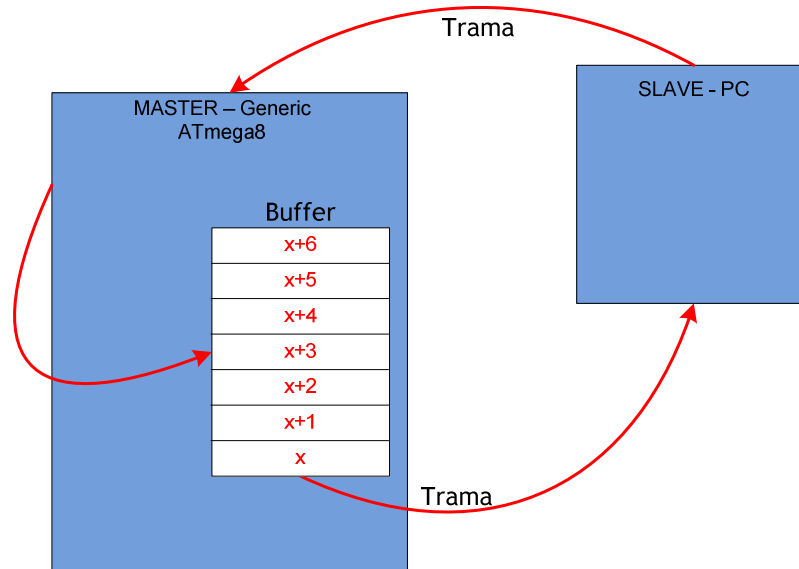


Figura 3.26 - Esquema do protocolo de comunicação entre PC e controlador (*Generic ATmega8*).

Na Figura 3.15 foi apresentada a arquitectura de processamento do robô. Pôde ver-se que está distribuída em duas camadas distintas.

O processamento da camada de nível inferior foi desenvolvido num micro-controlador (*ATmega8*), sendo que este é o *Master*, enquanto a camada de nível superior é implementada num PC portátil - *Slave*.

O *Master* - camada de nível inferior - envia dados a cada 40ms, ditando assim a cadência.

Num funcionamento normal é colocada uma trama no buffer do *Master* e daí enviada para o *Slave*. Depois, e antes que seja ultrapassado o tempo limite, é recebida uma trama de resposta. Se entre esse intervalo não acontecer o esperado, é mantida a trama no *buffer*.

Com o envio falhado de sete tramas, ou seja, depois do preenchimento do *buffer*, o robô entra num estado designado de *time-out*, em que o motor de tracção é desligado.

3.4.1. Detalhes do Protocolo

Tenha-se em atenção novamente a Figura 3.26. Quando o *Master* envia uma trama, que é primeiramente colocada no *buffer* e daí enviada, fica à espera da confirmação da mesma até que seja ultrapassado o limite de tempo (cadência de envio de tramas). Se essa trama não for confirmada não é retirada do *buffer* e deste modo irão permanecer duas tramas. Quando o *buffer* ficar completo, ou por outra, seja ultrapassado o seu tamanho limite (sete tramas), é então enviado o *time-out* para o *Slave* e desligado o motor.

Para que uma trama seja confirmada e assim apagada do *buffer*, tem que ser recebida

uma outra com o mesmo identificador¹⁰, esta enviada pelo *Slave*.

Após o restabelecimento da comunicação (transição do modo *time-out* para modo normal), o *Master* envia todas as tramas contidas dentro do *buffer*.

Com este protocolo garante-se que os dados são correctamente enviados, através de um *checksum* e de uma confirmação da trama, e enviados pela ordem devida, graças ao identificador. Além disso é garantido um certo determinismo temporal que suporta algumas falhas de comunicação transitórias, devido ao *buffer*.

De notar que a cadência de envio de cada uma das tramas por parte do *Master* é de 40ms. Pode perceber-se melhor pelo diagrama da Figura 3.27 o funcionamento do protocolo.

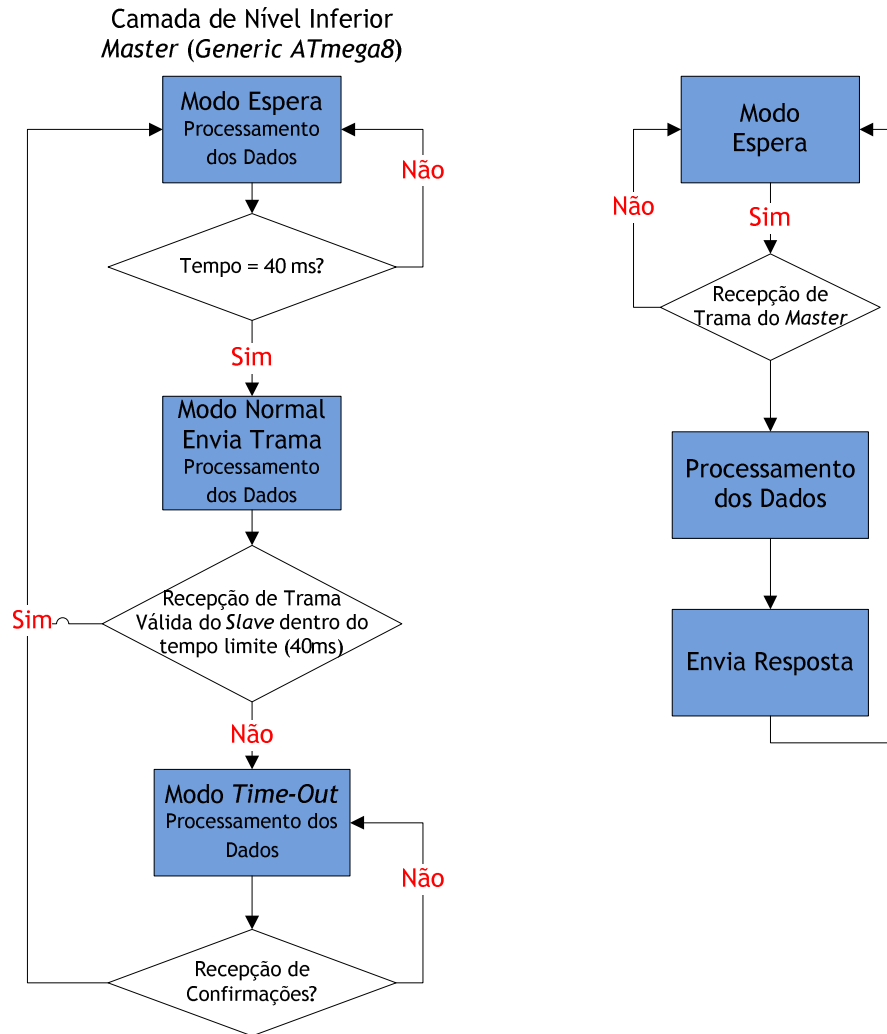


Figura 3.27 - Diagrama de blocos do funcionamento do protocolo entre o *Master* e o *Slave*.

¹⁰ Indica o número da trama. Tem o propósito de avaliar se a ordem das tramas foi trocada. Também é utilizada para permitir ao *Master* interpretar as confirmações de tramas.

Explicadas as linhas gerais do funcionamento, pode então passar-se aos pormenores da trama.

3.4.2. Composição da Trama

A trama é composta por algumas partes que interessa de sobremaneira abordar. Veja-se na tabela seguinte a constituição da mesma:

Tabela 3.1 - Constituição da trama

<i>Conteúdo:</i>	<i>Descrição:</i>	<i>Número de Bytes</i>
C	Caracter de inicialização	1 Byte
L ou R	No caso de não haver <i>time-out</i> é enviado o carácter “L”. Caso contrário é enviado um “R”.	1 Byte
I	Identificador. Indica o número da trama. Tem a finalidade de se poder avaliar se a ordem das tramas foi trocada e para permitir saber interpretar as confirmações de tramas.	1 Byte
Dados	Dados enviados pelas camadas de processamento.	<i>n</i> Bytes
Checksum	Código usado para verificar a integridade dos dados transmitidos.	1 Byte

Saliente-se que a trama enviada pelo *Master* tem um tamanho fixo bem como a trama recebida, podendo este tamanho ser diferente de uma para a outra.

3.4.2.1. Composição da Trama no Projecto

Para este projecto a trama enviada pela camada inferior tem um tamanho diferente e inferior ao tamanho da trama da camada superior. Isto deve-se sobretudo ao facto da camada superior ser a responsável pelo tratamento dos dados obtidos e pelos algoritmos de controlo mais exigentes ao nível de processamento.

Para que se saiba os dados enviados pelo controlador são:

- O número total de impulsos do encoder, a cada período de envio (40ms) - 2 Bytes.
- A informação em tensão (0-5V) dos quatro sensores IR (dois de obstáculos e dois de paredes) - 2 Bytes + 2 Bytes + 2 Bytes + 2 Bytes.
- A informação relativa aos dois sensores de detecção (ou não) de linhas brancas - 1 Byte + 1 Byte.

Quanto aos dados enviados pelo PC, eles são representados por:

- Referência de velocidade para o controlador do motor de tracção (em impulsos) - 1 Byte.
- Ângulo que se pretende transmitir às rodas dianteiras - 1 Byte.
- Ordem para os sensores IR de obstáculos serem lidos (ou não) - 1 Byte.
- Ordem para os sensores IR de parede serem lidos (ou não) - 1 Byte.

Capítulo 4

Localização e Navegação

4.1. Descrição Geral

Nesta dissertação, recorreu-se a um método baseado em hometria para medição relativa da posição do robô. No entanto esta deverá ser complementada com um sistema baseado em visão, sistema esse extremamente útil e necessário para a prova de condução autónoma do FNR como pôde retirar-se da leitura da secção 2.2.

Desta forma, através da hometria, que proporciona ao robô boa precisão a curto prazo, determina-se o seu estado actual relativamente à zona de pista que é percorrida, bem como às linhas da faixa de rodagem e passadeira.

Neste capítulo apresentam-se para além da localização relativa, o controlo e planeamento de trajectórias para o *FEUPCar*.

4.2. Hometria - Localização Relativa

A hometria é um dos métodos mais empregues na determinação a curtas distâncias da posição de um robô. Consiste na integração da informação incremental do movimento linear das rodas ao longo do tempo, de forma a medir o deslocamento efectuado. Sabendo então o estado anterior do robô e a distância percorrida no último intervalo de tempo, pode determinar-se a sua posição cartesiana e orientação, relativamente a um referencial global.

Na presente dissertação a informação incremental é facultada pelo encoder presente na plataforma robótica e acoplado ao veio da motor.

Apesar de tudo, a hometria tem uma grande limitação que se deve à acumulação de erro ao longo da trajectória. Tal facto provoca a divergência da posição calculada em relação ao valor correcto. Assim deve utilizar-se, periodicamente, um outro método que permita a correcção da posição (visão por exemplo).

A juntar a tudo isto, pode dizer-se que a hometria proporciona uma boa precisão em trajectórias pequenas e é económica de implementar.

4.2.1. Modelo Cinemático - Ackerman

O robô modelado está desenvolvido segundo uma geometria de *Ackerman*, onde uma transmissão é responsável por entregar o binário do motor até as rodas traseiras, estando a carga das rodas dianteiras a direcção. Veja-se a Figura 4.1.

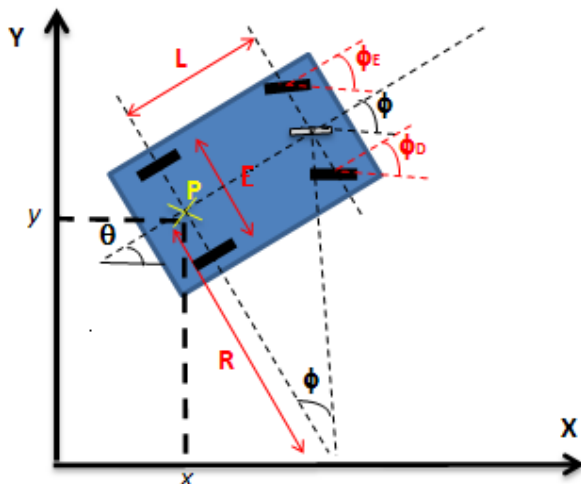


Figura 4.1 - Esquema cinemático do robô.

O veículo apresenta distâncias E entre as rodas e L entre os eixos dianteiro e traseiro. Considerando-se o referencial global XY , o ponto P caracteriza a posição (x, y) e a orientação θ do veículo. A orientação ϕ imposta ao sistema de direcção repercute-se, através da mecânica própria da actuação do tipo Ackerman, nos ângulos ϕ_E e ϕ_D das rodas dianteiras esquerda e direita respectivamente. R é o raio delineado pelo movimento do veículo.

O modelo hodométrico implementado é baseado na geometria de movimento do veículo e aplicando uma simplificação pelo princípio do modelo uniclo.

Assim os parâmetros decisivos para obtenção da hodometria são o ângulo de direcção (ϕ), a distância percorrida (Δd) e a distância entre eixos (L) [25].

Neste caso o modelo cinemático na forma discreta resulta do seguinte:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \Delta d_k \cdot \cos\left(\theta_k + \frac{\Delta\theta_k}{2}\right) \\ y_k + \Delta d_k \cdot \sin\left(\theta_k + \frac{\Delta\theta_k}{2}\right) \\ \theta_k + \Delta\theta_k \end{bmatrix}, \quad (4.1)$$

em que a entrada do sistema é a distância percorrida num instante de tempo - Δd_k e o ângulo de direcção - ϕ_k :

$$u_k = \begin{bmatrix} \Delta d_k \\ \phi_k \end{bmatrix} \quad (4.2)$$

e a relação entre o ângulo de direcção de direcção - ϕ_k e o desvio do ângulo realizado com o referencial num determinado intervalo de tempo - $\Delta\theta$ é:

$$\Delta\theta_k = \frac{\Delta d_k}{L} \cdot \tan(\phi_k) \quad (4.3)$$

A hodometria do robô é obtida, como já referido, através da integração da informação do encoder em cada período de amostragem e permite estimar com base no estado actual, o

deslocamento do robô. Assim torna-se necessário relacionar o deslocamento das rodas com os impulsos decorrentes do movimento do motor.

O deslocamento - Δd_k - é dado por:

$$\Delta d_k = \frac{NI_k}{C_{i/m}}, \quad (4.4)$$

onde NI_k representa o número de impulsos do encoder num determinado intervalo de tempo e $C_{i/m}$ representa o factor de conversão de impulsos para distância percorrida pelo robô.

Este último pode ser obtido da seguinte forma:

$$C_{i/m} = \frac{n \cdot R_e}{\pi \cdot D}, \quad (4.5)$$

em que D é o diâmetro das rodas do robô, n a relação da caixa redutora e R_e a resolução do encoder utilizado.

Já para a obtenção do ângulo de direcção, necessário para o cálculo da posição, o que foi realizado foi utilizar-se o ângulo inculido ao motor servo da direcção.

Expostas que estão as equações que determinam o modelo cinemático do robô em estudo, haverá agora que calibrar da melhor forma a hodometria.

4.2.2. Calibração da hodometria

A calibração da hodometria é um processo importante quando se pretende um resultado fiável das equações enumeradas na secção 4.2.1. Isto porque geralmente ocorrem erros - erros esses, sistemáticos - que podem ter origem nas medidas do diâmetro das rodas ou da distância entre eixos (L), no alinhamento das rodas ou até pela alteração da carga do robô, que poderá fazer variar o diâmetro das rodas.

A incorrecta calibração vai causar erros de grande nível devido ao erro acumulativo, decorrente da sua integração em cada período de amostragem.

O veículo sofreu um aumento do seu peso inicial pelo facto de lhe ser colocado um *PC* portátil. Assim os valores da distância entre eixos (L) e do diâmetro das rodas, que à partida já poderiam não ser os exactos, sofreram alterações. Neste caso, é necessário afinar estes dois parâmetros.

Uma vez que a distância entre eixos (L) não influencia o cálculo da hodometria para movimentos em linha recta (ângulo direcção igual a 0°), começou-se por ajustar o diâmetro das rodas. Consideram-se as duas rodas com o mesmo diâmetro, até pelo facto de não se tratar de uma tracção diferencial.

Então colocou-se o robô a realizar um movimento em linha recta para diferentes distâncias e armazenou-se a quantidade de impulsos medidos pelo encoder.

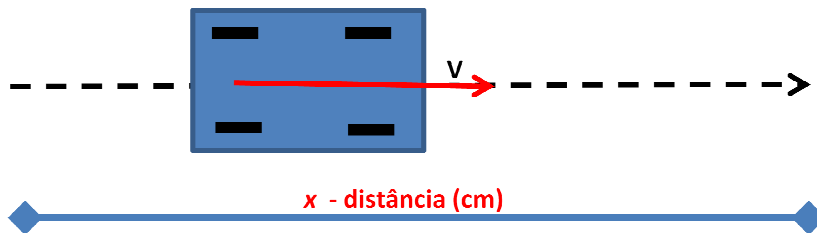


Figura 4.2 - Movimento em linha recta do robô.

Conclui-se assim que o robô efectuava 18.01 impulsos por cada centímetro, o que pela equação (4.5) equivale a dizer que o valor do diâmetro das rodas (D) é de 10.61cm.

Calibrado o diâmetro das rodas pode proceder-se à calibração da distância entre eixos.

Para tal colocou-se o robô a realizar movimentos com vários ângulos de direcção e registou-se a quantidade de impulsos obtidos, o ângulo efectuado com a horizontal e o ângulo de direcção aplicado ao veículo.

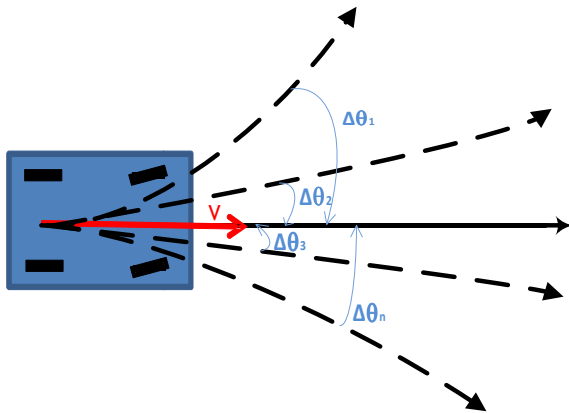


Figura 4.3 - Movimentos do robô para vários ângulos de direcção.

Como resultado deste processo, e aplicando as equações (4.3) e (4.4), obteve-se o valor de 35.51cm para a distância entre eixos (L).

Tabela 4.1 - Parâmetros, após calibração, úteis no cálculo odométrico, do FEUPCar.

Diâmetro das rodas - D	10.61 cm
Distância entre eixos - L	35.51 cm

Mesmo desta forma as calibrações descritas podem não garantir com elevada precisão a determinação dos parâmetros do sistema de hodometria, pois envolvem a aplicação de um sensor (encoder), e porque podem haver erros na avaliação dos ângulos ou distâncias.

Capítulo 5

Plataforma de Simulação

Neste capítulo é realçada a mais-valia da utilização de uma ferramenta que possibilite a realização de simulações do sistema em desenvolvimento e aperfeiçoamento.

Com a plataforma de simulação, pode evitar-se danos no veículo, bem como nos sensores e ainda nos possibilita a experimentação das diferentes estratégias de controlo e decisão a aplicar.

Assim nesta dissertação utilizou-se uma plataforma de simulação que dá pelo nome de SimTwo [26], e foi desenvolvida por um dos professores da FEUP, o Professor Paulo Costa.

O simulador referido, contempla já uma pista de condução autónoma, semelhante à do Festival Nacional de Robótica, e permite verificar os resultados em tempo real através de uma interface gráfica 3D. Pode, através desta, verificar-se o movimento do robô, bem como observar-se como evoluem os parâmetros da hodometria relativamente ao referencial global e efectuar-se a correcção dos algoritmos utilizados para o controlo.

5.1. Descrição Geral do *SimTwo*

A plataforma de SimTwo apresenta-se como um sistema de simulação realista, onde se podem implementar vários tipos de robôs¹¹ [26] e onde são utilizadas bibliotecas “*OpenSource*”. O seu realismo é conseguido pela decomposição do robô num sistema de corpos rígidos e motores eléctricos. Os corpos têm propriedades físicas como a sua forma, massa e momentos de inércia, atritos das superfícies e elasticidade. Existem também articulações típicas dos corpos/objectos reais como os eixos e calhas que podem ser associadas a sistemas de accionamento de sensores [26] e que unem os corpos rígidos.

¹¹ Robôs móveis com diferentes tipos de configurações (diferenciais, omnidireccionais, geometria de Ackerman, etc.), manipuladores robóticos, quadrúpedes, humanóides, qualquer tipo de robô terrestre que possa ser descrito com uma mistura de articulações rotativas e rodas clássicas/omnidireccionais, ou até veículos mais leves que o ar com ou sem hélices para propulsão.

Recorre-se para isto à biblioteca “*OpenSource*” *ODE* - Motor de simulação de corpos rígidos. Já a visualização 3D fica a cargo da biblioteca *GLScene*.

O sistema faculta um motor DC, com ou sem caixa redutora, com um encoder óptico e um controlador que poderá ser um PID em controlo de posição ou velocidade ou mesmo uma realimentação de estado. O seu modelo (motor) contém vários parâmetros presentes em qualquer motor real como saturação da tensão aplicada, limite de corrente e atrito de *Coulomb* [26].

O SimTwo oferece dois níveis de controlo do robô. Um sistema de baixo nível responsável pelo controlo dos motores, referido nos parágrafos anteriores, que por defeito, é invocado de forma periódica a cada 10ms, e um controlo de mais alto nível onde o utilizador pode explicar todo o seus algoritmos mais complexos e que, por seu lado, é invocado a cada 40ms.

Este controlador pode ser implementado recorrendo à biblioteca *Pascal Script* da própria plataforma de simulação onde se pode editar e compilar ou então através de um programa remoto que comunica via rede ou porta série [26].

5.1.1. Apresentação do Ambiente de desenvolvimento *SimTwo*

Descrito o SimTwo de uma forma geral, irá ser aclarado o ambiente de desenvolvimento que um comum utilizador pode encontrar.

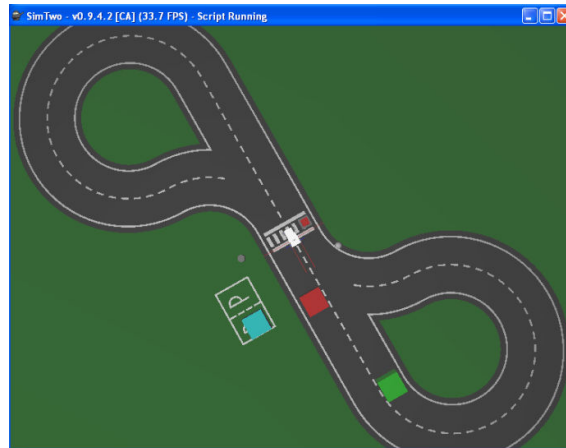


Figura 5.1 - Janela de Ambiente 3D do *SimTwo*.

Na Figura 5.1 - Janela de Ambiente 3D do SimTwo. Pode ver-se a interface gráfica de três dimensões, em que se poderá identificar a pista de Condução Autônoma, idêntica à da prova homónima do FNR, com obstáculos, passadeiras e outros elementos.

Toda esta interface poderá ser modelada a partir da janela de edição ilustrada na Figura 5.2.

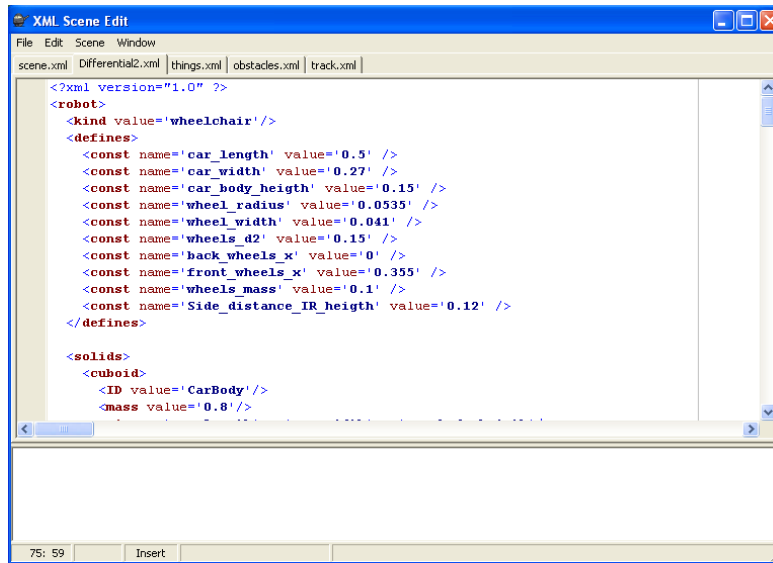


Figura 5.2 - Janela de edição dos robôs em *SimTwo*.

Para tal, este simulador recorre a ficheiros com sintaxe XML editáveis.

Ao iniciar a aplicação é reproduzido o ficheiro *scene.xml*, onde se encontram definidos os ficheiros XML que contemplam a configuração de cinco classes: cena, onde se inserem as classes que se deseja obter no ambiente de simulação, robô, obstáculos, objectos e pista.

O simulador detém também uma janela que permite a alteração dos parâmetros 3D da interface gráfica, configuração do tempo de ciclo, frames/segundo, entre outras coisas.

A janela discriminada apresenta-se à esquerda na figura seguinte.

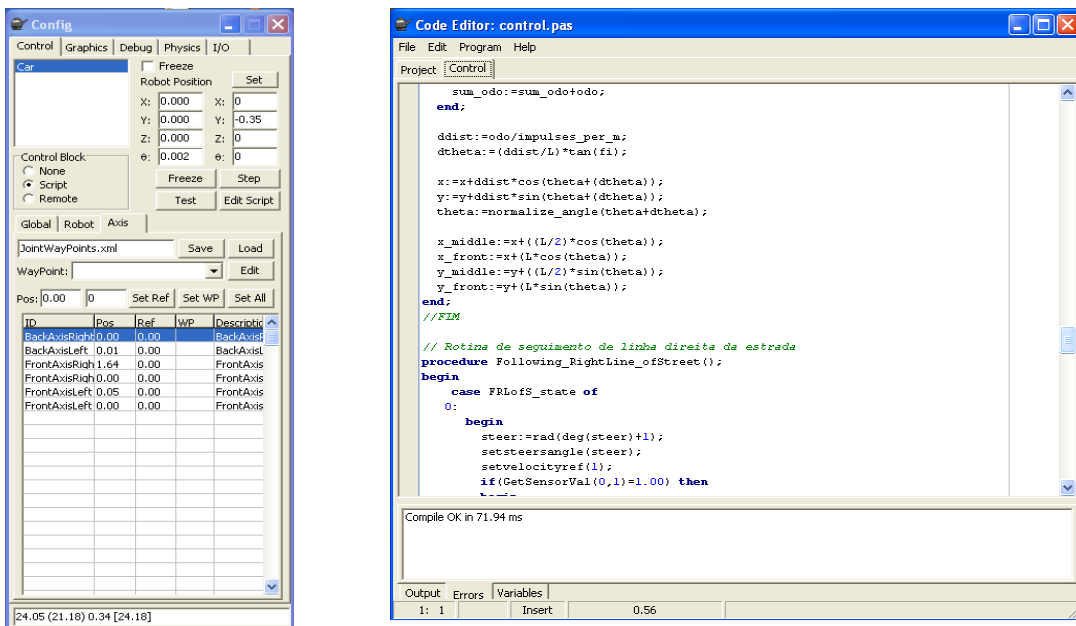


Figura 5.3 - Janela de configuração (à esquerda) e de programação em Pascal (à direita) existentes no *SimTwo*.

Presente na mesma figura, mas do lado direito, pode visualizar-se a janela *Code Editor* que permite a execução de código escrito em *ObjectPascal*¹² responsável pelo controlo dos algoritmos do robô.

Por fim apresenta-se a janela de gráficos - Figura 5.4, onde pode registar-se a evolução de algumas variáveis em função do tempo, tanto em modo *online* com em modo *offline*.

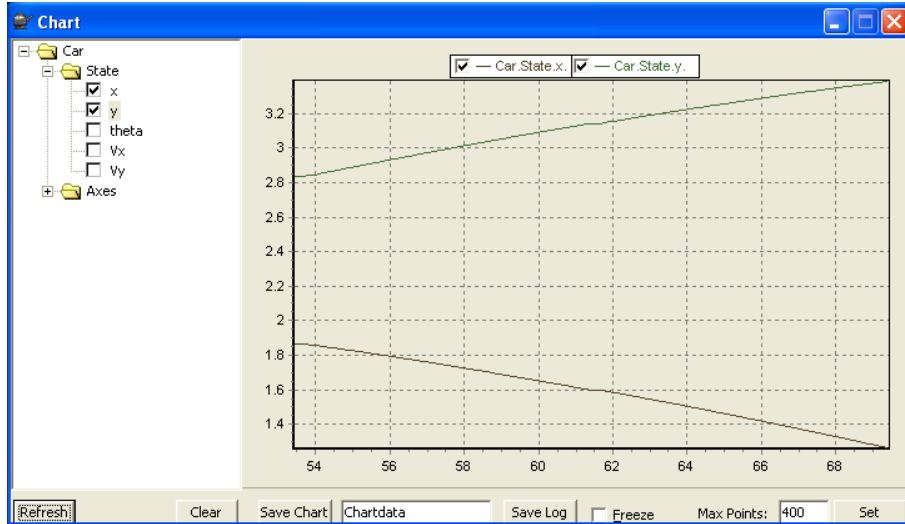


Figura 5.4 - Janela de gráficos de evolução do *SimTwo*.

5.2. Modelação do *FEUPCar*

A utilização do simulador *SimTwo* deve-se, como já referido, ao teste do veículo e seus algoritmos na pista de condução autónoma. Assim, e como já foi preparada e dimensionada toda a plataforma *FEUPCar* para uma futura de navegação controlada na prova referida, houve o cuidado de modelar a mesma plataforma bem como a sua pista para que se pudesse desenvolver todo o controlo e este pudesse ser experimentado/testado em simulação.

A pista, que irá ser encontrada na prova, já tinha sido modelada pelo autor do simulador, o que já foi uma vantagem. Assim fica a faltar a modelação do veículo.

A configuração do robô no *SimTwo* é realizada por intermédio da janela da Figura 5.2 alterando portanto o ficheiro com formato *XML* da classe robô. Inerente a esta classe irá ter que constar todas as subclasses que se desejam fornecer ao robô (sensores, corpo, etc.). Como o simulador continha já um veículo com a configuração geométrica de Ackerman, a mesma do *FEUPCar*, houve apenas que passar-se à colocação dos sensores nos seus devidos locais, dimensionar o veículo para as mesmas medidas características, bem como dotar o motor e encoder das mesmas propriedades aplicadas no robô real.

¹² Variante da linguagem *Pascal*.

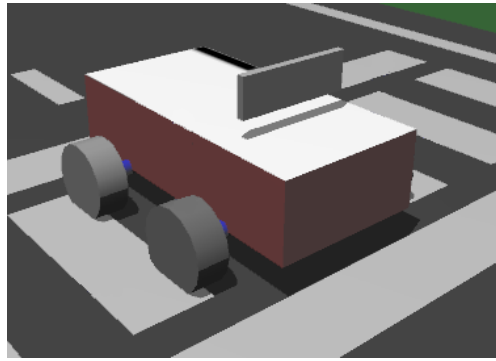


Figura 5.5 - Imagem do veículo original do simulador.

As subclasses utilizadas no SimTwo para a construção do veículo são as seguintes [26]:

- *Solids* - Representam um elemento físico com uma dada forma (Paralelepípedo, Esfera, Cilindro) ao qual está associada uma massa;
- *Shells* - Representam as superfícies de contacto para o modelo de colisões;
- *Articulations* - Responsável por todo o sistema de movimentação, desde os motores até às rodas;
- *Sensors* - Encarregue de introduzir sensores no robô.

O corpo do veículo é desenhado com um paralelepípedo da subclasse *solids* com as dimensões (largura, comprimento e altura) do corpo do veículo real, sendo que é definida a massa do mesmo, ajustaram-se então as dimensões e a massa do veículo.

Quanto às rodas, foram desenhadas com cilindros (subclasse *solids*), e modeladas com base em pressupostos previamente adquiridos (raio e largura das rodas).

A junção de rodas e corpo foi realizada através de eixos da subclasse *articulations* e dotaram o veículo de uma geometria de Ackerman e permitindo que o mesmo se pudesse movimentar.

Os eixos enumerados no parágrafo anterior contêm uma importante propriedade - o sistema de actuação que engloba os motores, com as respectivas caixas redutoras, *encoders* e controladores. Assim, os parâmetros do motor (*etiqueta motor*), da relação de transmissão (*etiqueta ratio*) e a resolução do *encoder* (*etiqueta ppr*) são estabelecidos conforme os dados dos *datasheets* dos fabricantes e dos previamente retirados. Já os dados do controlador (*etiqueta controller*) foram baseados nos valores obtidos na secção 3.3.2.1.

Para além disto o veículo continha uma superfície de contacto que dá ao veículo, uma aerodinâmica mais parecida com a realidade, visto ele não ser exactamente um paralelepípedo. Pode ver-se a superfície de contacto na Figura 5.5, representada por um rectângulo perpendicular ao corpo do veículo e acima do mesmo.

A detecção das linhas bem como das paredes e obstáculos é realizada através de sensores, que são introduzidos pela subclasse *sensors*.

Interiormente a esta subclasse existe ainda a classe de sensores de linhas - *floorline* - e a de sensores de distância - *IRSharp*.

Deste modo haverá apenas que alterar o valor dos comprimentos dos feixes (*etiqueta beam*) e definir a sua orientação e posicionamento no veículo (*etiquetas rot_deg* e *pos* respectivamente), para que se obtenha a modelação total do veículo.

Para que se perceba melhor o que são as etiquetas, mostra-se um exemplo de código de um sensor implementado:

```

<IRSharp>
  <absolute_coords/>
  <ID value='SideLeftDistance_IR'/>
  <beam length='0.8' initial_width='0.01' final_width='0.01'/>
  <pos x='car_length-0.25' y='car_width/2' z='Side_distance_IR_height'/>
  <rot_deg x='0' y='0' z='90'/>
  <color_rgb r='200' g='0' b='0'/>
</IRSharp>

```

O aspecto final pode ser visto na Figura 5.6.

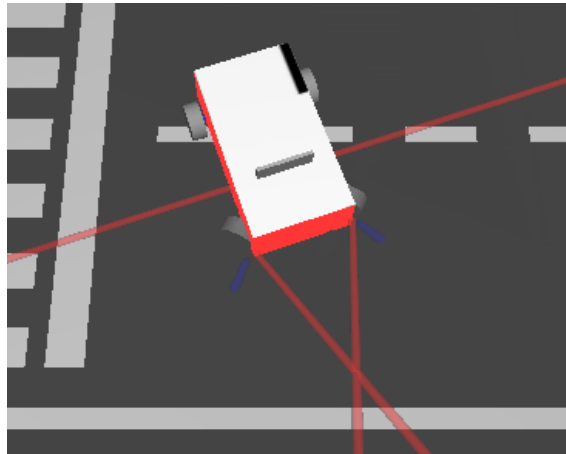


Figura 5.6 - Aspecto final da plataforma *FEUPCar* simulada.

Veja-se que os sensores de infra-vermelhos frontais - representados a vermelho - foram colocados de acordo com o referido em 3.2.6 para que se possam identificar objectos pequenos na frente do veículo.

Os sensores de detecção de linhas - representados a azul - permitem detectar linhas ainda antes de estar sobre elas, como previsto.

Capítulo 6

Conclusões e Trabalhos Futuros

6.1. Conclusões

O objectivo inicial deste projecto era conceber um sistema autónomo versátil e de baixo custo capaz de vir a participar em provas de robótica. Cumpriu-se com tais pressupostos, utilizando componentes, sensores, e até material existente no laboratório que representaria um baixo custo, como por exemplo os suportes para sensores e actuadores.

Este projecto abrange áreas de conhecimento bastante alargadas, como a mecânica, a electrónica e programação, podendo ser sempre alvo de melhoramentos e incorporação de novas funcionalidades a qualquer momento.

Foram estudados primeiramente os requisitos de necessidade do veículo na prova de condução autónoma do FNR o que contribuiu muito para o esclarecimento dos sensores e actuadores necessários para o efeito. Foi uma tarefa de investigação que se revelou fundamental.

Dotou-se o veículo do que necessitava para acompanhamento e detecção de situações limite que possam ocorrer no percurso, sendo que a câmara é vital para as questões principais, como a detecção de sinais e percurso da pista (correção da hodometria).

Pode dizer-se que os resultados, no que toca à sensorização, foram bastante satisfatórios.

Os sensores de detecção de linha permitem detectar a diferença entre as linhas brancas e a estrada, podendo mesmo detectar diferentes cores, através da regulação do potenciómetro do circuito.

Quanto aos sensores de detecção de obstáculos bem como das paredes do túnel, apesar de inicialmente terem muita variância, revelaram bons resultados com a aplicação do filtro de mediana, para distâncias compreendidas entre os 8 e os 60 cm, variando abruptamente para distâncias superiores. Atendendo às dimensões da pista não será necessário distâncias superiores às demais.

Planeou-se e implantou-se a ideia da colocação dos sensores de parede em posição estratégica de modo a que se desdobrasse em nova tarefa, que será a detecção das fitas da zona de obras. Mais uma opção para um futuro controlo.

Para estimação da posição, orientação e velocidade estudou-se uma técnica de calibração da hodometria que permitisse uma correcção eficiente dos erros sistemáticos. Neste caso os parâmetros que poderiam afectar os resultados são a distância entre eixos e o diâmetro das rodas e foram afinados, resultando melhores resultados. Apesar disso a hodometria acumula outro tipo de erros, aleatórios, que poderão afectar a estimação.

Concluiu-se que o robô acumulava erros decorrentes do arranque, ou seja, se o arranque não fosse expedito o suficiente, o robô teria um período de tempo de arranque em que contabilizava movimento sem que isso se detectasse. Deve ser evitado que tal aconteça num futuro controlo do mesmo.

Assim um outro método de correcção é sempre necessário. Terá que recorrer-se à câmara ou a pontos detectáveis pelos sensores.

Após implementação e teste, conclui-se também que o controlador *PI* de velocidades teve bons resultados, como se comprova pela Figura 3.25.

O controlo da direcção, através de um motor servo determinou-se uma tarefa bastante acessível, provando que estes motores são muito úteis em tarefas que envolvam o controlo de posição.

Uma interface, onde se podem ver todos os dados relativos aos sensores e actuadores foi criada, possibilitando assim que futuramente no desenvolvimento do controlo e decisão possa haver um apoio.

Um protocolo de comunicação foi implementado fornecendo uma comunicação adequada, sem erros ordenada e suportando falhas de comunicação momentâneas.

Foi modelado o *FEUPCar* numa plataforma de simulação que permite simular o comportamento do robô e aperfeiçoar os algoritmos de decisão e controlo que regem o mecanismo deste sistema, permitindo também detectar de uma forma mais transparente possíveis erros introduzidos durante a implementação do software.

Resumindo a plataforma está apta para no futuro - continuação do projecto - ser dotada de algoritmos “inteligentes” que ira interagir com a arquitectura distribuída de interface com todo o hardware.

6.2. Trabalhos Futuros

Existe ainda muito para implementar para que o robô esteja preparado para a prova de condução autónoma. Construída que está a plataforma, e com a consciência de que existe sempre algo para melhorar e o tempo é limitado, vários testes deverão ser afincadamente realizados.

No que aos sensores de distância diz respeito, deverão ser realizados um sem número de medições espaçadas do mínimo de distância possível para que a curva que determina as distâncias reais seja o mais próximo da realidade possível. Seria útil limitar as distâncias

medidas erradamente, ou seja, as medidas para além da faixa de medidas aceitável (8 aos 60cm), devendo essa decisão ser tomada em função da variância da estimativa.

Seria também muito útil efectuar um maior número de medições para a calibração de hometria.

Quanto ao driver do motor, poderia completar-se com um dispositivo que conseguisse inverter a tensão aplicada para o dotar de movimentos para ambos os sentidos. Completando ainda o robô com um *encoder* de duas fases.

Propõe-se também, como trabalho futuro, desenvolver algoritmos de visão capazes de detectar sinais de trânsito, a via de trânsito, bem como de todos os componentes que poderão surgir no desenrolar da prova de Condução Autónoma.

É também proposto a estimação do estado do robô através da fusão da hometria e da informação fornecida pelos algoritmos de visão com recurso a um filtro de *Kalman*.

Por fim pode proceder-se à substituição do computador portátil por outra plataforma computacional mais económica.

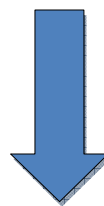
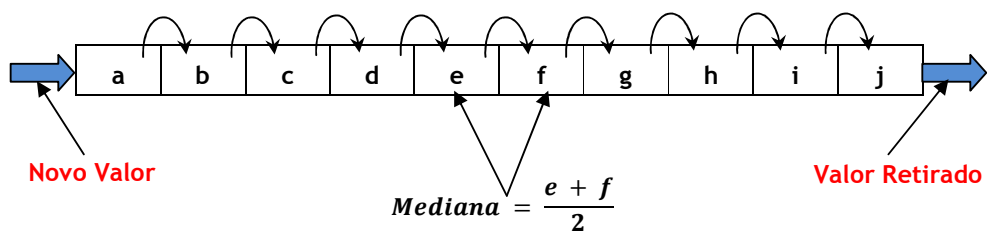
23	1.246	1.246	1.246	1.246	1.246	1.246	1.246	1.246	1.246	1.246	1.246	1.246
25	1.173	1.173	1.173	1.173	1.173	1.173	1.173	1.173	1.173	1.173	1.173	1.173
28	1.011	1.094	1.011	1.011	1.011	1.011	1.011	1.094	1.011	1.011	1.011	1.0276
30	0.972	0.972	0.972	0.972	0.972	0.972	0.972	0.972	0.972	0.972	0.972	0.972
33	0.938	0.938	0.938	0.938	0.938	0.938	0.938	0.938	0.953	0.938	0.938	0.9395
35	0.86	0.86	0.86	0.938	0.938	0.86	0.86	0.86	0.86	0.86	0.86	0.8756
38	0.86	0.0777	0.777	0.938	0.938	0.777	0.777	0.777	0.938	0.777	0.777	0.76367
40	0.62	0.777	0.777	0.62	0.777	0.762	0.762	0.762	0.762	0.762	0.762	0.7381
43	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
45	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
50	0.606	0.586	0.606	0.606	0.606	0.586	0.606	0.606	0.606	0.606	0.606	0.602
53	0.469	0.469	0.469	0.469	0.469	0.469	0.469	0.469	0.469	0.469	0.469	0.469
55	0.586	0.586	0.586	0.586	0.586	0.586	0.586	0.586	0.586	0.586	0.547	0.5821
57	0.547	0.547	0.547	0.547	0.586	0.547	0.547	0.586	0.547	0.547	0.547	0.5548
60	0.547	0.547	0.547	0.547	0.547	0.547	0.547	0.547	0.547	0.547	0.547	0.547
63	0.547	0.586	0.606	0.547	0.547	0.547	0.547	0.586	0.547	0.606	0.606	0.5666
65	0.469	0.474	0.474	0.469	0.474	0.474	0.469	0.474	0.474	0.474	0.469	0.472
68	0.469	0.469	0.469	0.469	0.547	0.469	0.469	0.469	0.469	0.469	0.469	0.4768
73	0.469	0.469	0.469	0.469	0.469	0.469	0.469	0.469	0.469	0.43	0.469	0.4651
75	0.307	0.307	0.307	0.307	0.347	0.347	0.386	0.386	0.386	0.386	0.347	0.3427
80	0.307	0.307	0.307	0.347	0.307	0.307	0.307	0.307	0.307	0.307	0.307	0.311
83	0.307	0.307	0.307	0.307	0.307	0.307	0.307	0.307	0.307	0.307	0.307	0.307
85	0.307	0.307	0.307	0.307	0.307	0.307	0.307	0.307	0.307	0.307	0.307	0.307
87	0.293	0.307	0.307	0.307	0.293	0.273	0.307	0.307	0.307	0.273	0.273	0.2974
89	0.273	0.273	0.273	0.303	0.273	0.234	0.234	0.273	0.273	0.273	0.307	0.2716
90	0.307	0.307	0.307	0.307	0.307	0.307	0.307	0.307	0.307	0.307	0.307	0.307
95	0.303	0.307	0.307	0.307	0.307	0.307	0.307	0.293	0.307	0.307	0.293	0.3038

100	0.234	0.234	0.151	0.19	0.19	0.234	0.19	0.273	0.273	0.19	0.2159
103	0.234	0.0014	0.136	0.034	0.034	0.136	0.034	0.151	0.136	0.136	0.10324
107	0.004	0.073	0.117	0.014	0.151	0.034	0.034	0.034	0.034	0.034	0.0529
110	0.073	0.014	0.151	0.034	0.014	0.014	0.073	0.034	0.073	0.073	0.0553
113	0.034	0.034	0.034	0.034	0.136	0.034	0.034	0.034	0.234	0.014	0.0622
116	0.136	0.014	0.073	0.034	0.014	0.014	0.014	0.034	0.014	0.004	0.0351
118	0.073	0.0151	0.014	0.014	0.014	0.004	0.117	0.034	0.004	0.014	0.03031
122	0.034	0.014	0.004	0.073	0.073	0.151	0.014	0.014	0.034	0.004	0.0415
125	0.004	0.004	0.034	0.034	0.034	0.004	0.004	0.034	0.004	0.136	0.0292

Anexo B

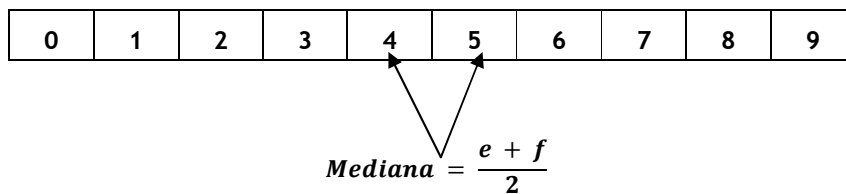
Filtro de Mediana

- Algoritmo:



Ordenar Vector:
→ Algoritmo Bubble Sort

Vector Ordenado



- **Código Implementado:**

```
function MedianFilter (size:Integer; vector: real_array):Real;
var
  i, j:Integer;
  temp: Real;
begin
  for i:=1 to size-1 do
    begin
      for j:=i+1 to size do
        begin
          if (vector[j]<vector[i]) then
            begin
              temp:=vector[i];
              vector[i]:=vector[j];
              vector[j]:=temp;
            end;
          end;
        end;
      result:=(vector[5]+vector[6])/2;
    end;
```

Referências

- [1] História da Condução Autónoma - *State Master*, Enciclopédia. Disponível em <http://www.statemaster.com/encyclopedia/Driverless-car>. Acesso em 19/Abril/2010.
- [2] Acerca da organização DARPA. Disponível em <http://www.darpa.mil/about.html>. Acesso em 20/Abril/2010.
- [3] Festival Nacional de Robótica 2010. Disponível em <http://robotica2010.ipleiria.pt/robotica2010/index.php?lang=pt>. Acesso em 19/Abril/2010.
- [4] Sobre iniciativa Robocup. Disponível em <http://www.robocup.org/>. Acesso em 19/Abril/2010.
- [5] Robótica 20XX. Disponível em <http://www.spr.ua.pt/fnr/v7/mainFrame.html>. Acesso em 28/Abril/2010.
- [6] DARPA Grand Challenge. Disponível em <http://www.statemaster.com/encyclopedia/DARPA-Grand-Challenge>. Acesso em 24/Abril/2010.
- [7] História da Condução Autónoma. Disponível em http://wapedia.mobi/en/Driverless_car. Acesso em 24/Abril/2010.
- [8] DARPA Grand Challenge. Disponível em http://wapedia.mobi/en/DARPA_Grand_Challenge. Acesso em 24/Abril/2010.
- [9] *Unmanned Vehicles Come of Age: The DARPA Grand Challenge*. Disponível em www.computer.org/comp/mags/co/2006/12/rz026.pdf. Acesso em 24/Abril/2010.
- [10] Sobre Laser Radar (LADAR). Disponível em <http://defense-update.com/products/l/ladar.htm>. Acesso em 30/Abril/2010.
- [11] O que é a paralaxe. Disponível em <http://docs.kde.org/stable/pt/kdeedu/kstars/ai-parallax.html>. Acesso em 30/Abril/2010.
- [12] Sobre o GPS. Disponível em <http://www.cienciaviva.pt/latlong/anterior/gps.asp>. Acesso em 30/Abril/2010.
- [13] Carvalhosa, A. M. F., Leite, T. L. B. (2006). *Versa Robot: Robot móvel versátil para competições em provas de robótica*.
- [14] Valente, V. (2007). *Rota 2007: Um carrinho sem condutor*.
- [15] Carvalho, J. N. S. (2008). *Rota 2008: Um Robô para Condução Autónoma*.
- [16] Alves, L. F. R. (2008). *ISRobot: Desenvolvimento e Integração de Sistemas e Módulos para Condução Autónoma*.
- [17] ELROB - *The European Robot Trial Website*. Disponível em <http://www.elrob.org/>. Acesso em 30/Abril/2010.
- [18] LRP *Electronic* - Sobre a bateria utilizada. Disponível em <http://67.199.85.166/lrpelectronic/lrphome.php>. Acesso em 19/Junho/2010.

- [19]Vaccaro, R. (1995) Digital Control: A state-space approach. McGraw-Hill College.
- [20]SHARP GP2D12 Datasheet. Disponível em <http://www.acroname.com/robotics/parts/SharpGP2D12-15.pdf>. Acesso em 19/Junho/2010.
- [21]BPW96 Datasheet - *Photo Detectors* - Vishay. Disponível em <http://www.vishay.com/docs/81532/bpw96.pdf>. Acesso em 19/Junho/2010.
- [22]TSAL6100 Datasheet - *Infrared emitters* - Vishay. Disponível em <http://www.vishay.com/docs/81009/tsal6100.pdf>. Acesso em 19/Junho/2010.
- [23]FDP3632 Datasheet - *100V N-Channel MOSFET*- FairChild. Disponível em <http://www.fairchildsemi.com/ds/FD/FDP3632.pdf>. Acesso em 19/Junho/2010.
- [24]UCC37324 Datasheet- *MOSFET Driver* - Texas Instruments. Disponível em <http://www.datasheetcatalog.org/datasheet/texasinstruments/ucc37324.pdf>. Acesso em 19/Junho/2010.
- [25]Mogensen, L.V., Andersen, N.A., Ravn, O. (2007). *Kalmtree - Navigation of Mobile Robots*. Disponível em http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/4883/pdf/imm4883.pdf. Acesso em 20/Junho/2010.
- [26]Costa, P. (2010). SimTwo. Disponível em <http://paginas.fe.up.pt/~paco/wiki/index.php?n=Main.SimTwo>. Acesso em 24/Junho/210.