

Faculdade de Engenharia da Universidade do Porto



FEUP

Telepresença - Robô Vigilante

Fillipe Lunardelli Ribeiro

VERSÃO FINAL

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

Orientador: Prof. Doutor Armando Luís Sousa Araújo

Julho de 2011

Resumo

A telepresença consiste num conjunto de tecnologias que permite ligar, em tempo real, pessoas ou objectos, que se encontram geograficamente separados. Actualmente começam a surgir cada vez mais robôs de telepresença que, controlados remotamente, através da Internet, podem ajudar na execução de várias tarefas à distância. Já foram testados robôs deste género como auxiliares em hospitais, permitindo que os médicos possam visitar doentes sem estarem presentes no hospital. Outras aplicações interessantes são os sistemas de vigilância e a prevenção de incêndios florestais.

Assim, o presente documento relata o projecto, desenvolvimento e teste de um robô de telepresença, com o objectivo deste efectuar missões de vigilância em todo o espaço exterior e interior da Faculdade.

O robô tem como plataforma de base um veículo que se auto-equilibra em duas rodas, tal como um Segway. Deste modo, o desafio do projecto passou pela modificação do veículo existente, de forma a permitir que o mesmo pudesse servir de plataforma móvel para o robô. Também foi necessária a instalação de toda a tecnologia necessária para permitir operar o robô à distância através da Internet, possibilitando visualização de imagem em tempo real.

Foi ainda desenvolvido um controlador *Fuzzy* para efectuar o controlo de velocidade do robô. O controlador foi inicialmente testado num sistema de simulação realista e tridimensional do robô e, por fim, implementado e testado no robô real.

O sistema provou ter uma estrutura que lhe confere bastante mobilidade, sendo capaz de se deslocar rapidamente em todos os espaços interiores e exteriores da Faculdade. As imagens captadas permitem o correcto controlo do robô à distância. Possibilitam também o contacto visual com pessoas, permitindo o seu uso como equipamento de vigilância para a Faculdade.

Abstract

Telepresence is a set of technologies which allows real time connection between people or objects that are geographically apart. Nowadays it starts increasingly to appear telerobots that, remotely controlled over Internet, can help in performing multiple tasks. These sorts of robots have already been tested in hospitals, as assistant's helpers, allowing doctors to see their patients without being present in the hospital. They also have an interesting application as surveillance systems and as prevention of forest fires.

So, this document reports the development of a telerobot with the purpose of making surveillance missions throughout the exterior and interior space of our university. The robot is based on a Segway, which is a mobile platform capable of balancing on two wheels. As so, the project challenge went through the modification of the existing Segway to allow it to be remotely controlled, as well as installing all the necessary technology to allow it to operate remotely over the Internet, enabling real-time image visualization.

A fuzzy controller was developed in order to perform robot speed control. The controller was initially tested in a realistic three-dimensional simulation of the robot and, finally, implemented and tested on the real robot.

The developed robot proved to have a structure that allows it fair mobility and is able to move quickly on all interior and exterior spaces of the university. The captured images allow proper distance control of the robot. It also grants visual contact with people, proving to be a good surveillance equipment for the university.

Agradecimentos

Ao meu orientador, Prof. Doutor Armando Luís Sousa Araújo, por ter acreditado em mim e no projecto proposto. Por nunca ter duvidado das minhas competências, mesmo quando as coisas não corriam pelo melhor, e por me ter dado liberdade total na execução deste projecto.

Um agradecimento especial aos meus colegas Bruno Moreira, Pedro Machado e Paulo Ferreira, por me terem ajudado a ultrapassar muitas das barreiras encontradas ao longo deste percurso.

Ao meu colega José Reis, por me ter emprestado várias vezes a sua câmara fotográfica e por me ter ajudado nas filmagens do robô.

Gostaria também de agradecer ao técnico da empresa Novatronica, Nuno Trocado, soldador experiente, por me ter auxiliado a soldar alguns componentes SMD mais complicados e necessários para a placa de controlo do robô.

Não podia deixar de agradecer também ao Prof. Doutor Paulo Portugal e ao Técnico Superior da Unidade de Redes de Comunicações da FEUP, Fernando Romão, pelo interesse demonstrado no projecto, pelas dicas e pela ajuda prestada nas comunicações wireless do robô.

Por fim, mas não menos importante, aos meus pais e ao meu irmão, que sempre me apoiaram, proporcionando-me todas as condições para que atingisse o sucesso.

Índice

Resumo	i
Abstract.....	iii
Agradecimentos	v
Índice.....	vii
Lista de Figuras	ix
Lista de Tabelas	xii
Abreviaturas	xiii
Capítulo 1	1
Introdução.....	1
1.1 - Enquadramento da dissertação.....	1
1.2 - Objectivo da dissertação	2
1.3 - Estratégia	2
1.4 - Organização do documento	3
Capítulo 2	5
Estado da Arte.....	5
2.1 - Introdução.....	5
2.2 - Robôs com auto-equilíbrio em duas rodas.....	5
2.2.1 - Rover: The Mobile Robotic Target System	7
2.2.2 - Octavia Robot	8
2.2.3 - Tbot: The Self-Balancing Transformer Robot.....	9
2.2.4 - PUMA - EN-V Project	10
2.3 - Robôs de Telepresença	12
2.3.1 - Rovio.....	12
2.3.2 - Texas Robot	13
2.3.3 - VGO	14
2.3.4 - Anybots QB/QA - Your Personal Avatar	14
2.4 - Sumário e Principais Conclusões	16
Capítulo 3	17

Hardware Utilizado	17
3.1 - Introdução	17
3.2 - Hardware Adquirido	17
3.2.1 - Veículo auto-equilibrado de duas rodas - <i>Segway</i>	18
3.2.2 - Motor de Passo e Calha Linear	19
3.2.3 - Controlador para Motor Passo a Passo	23
3.2.4 - PIC32 Ethernet Starter Kit	23
3.2.5 - Wireless Bridge	25
3.2.6 - Câmara IP	25
3.3 - Hardware Desenvolvido	26
3.3.1 - Placas de Circuito Impresso (PCBs)	27
3.3.2 - Estrutura Mecânica do Robô	37
3.4 - Sumário e Principais Conclusões	44
Capítulo 4	47
Adaptação de um Segway para uma Plataforma Robótica Móvel	47
4.1 - Introdução	47
4.2 - Simulação da dinâmica do robô recorrendo à plataforma SimTwo	48
4.3 - Desenvolvimento de um Controlador de Velocidade para o Segway e sua Implementação e teste no SimTwo	50
4.3.1 - Controlo responsável por manter o robô equilibrado em duas rodas (controlo do Segway)	51
4.3.2 - Controlo de velocidade do robô (simulação do comportamento de uma pessoa em cima do <i>Segway</i>)	53
4.4 - Implementação do Controlador de Velocidade no robô real	57
4.4.1 - Controlo do motor passo a passo	57
4.4.2 - Importação do controlador Fuzzy desenvolvido para o PIC32	59
4.4.3 - Implementação de sistema baseado em ultra-sons para evitar obstáculos	61
4.4.4 - Arquitectura de todo o código desenvolvido no PIC32	62
4.5 - Sumário e Principais Conclusões	64
Capítulo 5	65
Controlo Remoto do Robô recorrendo a Comunicações Wireless	65
5.1 - Introdução	65
5.2 - Comunicações Ethernet com o PIC32	65
5.3 - Comunicações Wireless	68
5.4 - Desenvolvimento de uma página Web para Controlo do robô e Visualização de Imagem	69
5.5 - Sumário e Principais Conclusões	72
Capítulo 6	73
Conclusões e Futuros Desenvolvimentos	73
6.1 - Conclusões gerais	73
6.2 - Futuros Desenvolvimentos	74
Referências	77

Lista de Figuras

Figura 2-1 - Segway PT i2 [4].	6
Figura 2-2 - Segway RMP 100 e RMP 200 [5].	7
Figura 2-3 - Rover: The Mobile Robotic Target System [9].	8
Figura 2-4 - Octavia Robot [10].	9
Figura 2-5 - Tbot - IHMC [11].	10
Figura 2-6 - PUMA - EN-V Project [12].	11
Figura 2-7 - Princípio de funcionamento do PUMA [12].	11
Figura 2-8 - Rovio [6].	13
Figura 2-9 - Texas Robot [13].	13
Figura 2-10 - VGO Telepresence Robot [14].	14
Figura 2-11 - QB Anybots Telepresence Robot [7].	15
Figura 2-12 - QA Anybots Telepresence Robot [16].	15
Figura 3-1 - Diagrama de ligação entre o Hardware Adquirido.	18
Figura 3-2 - Elecktor Wheelie Segway.	19
Figura 3-3 - Aceleração e Desaceleração do Segway com movimentação de uma massa M. ...	20
Figura 3-4 - DryLin ZLW-0630-Basic-02-300mm [18].	20
Figura 3-5 - Relação entre o Binário e a Carga para várias acelerações em movimento horizontal [18].	21
Figura 3-6 - Relação Binário/Velocidade do motor de passo escolhido [19].	22
Figura 3-7 - Motor de Passo NANOTEC ST4118D1804 [19].	22
Figura 3-8 - Controlador NANOTEC SMC42 [20].	23
Figura 3-9 - PIC32 Ethernet Starter Kit [21].	24

Figura 3-10 - PIC32 I/O Expansion Board e Conector Hirose FX10A-120S [21].	24
Figura 3-11 - Wireless Bridge D-Link DAP-1522 [22].	25
Figura 3-12 - Câmara IP ZAVIO F210A [23].	26
Figura 3-13 - Estrutura definida inicialmente para a construção do robô.	27
Figura 3-14 - PCB para Alimentação e dar acesso às I/Os do PIC32 Ethernet Starter Kit.	28
Figura 3-15 - Circuito subtractor.	29
Figura 3-16 - Circuito para o controlo da corrente do LM334.	30
Figura 3-17 - Excerto do esquema eléctrico do sistema de controlo original do <i>Segway</i> da Elektor Wheelie [17].	31
Figura 3-18 - PCB para aquisição e condicionamento de sinais do <i>Segway</i> (Ultiboard).	32
Figura 3-19 - Esquema eléctrico para condicionamento do sinal do acelerómetro (Multisim).	33
Figura 3-20 - Esquema eléctrico para condicionamento dos sinais que permitem determinar o nível da bateria e a velocidade do robô.	33
Figura 3-21 - Esquema eléctrico do circuito que permitiu dar potência aos sinais de saída do PIC32.	34
Figura 3-22 - Interior do <i>Segway</i> original.	35
Figura 3-23 - Ligação de cabos à placa original do <i>Segway</i> para obtenção de sinais: ângulo de inclinação, velocidade e nível da bateria.	35
Figura 3-24 - Primeira versão da placa que permite dar acesso aos IOs do PIC32.	36
Figura 3-25 - PIC32 Ethernet Starter Kit ligado à placa desenvolvida.	36
Figura 3-26 - Versão final das duas placas desenvolvidas - ligadas ao PIC32 e ao robô.	37
Figura 3-27 - Fase inicial da estrutura mecânica construída.	38
Figura 3-28 - A calha, o <i>router</i> e o controlador do motor, afixados ao acrílico.	38
Figura 3-29 - Fase final da estrutura mecânica construída (foto nº1).	39
Figura 3-30 - Fase final da estrutura mecânica construída (foto nº2).	39
Figura 3-31 - Motor de passo, acoplado à calha linear.	40
Figura 3-32 - Ultra-sons instalados na parte da frente do robô.	40
Figura 3-33 - <i>Veroboard</i> desenvolvida para Sensor de Ultra-sons.	41
Figura 3-34 - Câmara IP com lente de grande angular instalada (<i>wide angle lens</i>).	42
Figura 3-35 - Robô, num passeio pela FEUP.	42
Figura 3-36 - robô Vigilante.	43

Figura 4-1 - Plataforma SimTwo [24].	49
Figura 4-2 - Simulação do robô Vigilante.	50
Figura 4-3 - Regras Fuzzy para o controlo de equilíbrio do robô no SimTwo.....	51
Figura 4-4 - Superfície <i>Fuzzy</i> responsável pelo equilíbrio do robô na posição vertical.	52
Figura 4-5 - Regras Fuzzy para o controlo de velocidade do robô	54
Figura 4-6 - Superfície <i>Fuzzy</i> responsável pelo controlo de velocidade do robô.	55
Figura 4-7 - Funções de pertença das entradas e da saída do controlador <i>Fuzzy</i> responsável pelo controlo de velocidade do robô	55
Figura 4-8 - Ligações eléctricas do controlador do motor passo a passo SMC42.....	57
Figura 4-9 - Ultra-sons HC-SR04.	61
Figura 4-10 - Diagrama temporal de funcionamento do HC-SR04.	62
Figura 4-11 - Algoritmo de Controlo para o correcto funcionamento do robô Vigilante.	63
Figura 5-1 - Camadas da Stack TCP/IP desenvolvida pela Microchip para o PIC32 [26].	66
Figura 5-2 - Exemplo de utilização de variáveis dinâmicas.	67
Figura 5-3 - Pagina Web de controlo do robô Vigilante (http://segway/index.html).	70
Figura 5-4 - Página para calibração do ângulo e ajuste dos ganhos do controlador Fuzzy (http://segway/parametros.html).	71
Figura 5-5 - Programa da Microchip que permite converter as páginas Web em código C.	71

Lista de Tabelas

Tabela 3-1 - Registo dos valores de tensão do acelerómetro para os ângulos de interesse.	29
Tabela 3-2 - Material encomendado.....	44
Tabela 5-1 - Atribuição de valores às variáveis de controlo do robô Vigilante.	66

Abreviaturas

Lista de abreviaturas (ordenadas por ordem alfabética)

ADC	<i>Analog to Digital Converter.</i>
DARPA	<i>Defense Advanced Research Projects Agency.</i>
FEUP	Faculdade de Engenharia da Universidade do Porto.
GM	<i>General Motors.</i>
IHMC	<i>Institute for Human & Machine Cognition.</i>
PC	Computador.
PIC32	<i>Programmable Intelligent Computer</i> (Microcontrolador que funciona a 32 bits).
PWM	<i>Pulse With Modulation.</i>
SMD	<i>Surface Mounting Devices.</i> Um componente SMD é geralmente menor do que seu equivalente convencional.

Capítulo 1

Introdução

1.1 - Enquadramento da dissertação

No mundo em que vivemos actualmente, a tecnologia possibilita que possamos de alguma forma presenciar um local, mesmo não estando lá fisicamente. A Internet veio quebrar fronteiras, hoje podemos fazer compras, pesquisar, comunicar com família e amigos, conhecer pessoas novas, tudo sem sair de casa. Da ideia de que podemos interagir com outras pessoas à distância, surgiu o conceito da telepresença [1]. A telepresença consiste num conjunto de tecnologias que permite ligar em tempo real, pessoas ou objectos, que se encontram geograficamente separadas, através de ligações vídeo e áudio. Um sistema de Telepresença muito utilizado em empresas é a vídeo-conferência, permitindo uma fácil e rápida interacção entre membros da empresa presentes noutros países [2]. No entanto, tem vindo a surgir recentemente outro tipo de abordagem para a telepresença. Robôs controlados remotamente através da Internet, que permitem a interacção com pessoas através de sistemas de vídeo e áudio. Robôs como estes possibilitam interligações com o local envolvente, que nenhum sistema de vídeo-conferência, por mais bem montado que esteja, consegue oferecer [3].

Outra tecnologia bastante interessante, da actualidade, é o veículo com auto-equilíbrio em duas rodas, mais comumente chamado de Segway [4], devido ao facto de ter sido a marca Segway a lançar o conceito. É uma tecnologia cada vez mais utilizada em projectos com robôs móveis onde se pretende que haja alguma interacção do robô com pessoas, uma vez que estes robôs podem apresentar uma estatura e mobilidade semelhantes à de uma pessoa adulta. Sistemas baseados no Segway têm como principal vantagem, face a outros tipos de sistemas de locomoção móvel mais tradicionais, o facto de serem sistemas capazes de manter o equilíbrio mesmo sujeitos a grandes perturbações, factor este fundamental quando se está a falar de robôs com a estatura de uma pessoa.

1.2 - Objectivo da dissertação

A presente dissertação tem como finalidade o desenvolvimento de um robô de telepresença, com o objectivo deste, efectuar missões de vigilância em todo o espaço exterior e interior da Faculdade. O robô deverá ser controlado remotamente, recorrendo à rede de Internet sem fios disponível em toda a Faculdade e recolherá imagens em tempo real. A dissertação tem como base um veículo com auto-equilíbrio de duas rodas, já disponível na Faculdade, que terá de ser alterado de forma a permitir que este possa ser controlado remotamente. Numa primeira fase o controlo será realizado recorrendo a ligações por cabo, sendo no entanto o objectivo final dotá-lo da capacidade de controlo sem fios, com recurso à rede *Wireless* disponível na Faculdade. Para alcançar o objectivo final, a ordem de trabalhos definida para a presente dissertação foi a seguinte:

- ✓ Desenvolvimento de um controlador *Fuzzy* em Matlab, que permita efectuar o controlo de velocidade do *Segway*;
- ✓ Projecto, desenvolvimento e teste de um simulador da dinâmica do robô, recorrendo à plataforma SimTwo e implementação, em simulação, do controlador de velocidade desenvolvido;
- ✓ Implementação do controlador de velocidade desenvolvido, no robô real (recorrendo a ligação por cabo);
- ✓ Projecto e desenvolvimento de um sistema de controlo remoto utilizando a rede *Wireless* existente na Faculdade, permitindo visualização em tempo real de imagem e áudio;
- ✓ Projecto, desenvolvimento e teste de um sistema que permita ao robô evitar obstáculos.

1.3 - Estratégia

Uma vez que o robô a desenvolver, tem por base um *Segway* comum para transporte de pessoas, a estratégia escolhida para o controlo remoto do robô, visa o desenvolvimento de um sistema que permita controlar a velocidade do robô, modificando o seu centro de massa, simulando assim a suposta inclinação da pessoa em cima do equipamento. Esta abordagem permite a utilização do equipamento já existente, sem ser necessário a reprogramação do

controlador ou alteração do equipamento de um *Segway* vulgar, para coloca-lo a funcionar por controlo remoto.

1.4 - Organização do documento

Este documento está organizado em seis capítulos onde se descreve todo o trabalho realizado. Os seus conteúdos serão descritos, de forma sucinta, na presente secção. Todos os capítulos estão estruturados com uma introdução ao tema a apresentar e uma conclusão no final do capítulo.

O primeiro capítulo é o capítulo introdutório, que se destina a fazer o enquadramento da dissertação, bem como definir quais os objectivos pretendidos com a realização desta dissertação.

No capítulo dois é feito o estudo do estado da arte no que diz respeito a robôs que utilizam a tecnologia de auto-equilíbrio em duas rodas, assim como de robôs de Telepresença já existentes. Serão analisados os projectos mais relevantes da actualidade, envolvendo as duas tecnologias.

O capítulo três apresenta todo o material utilizado na elaboração da presente dissertação, demonstrando a análise feita aquando da escolha do material.

No capítulo quatro é feita a descrição do trabalho desenvolvido para colocar o robô a movimentar-se sob a base de um *Segway*, utilizando ligações por cabo, assim como o sistema desenvolvido que permite que este consiga evitar obstáculos.

O capítulo cinco pretende demonstrar o trabalho realizado para colocar o robô a ser controlado remotamente pela rede *wireless*, com a capacidade de visualização de imagem e áudio em tempo real.

Por fim, no capítulo seis, são apresentadas as conclusões da dissertação e sugeridos futuros desenvolvimentos nesta área.

Capítulo 2

Estado da Arte

2.1 - Introdução

Actualmente, existem muitos robôs que utilizam a tecnologia associada ao funcionamento de um Segway para a sua locomoção. O número de robôs com esta tecnologia cresceu principalmente com o aparecimento de uma nova gama de equipamentos, designada Segway® Robotic Mobility Platform (RMP) [5], desenvolvido pela Agência de Defesa Norte-Americana DARPA. O Segway RMP é uma plataforma móvel, que se auto-equilibra em duas rodas, servindo de base móvel para o desenvolvimento de robôs.

Existem também no mercado vários modelos de robôs de telepresença, que recorrem à rede Wireless para enviar ordens de comando e para obtenção de *feedback* por áudio e vídeo, alguns de pequena dimensão e mais acessíveis como o Rovio® [6] e outros de maior dimensão e mais sofisticados, que utilizam o mesmo conceito de locomoção em duas rodas do Segway, como é o caso do AnyBot® [7].

No presente capítulo será visto o estudo do estado da arte para o que se faz actualmente, quer com o desenvolvimento de robôs que se auto-equilibram em duas rodas, quer com o desenvolvimento de robôs de Telepresença controlados pela Internet.

2.2 - Robôs com auto-equilíbrio em duas rodas

O primeiro veículo de transporte humano com auto-equilíbrio em duas rodas desenvolvido foi o Segway® Personal Transporter (PT), inventado por Dean Kamen e lançado em 2001. A sua concepção revolucionária permite que o condutor se movimente com muita segurança para frente e para trás, com uma simples inclinação do corpo. O auto-equilíbrio é o aspecto mais surpreendente sobre o Segway PT e é também a chave para a sua operação. Para

compreender como o sistema funciona, basta ter em consideração o modelo que o inventor usou para o desenvolvimento do dispositivo: o corpo humano. Quando caminhamos, qualquer inclinação realizada pelo nosso corpo, é detectada pelo cérebro devido a uma alteração no fluido do ouvido interno, fazendo com que uma perna se movimente para frente, impedindo a queda. Se continuarmos a inclinar-nos para frente, o cérebro continuará a colocar uma perna à frente, mantendo-nos em pé. Ao invés de cairmos, andamos para a frente, um passo de cada vez. O Segway PT faz praticamente a mesma coisa, mas utiliza as rodas no lugar das pernas, um motor no lugar dos músculos, um grupo de microprocessadores no lugar do cérebro e um conjunto de sofisticados sensores de inclinação no lugar do sistema de equilíbrio do ouvido interno. Como o cérebro, o Segway sabe quando o condutor se inclina para frente. Para manter o equilíbrio e impedir a queda, movimenta os motores na velocidade certa fazendo assim o Segway andar [8]. Na Figura 2-1 pode ser visto um dos modelos mais recentes, o Segway PT i2.



Figura 2-1 - Segway PT i2 [4].

O Segway PT teve grande visibilidade e com o seu aparecimento muitos investigadores começaram a desenvolver os seus próprios veículos de duas rodas com auto-equilíbrio como o Segway. Actualmente sistemas de locomoção idênticos ao do Segway são muito utilizados em robôs, uma vez que permitem que robôs com grandes estaturas se tornem bastante estáveis, ágeis, rápidos e robustos, podendo circular facilmente tanto em espaços interiores como exteriores e em quase todos os tipos de terreno. O número de robôs com esta tecnologia cresceu principalmente com o aparecimento do Segway® RMP, desenvolvido pela Agência de Defesa Norte-Americana DARPA. O Segway RMP é uma plataforma móvel, que se auto-equilibra em duas rodas como um Segway PT, no entanto trata-se de um equipamento bem mais sofisticado, que contém também um sistema de controlo de velocidade e posição, podendo assim, servir de plataforma para o desenvolvimento de robôs móveis. Num Segway

PT é o ocupante do veículo que controla a posição do mesmo, através da inclinação do seu corpo, não sendo possível controlar directamente a sua velocidade, de forma a possibilitar o seu controlo remoto. O aparecimento do Segway RMP veio facilitar o desenvolvimento de robôs complexos, uma vez que toda a parte de locomoção pode ser comprada à parte. Existem actualmente vários modelos de Segways RMP. Na Figura 2-2 podem ser vistos os modelos RMP 100 e RMP 200, modelos já muito utilizados em projectos de robótica [5].



Figura 2-2 - Segway RMP 100 e RMP 200 [5].

De seguida serão analisados os principais projectos já desenvolvidos envolvendo robôs com auto-equilíbrio em duas rodas.

2.2.1 - Rover: The Mobile Robotic Target System

O corpo de fuzileiros navais da Austrália está a testar um novo conceito de treino para os atiradores furtivos, utilizando um robô desenvolvido pela *Marathon Robotics*. O Rover, mostrado na Figura 2-3, é um sistema de alvo móvel robótico concebido para executar um conjunto pré-programado de actividades, de forma a criar cenários para o treino dos fuzileiros. A introdução de comandos pré-programados elimina a necessidade de controlar individualmente cada robô, reduzindo o número de operadores necessários para conduzir um cenário.

Os robôs deslocam-se sob uma plataforma semelhante à de um Segway PT, mantendo o equilíbrio em duas rodas, o que lhes permite uma fácil e rápida locomoção em ambientes urbanos. O manequim colocado por cima da plataforma é inclinado para a frente e para trás para o robô se deslocar, da mesma forma que uma pessoa faria. A plataforma é blindada e o manequim é feito de um plástico resistente. Quando o robô é atingido, o manequim cai para trás, deslocando-se 90° em relação à plataforma, para dar uma representação visual de que o alvo foi atingido. O manequim pode ser depois restabelecido para a sua posição original automaticamente, com auxílio de um motor que desloca o manequim novamente para a posição vertical.

É fácil perceber a importância do uso de uma plataforma baseada num *Segway* neste tipo de aplicações. Permite a construção de robôs com a mesma estatura de um homem, permitindo simular o deslocamento humano e garantindo o seu equilíbrio. O auto-equilíbrio é no fundo, o ponto-chave para esta aplicação, uma vez que o uso de qualquer outro tipo de plataforma faria com que o robô tombasse ao ser atingido por um projectil [9].



Figura 2-3 - Rover: The Mobile Robotic Target System [9].

2.2.2 - Octavia Robot

Octavia é um robô social desenvolvido pela Marinha dos Estados Unidos com o intuito de melhorar as interações entre robôs e pessoas. Combina extrema mobilidade, capacidades verbais e expressões faciais. O que proporciona a mobilidade ao robô é uma plataforma do *Segway RPM 200*, que lhe confere extrema manobrabilidade, permitindo simultaneamente a construção de um robô bastante compacto (Figura 2-4). Pode-se realçar mais uma vez as vantagens de utilização de robôs sob a plataforma de um *Segway*, quando se pretende que os mesmos interajam com pessoas, como neste projecto [10].



Figura 2-4 - Octavia Robot [10].

2.2.3 - Tbot: The Self-Balancing Transformer Robot

O Tbot (Figura 2-5) é um robô desenvolvido pela IHMC (*Institute for Human & Machine Cognition*), instituto de desenvolvimento fundado pelo exército Norte-americano (DARPA) para testar diferentes tipos de robôs soldado, que possam vir a substituir no futuro os soldados Norte-americanos no Iraque e Afeganistão. Pretendia-se nesse projecto o desenvolvimento de um robô que se pudesse locomover com facilidade em ambiente urbano e que pudesse ser operado, tanto numa configuração de baixo perfil, para melhor se poder esconder, como numa configuração que permitisse ao robô transportar uma câmara ou uma arma, a uma altura aproximadamente ao nível dos olhos humanos. A solução implementada para satisfazer essas necessidades foi a construção de um robô que se equilibra na vertical em duas rodas, possuindo ainda “braços” com rodas, permitindo assim, que este possa comutar facilmente entre uma configuração em quatro rodas, de baixo perfil, e uma de duas rodas. A utilização de um robô que se equilibra em duas rodas foi considerada uma boa opção neste projecto, uma vez que os objectivos pretendidos eram muito difíceis de alcançar com um tradicional robô de quatro ou três rodas e por a construção de um robô bípede ser demasiado complexa.

O robô não tem nenhuma inteligência para além do sistema de controlo que o mantém equilibrado em duas rodas, sendo controlado remotamente por sinais de rádio. Para o seu

controlo foi utilizado um controlador PID inteligente e um Filtro de Kalman para o conseguir manter estabilizado numa determinada posição [11].

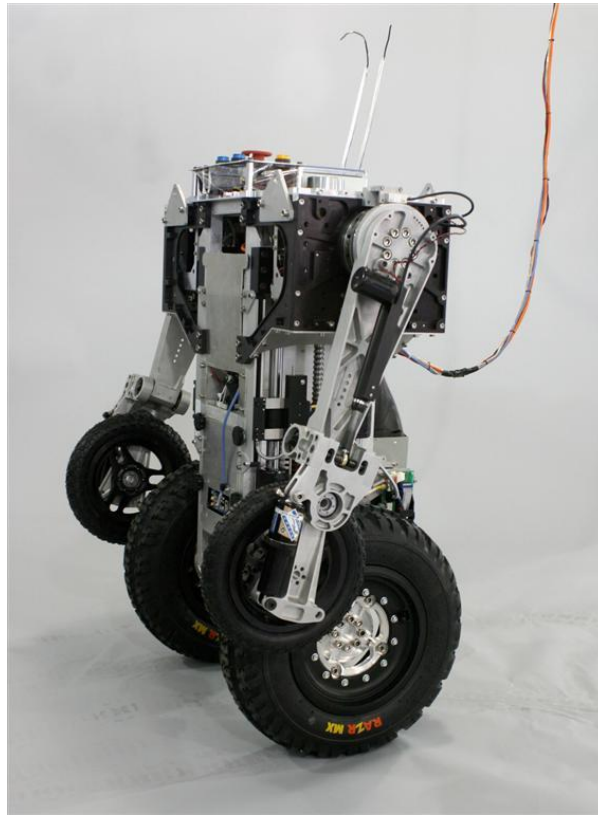


Figura 2-5 - Tbot - IHMC [11].

2.2.4 - PUMA - EN-V Project

PUMA é nome de um protótipo desenvolvido para um projecto denominado *EN-V*, que surgiu de uma parceria entre a Segway® e a General Motors (GM). Trata-se de um projecto futurista, que visa o desenvolvimento de um novo conceito de veículos de transporte urbano para duas pessoas, como substituto para os carros. O veículo baseia-se no funcionamento do Segway PT equilibrando-se em duas rodas, no entanto tem uma cabine coberta, com dois lugares sentados oferecendo o conforto de um carro. Um dos três protótipos criados pode ser visto na Figura 2-6.

O projecto EN-V estabelece a visão que em 2030 as pessoas irão utilizar este novo conceito de transporte. É um veículo que combate o congestionamento pelo seu tamanho reduzido, sendo também bastante ágil, eficiente e amigo do ambiente [12].



Figura 2-6 - PUMA - EN-V Project [12].

Apesar de o veículo ser semelhante a um Segway, e funcionar exactamente da mesma forma que um Segway PT, não é controlado da mesma maneira. O condutor possui um volante para controlar o veículo, já não sendo necessário inclinar-se para frente, ou para trás, para o movimentar, como acontece no Segway PT. No volante existe um sistema semelhante a um *joystick* que, accionado, movimenta um motor linear, existente na base do veículo, que desloca toda a carroçaria para frente e para trás consoante os comandos do condutor. Accionando a movimentação da carroçaria, o condutor consegue mudar o centro de massa do veículo e assim, mover o veículo para frente e para trás. Na Figura 2-7 pode ser visto o motor linear existente na base do veículo.

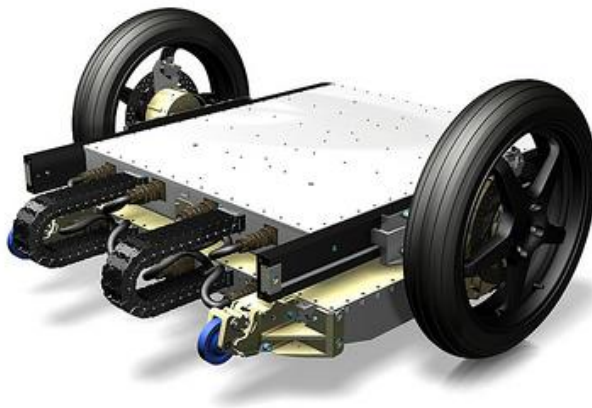


Figura 2-7 - Princípio de funcionamento do PUMA [12].

2.3 - Robôs de Telepresença

Os robôs de Telepresença são cada vez mais uma realidade hoje em dia. O actual crescimento da tecnologia *wireless* torna possível o controlo remoto deste tipo de robôs através da Internet, com a vantagem de ser possível aceder à rede, onde se encontra ligado o robô, em qualquer lugar do Mundo. Sistemas de Telepresença como a videoconferência, já são amplamente utilizados por empresas, possibilitando que pessoas de países ou cidades diferentes possam interagir em reuniões de trabalho sem a necessidade de se deslocarem fisicamente ao mesmo local. No entanto, a utilização de um robô de Telepresença permite um novo tipo de interacção com as pessoas e com o lugar envolvente, fazendo com que a pessoa que opera o robô se sinta realmente presente no local, dando-lhe mais liberdade, uma vez que pode se deslocar para onde quiser e conversar com quem entender, mantendo contacto visual com a pessoa em questão. Este novo conceito de Telepresença, não só se torna mais atractivo para as empresas como substituto dos actuais sistemas de Telepresença por videoconferência, como também abre um novo leque de aplicações para este tipo de sistemas. Robôs de Telepresença já foram desenvolvidos e testados como auxiliares em hospitais, permitindo que os médicos possam visitar doentes sem estarem presentes no hospital. Têm também uma aplicabilidade interessante como sistemas de vigilância, sistemas de diagnóstico de falhas em subestações isoladas, em prevenção de incêndios florestais e até como ajuda a crianças incapacitadas de irem à escola, que, através de um robô de Telepresença, podem assistir às aulas e conviver com os colegas a partir de casa.

De seguida serão analisados alguns robôs de Telepresença já existentes no mercado, assim como as tecnologias utilizadas em alguns projectos já desenvolvidos envolvendo controlo de robôs através da Internet.

2.3.1 - Rovio

O Rovio (Figura 2-8), desenvolvido pela WooWee, é dos robôs de Telepresença mais simples e acessíveis existentes no mercado. Trata-se de um robô pequeno e com extrema mobilidade, uma vez que se desloca sobre 3 rodas omnidireccionais. Pode ser controlado por praticamente qualquer equipamento com ligação à Internet, desde um PC até um simples telemóvel ou PDA. Tem uma câmara que pode ser deslocada para três posições distintas e um sistema de áudio bidireccional. O robô tem um sistema de localização baseado em sensores de infravermelho (IR). Sensores de IR também asseguram que o robô não embata em obstáculos. O utilizador pode também programar caminhos, gravando pontos de referência [6].



Figura 2-8 - Rovio [6].

2.3.2 - Texas Robot

Texas é um robô de Telepresença desenvolvido pela empresa de robótica Willow Garage. O projecto começou com um funcionário da empresa, Dallas Goecker residente em Indiana, que descontente com o facto de não poder estar presente no escritório principal, utilizou peças da reposição para criar um robô que lhe permitisse comunicar com os colegas do escritório em *Silicon Valley*. Trata-se de um projecto *open-source* e que já está a ser testado em algumas empresas.

O robô utiliza um PC, um *router*, uma webcam, um monitor de PC e colunas de som. É utilizado o programa Skype para estabelecer a vídeo chamada [13].



Figura 2-9 - Texas Robot [13].

2.3.3 - VGO

VGO (Figura 2-10) é o resultado de mais de dois anos de desenvolvimento e mais de um ano de testes com potências clientes. É um robô de baixa estatura em relação a uma pessoa adulta, movido sobre quatro rodas. É dos robôs de Telepresença mais acessíveis do mercado e está actualmente a ser utilizado com sucesso tanto em algumas empresas como por algumas pessoas singulares. Foi notícia o caso de um aluno do ensino básico, com problemas no sistema imunitário que o impediam de ir à escola, e que agora utiliza o VGO para assistir às aulas e conviver com os colegas a partir de casa [14].

De realçar como desvantagens deste robô, o facto de ter baixa estatura, se movimentar relativamente devagar e não ter auto-equilíbrio nem capacidade para circular em terrenos irregulares.



Figura 2-10 - VGO Telepresence Robot [14].

2.3.4 - Anybots QB/QA - Your Personal Avatar

QB é o nome do robô de Telepresença desenvolvido pela Anybots, uma empresa de robótica da Califórnia. Desloca-se como um Segway, equilibrando-se em duas rodas, o que o torna mais eficiente, manobrável e rápido, atingindo os 5.6 km/h. A cabeça do robô é apoiada numa barra fina e cumprida, que pode ser ajustada conforme a altura desejada (Figura 2-11). Para além da câmara de 5-megapixel na cabeça, possui também uma câmara de baixa resolução, apontada para o chão, para facilitar as manobras e impedir colisões. Possui ainda um laser para o operador do robô poder apontar para onde quiser, três microfones e colunas de som de alta qualidade.

Como os restantes robôs de Telepresença já analisados, pode ser controlado remotamente pela Internet a partir de um Browser em qualquer PC. Possui um sistema de

detecção de colisões, potentes motores e baterias de lítio que possibilitam o funcionamento do robô por 8 horas, o suficiente para um dia completo no trabalho [7], [15].



Figura 2-11 - QB Anybots Telepresence Robot [7].

Com o robô QB já em comercialização a empresa Anybots encontra-se já a desenvolver uma nova versão deste robô, o QA, mostrado na Figura 2-12. O robô QA tem praticamente as mesmas funcionalidades que o QB, no entanto, apresenta um design mais atractivo e profissional. O QA tem também a capacidade de se sentar sozinho, para poupar bateria, e voltar a levantar-se de seguida, ficando equilibrado sob as duas rodas. No entanto, custa US\$30000, o dobro do preço do QB [16].



Figura 2-12 - QA Anybots Telepresence Robot [16].

2.4 - Sumário e Principais Conclusões

Neste capítulo foram analisados sistemas robóticos já existentes envolvendo Sistemas de Telepresença e robôs com auto-equilíbrio em duas rodas, como o Segway. A presente dissertação tem como base a modificação de um Segway já existente, de forma a desenvolver um robô de Telepresença controlado pela Internet. Como tal, pretendia-se com esta análise identificar as possíveis vantagens de utilização de um veículo com a capacidade de auto-equilíbrio em duas rodas, como plataforma móvel para um robô, no lugar dos tradicionais meios de locomoção em três ou quatro rodas, assim como identificar um meio de controlo apropriado para aplicar no Segway, de forma a conseguir controlar a sua velocidade remotamente. Após a análise feita, pôde-se concluir que a utilização de robô baseado num Segway, ou seja, com a capacidade de auto-equilíbrio em duas rodas é mais vantajoso que os tradicionais meios de locomoção, principalmente quando se trata do desenvolvimento de robôs que devem interagir com pessoas e se deslocarem entre elas. Robôs com a capacidade de se auto-equilibrarem podem ser mais altos e estreitos, com uma estatura semelhante à de um homem adulto e podem sofrer grandes perturbações sem tombarem, podem circular a grandes velocidades e transportar cargas pesadas em vários tipos de terrenos, sendo bastante manobráveis.

A análise do veículo PUMA, do projecto EN-V, serviu a inspiração para o desenvolvimento de um controlador que permita o controlo remoto do Segway, alterando o seu centro de massa através da movimentação de um peso com motores lineares, simulando assim a suposta inclinação da pessoa em cima do equipamento. Esta abordagem permite a utilização do equipamento já existente, sem ser necessário a reprogramação do controlador ou alteração do equipamento de um Segway normal, para o colocar a funcionar por controlo remoto. O conceito de simular a pessoa em cima de um Segway é também a estratégia utilizada pelo robô Rover, no entanto, é feito de uma forma mais realista e eficaz, tendo um sistema mecânico que inclina um manequim para a frente e para trás. Esta abordagem seria a mais indicada e mais de acordo com o que acontece na realidade, no entanto é mais complexa sob o ponto de vista mecânico.

Neste capítulo, pretendia-se também analisar o tipo de tecnologias utilizadas em robôs de Telepresença, constatando-se a necessidade de utilização de um *router* ou *Wifi Bridge*, uma câmara IP e um PC ou Microcontrolador com interface Ethernet entre os principais equipamentos. Verificou-se também a importância do desenvolvimento de um sistema autónomo de desvio de obstáculos, uma vez que o controlo de um robô através da Internet pode apresentar atrasos ou falhas súbitas na transmissão de dados, comprometendo o seu controlo.

Capítulo 3

Hardware Utilizado

3.1 - Introdução

O presente capítulo será dedicado à apresentação do material essencial para a elaboração da dissertação. Na construção do robô Vigilante houve Hardware que foi adquirido e Hardware que foi desenvolvido, como tal, a sua descrição será realizada em subcapítulos distintos. No subcapítulo Hardware Adquirido, será apresentado o material comprado e a análise tida em conta durante a escolha de cada componente. No subcapítulo Hardware Desenvolvido, será apresentado o material que foi desenvolvido e evidenciados os processos de construção levados a cabo na sua elaboração.

Por fim, será mostrada uma tabela com os detalhes de todo o material comprado, indicando o custo de cada componente, bem como o custo total em material para o projecto.

3.2 - Hardware Adquirido

A estratégia escolhida para o controlo remoto do *Segway*, visa a alteração do seu centro de massa de forma a simular a inclinação de uma pessoa e assim movimentar o robô no sentido pretendido. Para efectuar a alteração do centro de massa, optou-se pela utilização de uma calha linear associada a um motor de passo. Optou-se por motores de passo, por serem motores que, tendo em conta o binário que produzem, são de pequena dimensão, comparativamente a um motor DC. Para além disto são motores fáceis de controlar, uma vez que permitem um controlo de posição em malha aberta.

Para o controlo remoto do robô, através da rede *Wireless*, optou-se pela utilização do PIC32 Ethernet Starter Kit, por ser um Kit de desenvolvimento já com interface Ethernet. O PIC32 será posteriormente ligado a uma Ponte Wireless, que fará a interface entre o mesmo e a rede *Wireless* da Faculdade. Uma vez que a Ponte Wireless possui também um *Switch* para 4 entradas Ethernet é possível ligar também à Ponte Wireless uma câmara IP e assim efectuar a transmissão de imagem através da rede *Wireless*. Na Figura 3-1 pode ser visto o principal hardware adquirido assim como as interfaces de ligação entre eles.

De seguida, será feita uma descrição mais pormenorizada de cada um destes componentes.



Figura 3-1 - Diagrama de ligação entre o Hardware Adquirido.

3.2.1 - Veículo auto-equilibrado de duas rodas - *Segway*

Este projecto tem por base um veículo auto-equilibrado em duas rodas, de baixo custo e *opensource* desenvolvido pela Elektor Wheelie [17]. Tem uma estrutura mecânica muito simples, pesa 35 Kg, possui dois motores DC de 500 W, duas baterias de 12V/9Ah e é controlado por um Atmega32 (controlo do motor) e um ATtiny25 (controlo de corrente). Atinge uma velocidade máxima de, aproximadamente, 18 km/h e com uma carga completa consegue fazer aproximadamente 8 km. Custa cerca de 1500 euros e já existia na Faculdade antes da realização deste projecto.



Figura 3-2 - Elecktor Wheelie Segway.

3.2.2 - Motor de Passo e Calha Linear

O motor e a calha linear são os materiais mais importantes para este projecto, uma vez que o princípio de funcionamento do sistema de locomoção do robô baseia-se na alteração do seu centro de massa. Como tal, foi feita uma análise muito rigorosa e cuidada de todos os parâmetros do motor e da calha linear, de forma a garantir a escolha do material que satisfizesse as necessidades de controlo pretendidas. Foram sem dúvida, os componentes adquiridos onde se despendeu mais tempo em pesquisa e análise, até que os componentes certos fossem encontrados.

Antes de começar a procurar o motor e a calha linear, foram feitos pequenos ensaios. Colocaram-se vários pesos em cima do *Segway* e foi se alterando a sua posição, manualmente, de forma a tentar perceber a partir de que massas o *Segway* começaria a deslocar-se. Verificou-se que com uma massa de 6 Kg já se conseguiam obter boas acelerações para o veículo. Pôde-se constatar também que para passar de uma situação de aceleração para uma situação de desaceleração, ou seja, para travar o robô, quando este se encontra a acelerar, é importante que seja possível deslocar o peso rapidamente, entre os dois extremos da calha linear, de forma a ter um tempo de travagem reduzido. Pode-se compreender melhor esta situação analisando a Figura 3-3, que ilustra as duas situações referidas. No lado esquerdo da Figura 3-3 a massa é deslocada para um dos extremos do *Segway* e o robô começa a acelerar e a ganhar velocidade. Se por algum motivo o robô necessitar de travar, este apenas começa o processo de desaceleração quando a massa é deslocada para o outro extremo do robô, como se pode ver no lado direito da Figura 3-3, daí a importância de fazer deslocar a massa de um extremo ao outro o mais rápido possível, minimizando assim o tempo de travagem.

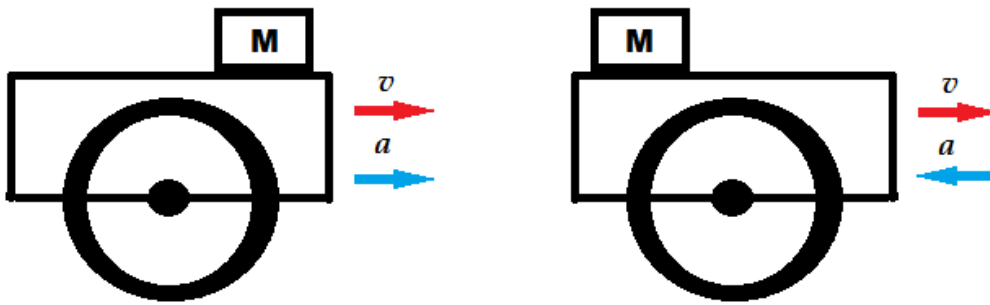


Figura 3-3 - Aceleração e Desaceleração do Segway com movimentação de uma massa M.

Com as experiências realizadas, concluiu-se que o motor linear deveria ser capaz de deslocar uma massa de 6 Kg, de um extremo ao outro da base do Segway, em, pelo menos, 1 segundo para assegurar um tempo de travagem aceitável. A base do Segway tem um comprimento de 30 cm, logo o motor deve ser capaz de deslocar linearmente o peso a uma velocidade média de cerca de 0.3 m/s.

Após esta primeira análise, procedeu-se a um trabalho de pesquisa para a calha linear, que deveria ter uma dimensão de 30cm e suportar velocidades superiores a 0.3 m/s. Recorreu-se à empresa IGUS, que disponibiliza uma grande variedade de calhas lineares feitas à medida. Optou-se pelo modelo DryLin ZLW por ser o único modelo a suportar as velocidades pretendidas. O modelo ZLW-0630-Basic-02 [18], mostrado na Figura 3-4, suporta velocidades de 2 m/s, mais do que suficiente para o pretendido.



Figura 3-4 - DryLin ZLW-0630-Basic-02-300mm [18].

A calha em questão tem uma relação movimento linear/angular de 54mm/volta. Como já foi dito, pretende-se deslocar o peso de uma extremidade à outra da calha com 0.3m em 1s, ou seja, pretende-se uma velocidade média de 0.3 m/s, o que corresponde a aplicar uma rotação na calha de, aproximadamente, 330 rpm. Pretende-se também que a calha atinja a velocidade de 0.3 m/s o mais rapidamente possível. Idealmente, para que o deslocamento de uma extremidade à outra da calha seja possível em 1s, é necessário que a calha se desloque a uma velocidade de 0.3 m/s desde o instante inicial até ao instante final, o que implica uma

aceleração infinita no início, uma vez que o peso arranca de uma velocidade nula. Como para o objectivo pretendido, o deslocamento total é realizado em 1s, é razoável admitir que o peso atinja a velocidade desejada num tempo 10 vezes inferior ao tempo total, ou seja, em 0.1s. Tomando isto em consideração, pretende-se que a calha consiga deslocar o peso de 6 Kg com uma aceleração: $\alpha = \frac{\Delta v}{\Delta t} = \frac{0.3}{0.1} = 3 \text{ m/s}^2$.

Pela análise do gráfico da Figura 3-5, retirado da *datasheet* da calha, é possível determinar o binário que o motor deverá ser capaz de produzir para conseguir ter acelerações de 3 m/s^2 com um peso de 6 Kg em cima da calha. O círculo vermelho assinala o ponto em questão, concluindo-se que é necessário um Motor com 0.5 Nm. Sintetizando, para os objectivos tomados em consideração, que possibilitam o correcto controlo do robô, é necessário um motor capaz de produzir um binário de 0.5 Nm a uma velocidade de 330 rpm.

Como já foi referido anteriormente, optou-se pelo uso de um motor de passo, pelo facto de ser de menor dimensão que um motor DC para o mesmo binário, e pelo facto de ser possível um controlo de posição em malha aberta. Foi possível encontrar um motor da Nanotec com as características de velocidade e binário apuradas anteriormente.

Required drive torque*; horizontal orientation – ZLW-0630, basic series – version 02

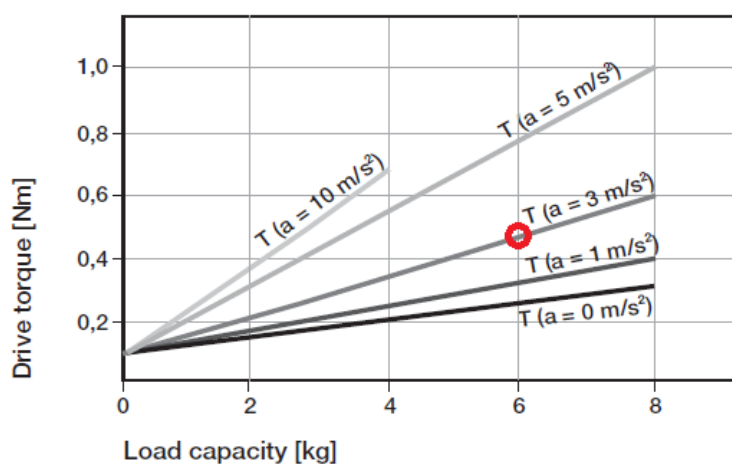


Figura 3-5 - Relação entre o Binário e a Carga para várias acelerações em movimento horizontal [18].

O motor escolhido foi o ST4118D1804, cujas características podem ser observadas na Figura 3-6. O círculo a vermelho assinala o ponto de interesse, onde com uma alimentação a 24V@1.8A o motor é capaz de gerar 0.5 Nm a uma velocidade superior a 300 rpm [19].

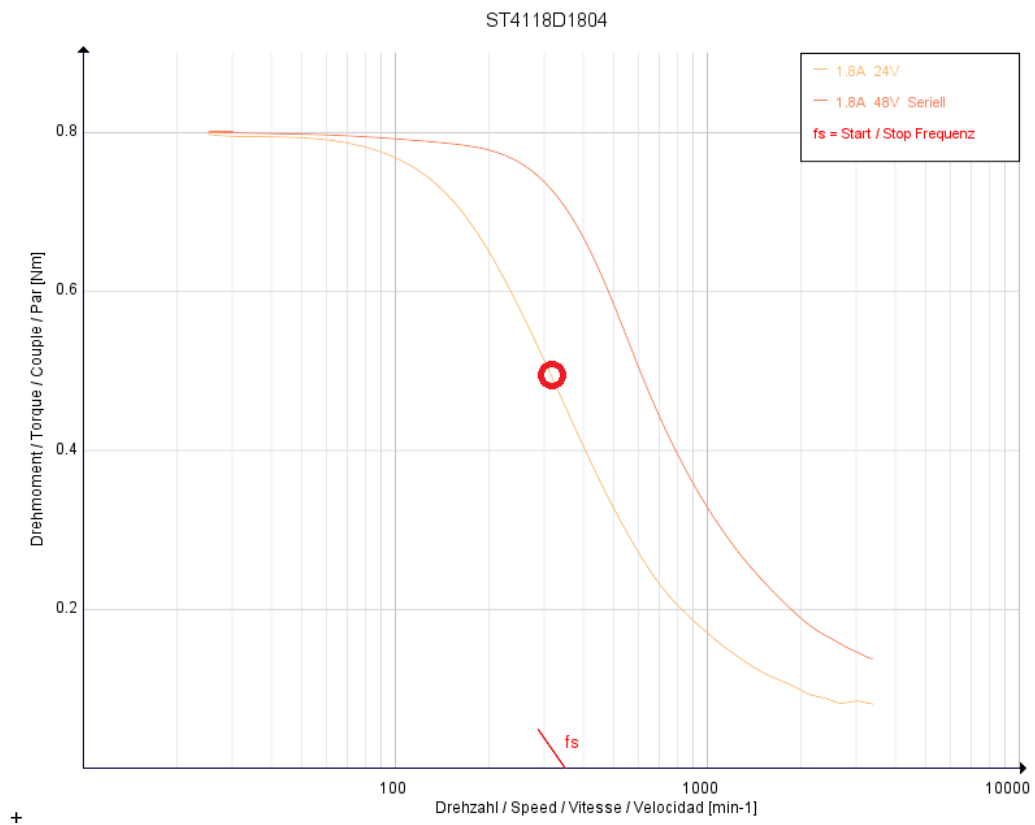


Figura 3-6 - Relação Binário/Velocidade do motor de passo escolhido [19].

O motor em questão é extremamente compacto, como se pode ver pela Figura 3-7, tendo apenas 6 cm de comprimento e 4 cm de largura, o que é bastante importante uma vez que o espaço disponível na base do Segway, não permite que o motor tenha comprimentos superiores a 15 cm.



Figura 3-7 - Motor de Passo NANOTEC ST4118D1804 [19].

3.2.3 - Controlador para Motor Passo a Passo

A escolha de um controlador apropriado para o motor passo a passo, também foi uma tarefa importante e morosa. A maior dificuldade adveio do facto de a maioria dos controladores existentes serem demasiado complexos para o pretendido, estarem preparados para um controlo a partir de um computador, oferecendo interfaces de comunicação como RS232, RS485 ou CAN, sendo necessário para a sua programação a aquisição dos respectivos cabos de programação, de elevado custo, assim como a utilização de programas específicos para o seu controlo. Não estão preparados para um controlo baseado em microprocessador, que é o que se pretende neste projecto.

O controlador SMC42 da NANOTEC adquirido, Figura 3-8, revelou-se compatível com as necessidades de controlo pretendidas. É pequeno, cerca de 4 x 5 cm, simples de controlar com um microprocessador, uma vez que como entrada de controlo admite um sinal PWM proporcional ao número de passos dados pelo motor, assim como, um sinal lógico que indica o sentido de rotação. Para além disto, satisfaz as características eléctricas de alimentação para o motor. Como já se viu o motor deve ser alimentado a 24V@1.8A, e o controlador SMC42 funciona numa gama de tensões que vai dos 21 V aos 37 V com correntes até 2 A por fase [20].



Figura 3-8 - Controlador NANOTEC SMC42 [20].

3.2.4 - PIC32 Ethernet Starter Kit

Como unidade de processamento e controlo do robô Vigilante, foi escolhido o PIC32 Ethernet Starter Kit da Microchip, por ser um sistema compacto e potente de processamento com interface Ethernet. Como se pretende efectuar o controlo do robô recorrendo à rede *Wireless*, será utilizada a interface Ethernet do PIC32 para o ligar a uma Ponte *Wireless* (*Wireless Bridge*), que por sua vez comunicará com a rede *Wireless* existente na Faculdade. Na Figura 3-9 pode ver-se uma imagem do PIC32 Ethernet Starter Kit adquirido. Trata-se de um Kit que contém interface Ethernet, CAN 2.0, assim como USB, mini-USB e micro-USB. Vem

equipado com o PIC mais potente existente actualmente, o PIC32MX795F512L, um microcontrolador de 32 bits que funciona a 80 MHz, tem 512k de Flash e 128k de RAM [21].

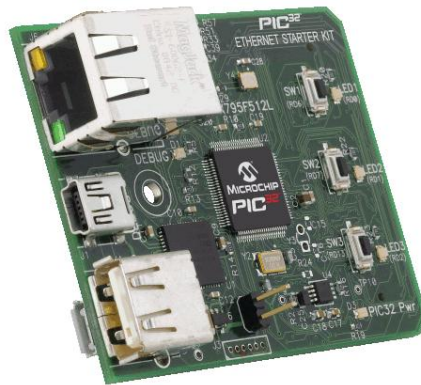


Figura 3-9 - PIC32 Ethernet Starter Kit [21].

A placa tem 6,5 cm de comprimento e 5 cm de largura, no entanto não disponibiliza o acesso directo aos pinos do PIC32. Para tal é necessário comprar outra placa, designada PIC32 I/O Expansion Board, que pode ser vista na Figura 3-10 à esquerda. A interface entre as duas placas é realizada através de um conector Hirose FX10A, presente na Figura 3-10 à direita. Uma vez que a respectiva placa de I/Os é grande e custa o mesmo que a placa do PIC32, aproximadamente 60 euros, considerou-se que a sua compra não se justificava. Para além disso, para o projecto do robô Vigilante apenas será necessário o acesso a uma dúzia de I/Os, não sendo preciso uma placa que dê acesso a todos os pinos do PIC32. Como tal, optou-se por comprar um conector macho Hirose FX10A-120S e desenvolver uma placa que dê acesso apenas às entradas e saídas necessárias para o controlo do robô. Será mostrada, à frente, a placa de circuito impresso desenvolvida para esse fim.

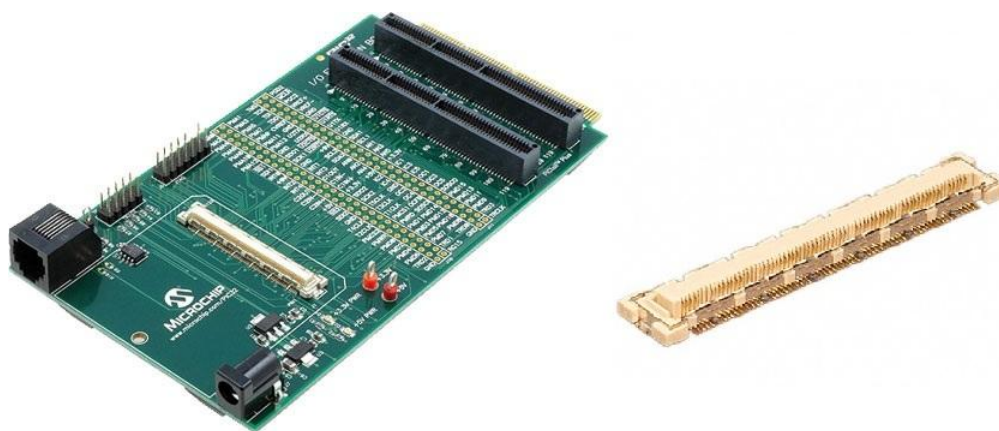


Figura 3-10 - PIC32 I/O Expansion Board e Conector Hirose FX10A-120S [21].

3.2.5 - Wireless Bridge

Uma vez que se pretende controlar o robô pela rede *Wireless* existente na Faculdade, e o microprocessador utilizado para o controlo do robô possui uma interface Ethernet, é necessário um equipamento que faça a ponte de ligação entre o cabo Ethernet e a rede sem fios Wifi. Trata-se de um equipamento designado por Ponte Wireless, semelhante a um *router*, no entanto funciona de maneira diferente. No lugar de propagar no ar o sinal da rede proveniente de um cabo, capta o sinal *Wireless* existente no meio envolvente e retransmite-o para um cabo Ethernet. Também havia a necessidade de ligar uma câmara IP, que comunicasse por cabo Ethernet, logo a Ponte Wireless também precisava de ter um *Switch* para pelo menos dois equipamentos. Tendo isso em consideração optou-se pela compra do equipamento D-Link DAP-1522. Trata-se de uma Ponte Wireless (*Wireless Bridge*) com um *Switch* para 4 entradas Ethernet. É um equipamento bastante compacto e possui a vantagem de permitir o funcionamento como Ponte Wireless, ou como Ponto de Acesso [22]. O pretendido neste projecto é o seu funcionamento como Ponte Wireless, de forma a permitir o controlo do robô em toda a extensão da Faculdade coberta pela rede. No entanto, o seu funcionamento como Ponto de Acesso também é interessante, uma vez que permite o controlo do robô em zonas onde não exista rede *wireless*. Neste modo de funcionamento o equipamento funciona como *router* e comunica directamente com um computador portátil, permitindo uma ligação mais rápida, no entanto, apenas com a extensão do alcance do *router*, ou seja, permite o controlo do robô a uma distância máxima de aproximadamente 300 m em campo aberto.



Figura 3-11 - Wireless Bridge D-Link DAP-1522 [22].

3.2.6 - Câmara IP

O robô será controlado à distância tendo como feedback de controlo as imagens em tempo real captadas. Como tal, pretendia-se a instalação de uma câmara que pudesse enviar as imagens captadas pela rede de Internet, sem causar atrasos na transmissão. Existem câmaras por IP e câmaras *Wireless*, as câmaras IP funcionam por cabo Ethernet e podem ser ligadas à Ponte Wireless que por sua vez comunica com a rede Wifi. As câmaras *Wireless* possuem uma antena, que permite que as mesmas possam comunicar directamente com a rede Wifi. No entanto o uso deste tipo de câmara, implicaria que fosse necessário efectuar a

configuração de acesso à rede da Faculdade, quer na Ponte Wireless quer na câmara, podendo ainda haver complicações em configurar a câmara à rede Wireless da Faculdade, devido às protecções de acesso existentes. Como tal, optou-se desde logo pela escolha de uma câmara IP, sendo apenas necessário ligá-la por cabo Ethernet à Ponte Wireless para a colocar a funcionar.

Pretendia-se também que a câmara permitisse a transmissão de áudio bidireccional, de forma a conseguir, de uma forma simples, integrar no robô um sistema de “vídeo-chamada”, permitindo ao operador do robô não só visualizar a pessoa, como ter uma conversa com a mesma à distância.

Tendo tudo isto em consideração optou-se pela câmara Zavio F210A, mostrada na Figura 3-13. Optou-se por esta marca, por ser a câmara mais acessível que cumpria os requisitos mencionados.



Figura 3-12 - Câmara IP ZAVIO F210A [23].

3.3 - Hardware Desenvolvido

Para além do material comprado foi necessário desenvolver algum Hardware, como Placas de Circuito Impresso (*PCB - Printed Circuit Board*) e partes mecânicas para o suporte do material no robô. Na Figura 3-13, em baixo, pode ver-se a estrutura mecânica definida previamente para a construção do robô. Como se pode ver na Figura, a ideia inicial passava por afixar uma placa de acrílico por cima do *Segway* e, nessa placa, criar os suportes para o material principal, como a calha e o motor. Pretendia-se posteriormente a colocação de um manequim por cima dessa estrutura, de forma a esconder todo o Hardware e tornar o robô mais alto e mais amigável. O *router* e a câmara serão colocados nas posições superiores, fixados ao manequim. A câmara será colocada na cabeça do manequim de forma a oferecer uma imagem mais privilegiada e o *router* na zona do peito, para captar melhor o sinal da rede *wireless*.

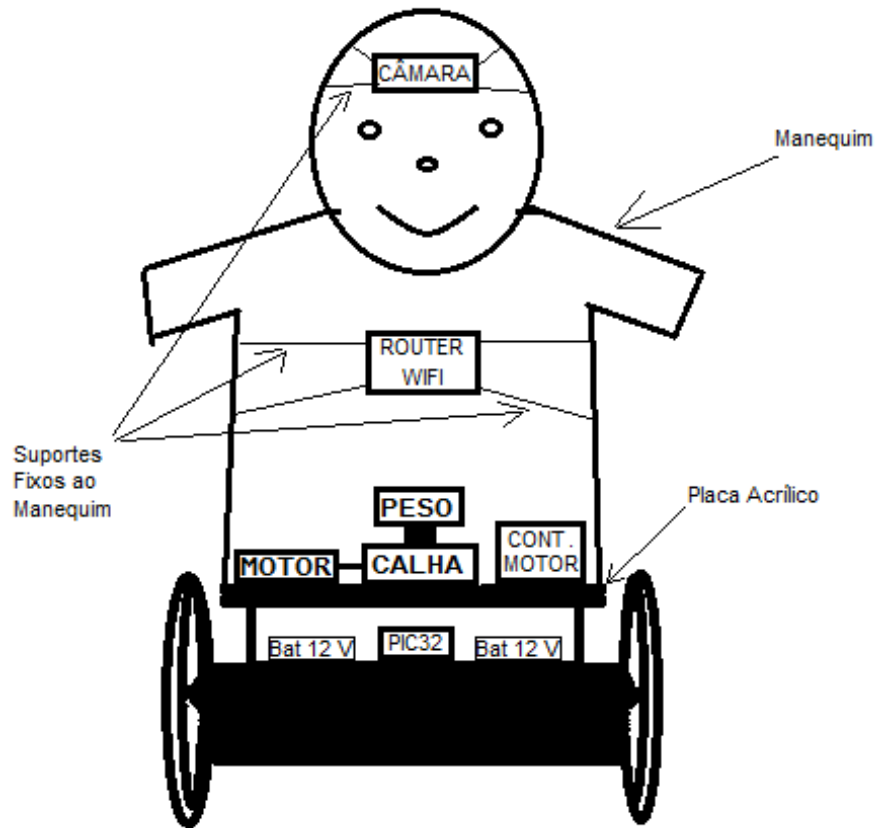


Figura 3-13 - Estrutura definida inicialmente para a construção do robô.

De seguida será apresentado todo o material desenvolvido e ilustrados os processos de desenvolvimento utilizados.

3.3.1 - Placas de Circuito Impresso (PCBs)

Como já foi referido o PIC32 Ethernet Starter Kit, é um *kit* que não oferece o acesso directo aos I/Os do PIC32. Para tal existe um conector SMD, Hirose FX10A-120S, que possibilita a ligação de uma outra placa para acesso aos referidos pinos. Desenvolveu-se assim uma PCB para esse fim e para permitir também a alimentação do PIC32 a partir das baterias do robô. Uma vez que se encontrou na Internet uma biblioteca com o desenho do conector Hirose FX10A-120S para o software Eagle, o desenho desta placa foi efectuado utilizando este software. A Figura 3-14 apresenta a placa referida. Recorreu-se à *datasheet* do PIC32 Ethernet Starter Kit, para saber a localização dos pinos de interesse no conector Hirose FX10A-120S. Os pinos de interesse foram os ADCs, PWMs e a alimentação. O PIC32 deve ser alimentado a 5V. Assim, para ir buscar energia aos 24V das baterias, utilizou-se um conversor comutado de 5V@1,5A, também incluído nesta placa. Uma vez que será necessário alimentar outros equipamentos a 5V, como o *router* e a câmara IP, colocaram-se também nesta placa duas saídas de 5V disponíveis.

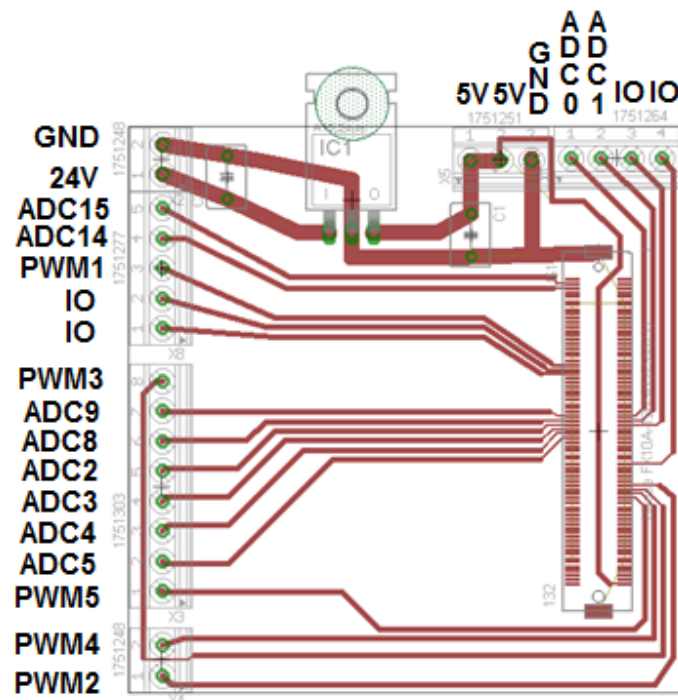


Figura 3-14 - PCB para Alimentação e dar acesso às I/Os do PIC32 Ethernet Starter Kit.

Para a implementação do controlador *Fuzzy*, responsável pelo controlo de velocidade do robô, foram necessários alguns sinais de *feedback* do sistema de controlo original do *Segway*. Sinais como o ângulo de inclinação e a velocidade actual do robô são fundamentais, uma vez que são os parâmetros de entrada para o controlador *Fuzzy* desenvolvido. Devido à importância da correcta aquisição desses sinais, será de seguida explicado em secções distintas, a forma como ambos os sinais foram obtidos.

3.3.1.1 Aquisição do ângulo

Para a obtenção do ângulo de inclinação, utilizou-se o sinal proveniente do acelerómetro do *Segway*. Verificou-se experimentalmente que o sinal resultante apresentava uma variação de aproximadamente 100mV para os ângulos de interesse (ver Tabela 3-1). De forma a obter a resolução máxima do ADC do PIC32, utilizou-se uma montagem baseada em AmpOps, numa configuração diferencial, de forma a colocar o sinal do acelerómetro a variar de 0V a 3,3V. A tensão de saída da configuração utilizada (Figura 3-15) é dada por:

$$V_{out} = \frac{R_2}{R_1} \times (V_2 - V_1) \quad (3.1)$$

onde o quociente entre as resistências R2 e R1 determina o ganho da subtracção dos dois sinais de entrada.

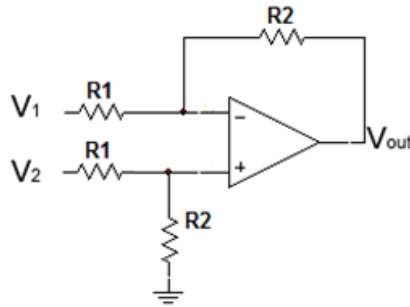


Figura 3-15 - Circuito subtrator.

Através da análise dos dados mostrados na Tabela 3-1, recolhidos experimentalmente, chegou-se à equação que relaciona linearmente a tensão, V_m , a ser lida pelo ADC do PIC32, e a tensão fornecida pelo acelerómetro, V_a :

$$V_m = 41.25 \times (V_a - 1.44) \quad (3.2)$$

Tabela 3-1 - Registo dos valores de tensão do acelerómetro para os ângulos de interesse.

Ângulo	Tensão acelerómetro (V_a)	Tensão para micro (V_m)
-30°	1.44V	0V
0°	1.48V	1.65V
30°	1.52V	3.3V

Fazendo a comparação entre a equação 3.1 e a equação 3.2, determinam-se os parâmetros para o circuito da Figura 3-15:

$$\begin{cases} V_2 = V_a \\ V_1 = 1.44 \\ R_2/R_1 = 41.25 \end{cases} \quad (3.3)$$

A fonte de tensão de 1.44V obteve-se com uma fonte de corrente ajustável. O integrado LM334 pode ser programado para gerar correntes de 1µA até 10mA e após a escolha da corrente pretendida, este mantém o valor da corrente inalterado, mesmo que a tensão de alimentação do circuito varie. Assim, basta escolher um valor adequado para a corrente e, a partir daí, calcular a resistência para obter a tensão pretendida. Após uma análise à *datasheet* do LM334, pôde-se verificar que a corrente, I_{SET} , gerada depende da resistência R_{SET} que deve ser colocada entre dois dos três terminais do integrado, como se pode ver pelo esquema eléctrico da Figura 3-16. A corrente depende também da temperatura e pode ser calculada pela seguinte expressão:

$$I_{set} = \frac{227\mu V / ^\circ K}{R_{set}} \quad (3.4)$$

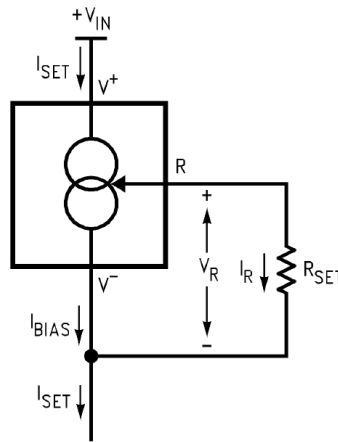


Figura 3-16 - Circuito para o controlo da corrente do LM334.

Assumindo uma temperatura de 20°C (293°K), para programar o integrado de forma a este fornecer a corrente máxima que o mesmo permite, ou seja, 10mA é necessário uma resistência R_{SET} de:

$$R_{set} = \frac{227\mu \times 293}{10m} \cong 6.7\Omega \quad (3.5)$$

A resistência mais próxima do valor calculado disponível é de 5.6Ω, no entanto, uma vez que com uma resistência inferior a 6.7Ω, o valor da corrente programada é superior aos 10mA pretendidos, o uso de uma resistência mais baixa que a calculada não causa qualquer problema, devido ao facto de o integrado estar limitado nos 10mA. Definiu-se portanto, uma resistência $R_{SET} = 5.6\Omega$. Por fim, para obter a tensão de 1.44V desejada com uma fonte de corrente fixa de 10mA, basta aplicar a Lei de Ohm para determinar a resistência a colocar à saída da fonte de corrente, concluindo ser necessário uma resistência de saída de: $R_o = 1.44/10m = 144\Omega$. Optou-se então pela colocação de um potenciómetro multi-volta de 500Ω à saída do LM334, de forma a facilitar o correcto ajuste da tensão pretendida.

3.3.1.2 Aquisição da velocidade

Uma vez que o *Segway* utilizado como base para a construção do robô não possui *encoders*, uma boa forma de estimar a velocidade do mesmo é através da tensão aplicada nos dois motores DC do veículo. Para tal, optou-se pela medição dos sinais de PWM enviados pela placa de controlo do *Segway*, em vez da medição directa da tensão aplicada nos motores. A tensão aplicada aos motores varia de -24V a 24V, enquanto os PWMs variam de 0V a 5V, representando a mesma variação e sendo mais simples de medir, por se tratarem sempre de sinais positivos. Pela análise do esquema eléctrico do *Segway* da Elektor Wheelie, pôde-se

verificar que o Atmega32 fornece 4 sinais, que estão relacionados com a velocidade aplicada a cada uma das rodas do veículo. Os sinais, assinalados pelo círculo a vermelho na Figura 3-17, correspondem aos PWMs que indicam a magnitude da velocidade, assim como os sinais digitais que indicam a direcção de rotação, para cada uma das rodas.

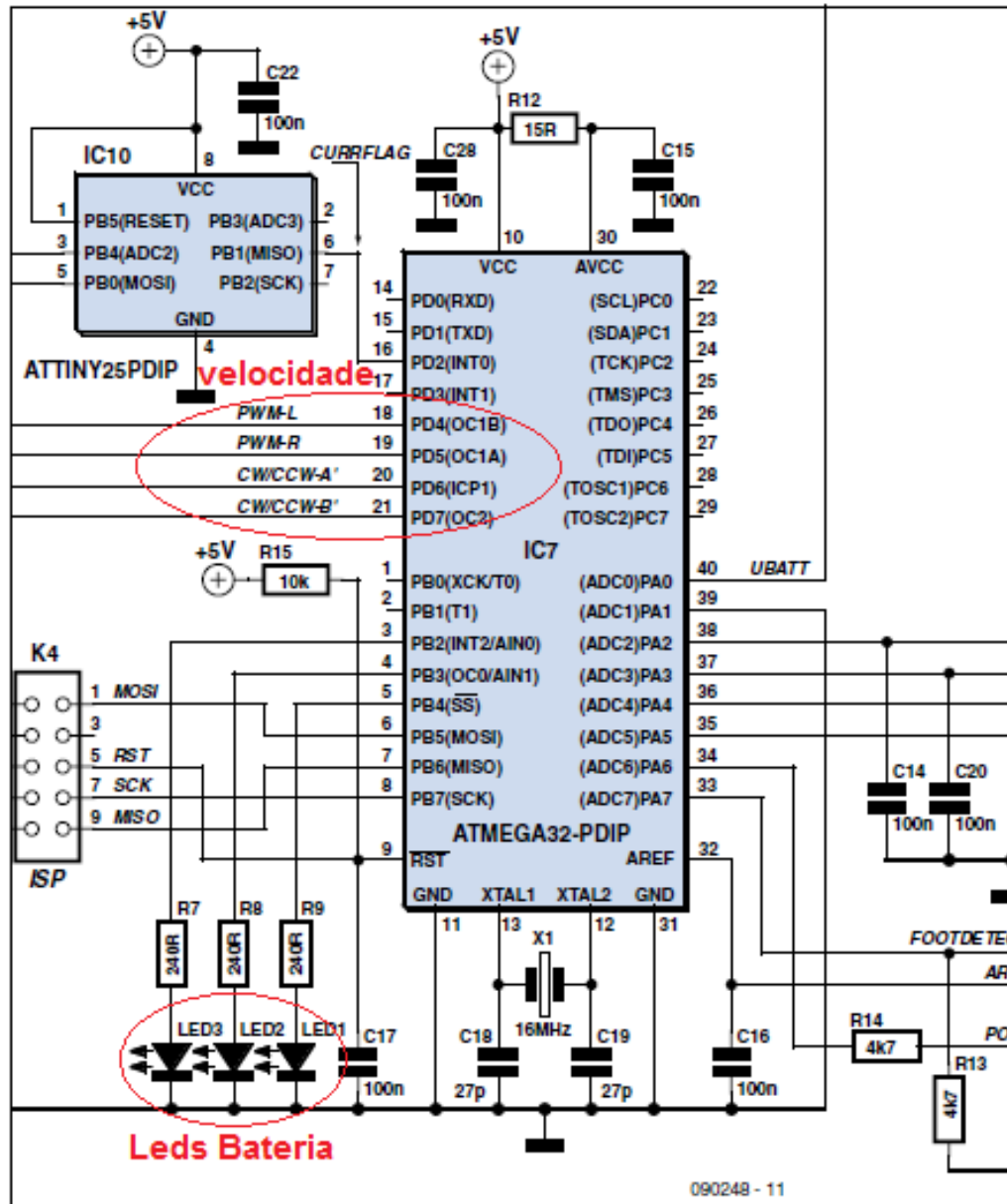


Figura 3-17 - Excerto do esquema eléctrico do sistema de controlo original do Segway da Elektor Wheelie [17].

Outro parâmetro importante, fornecido pela placa original de controlo do Segway, e que se pretendia medir era o estado de carga da bateria. Assim, o operador do robô pode prevenir situações de descarga que têm como consequência a perda de equilíbrio e consequente queda do robô. Uma vez que o Segway possui um conjunto de 3 leds (verde, amarelo e vermelho) que acendem consoante o estado da bateria, optou-se por ir adquirir os sinais que alimentam

os leds, para, assim, poder enviar a informação do estado da bateria do robô pela Internet. Na Figura 3-17 estão assinalados, com um círculo vermelho, os leds em questão. Os pinos do Atmega32 usados para a indicação do estado da bateria são os pinos 3, 4 e 5.

O condicionamento efectuado, nos sinais retirados directamente dos pinos do Atmega32, foi a conversão de 5V para 3.3V, para os mesmos, poderem ser lidos correctamente pelos ADCs do PIC32. Introduziu-se ainda um filtro passa-baixo passivo, de primeira ordem, para os sinais de PWM enviados para o controlo dos motores. Deste modo obteve-se uma tensão contínua, a ser lida pelos ADCs do PIC32, e capaz de fornecer informação sobre a amplitude da velocidade do *Segway*.

O condicionamento de sinal necessário, de forma a permitir ligar os sinais provenientes do sistema de controlo do *Segway* ao PIC32, usa uma PCB desenvolvida pelo Autor. A placa foi desenvolvida utilizando o software Multisim/Ultiboard e o seu desenho pode ser visto na Figura 3-18. Para além de incorporar todo o condicionamento de sinal necessário, esta placa amplifica em potência às saídas do PIC32, através de transístores. Foi adicionado também, nesta placa, um conversor comutado, de 24V para 5V, para alimentação de equipamentos externos. Juntando os dois conversores comutados existentes nas duas placas desenvolvidas, consegue-se obter uma fonte de tensão de 5V@3A, suficiente para alimentar o PIC32, o *router* e a câmara IP.

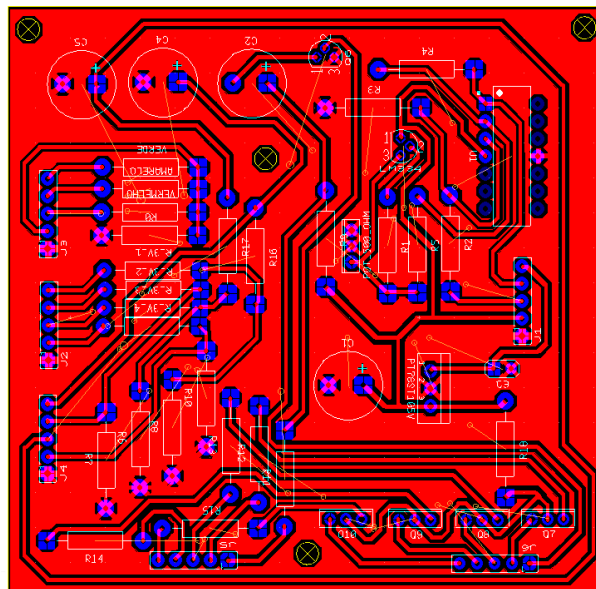


Figura 3-18 - PCB para aquisição e condicionamento de sinais do Segway (Ultiboard).

Na Figura 3-19, pode ser vista a parte do circuito eléctrico desenvolvido para a PCB de condicionamento, responsável pelo tratamento do sinal proveniente do acelerómetro. Como se pode ver, no conector J1 entra em Va o sinal do acelerómetro, saindo em Vm um sinal pronto a ser lido pelo ADC do PIC32, sinal esse que varia entre 0V e 3.3V.

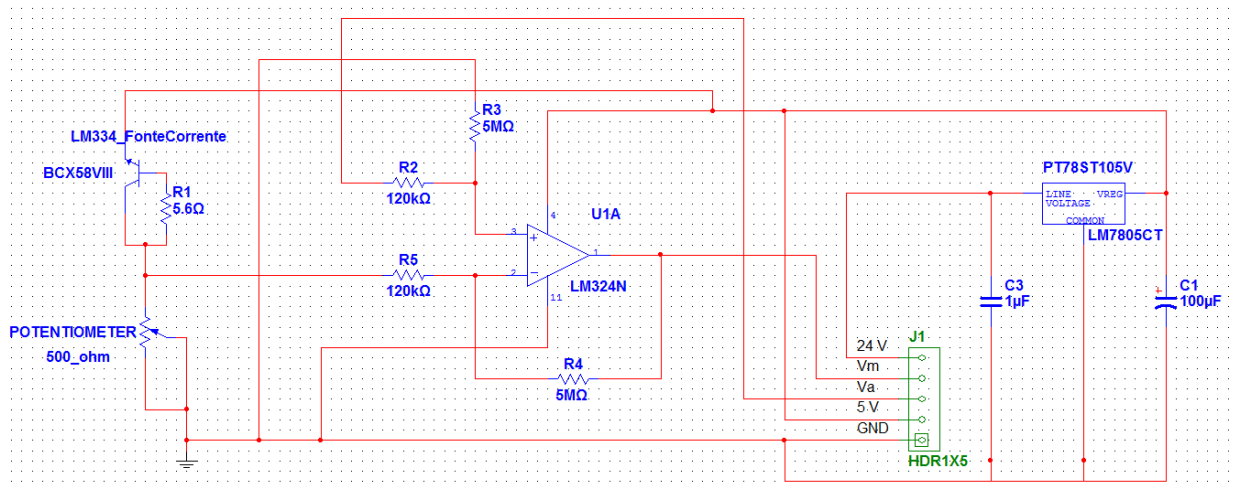


Figura 3-19 - Esquema elétrico para condicionamento do sinal do acelerômetro (Multisim).

Na Figura 3-20 em baixo, pode ser visto do lado esquerdo, o tratamento de sinal feito para obter o estado da bateria do robô e do lado direito, o tratamento para a aquisição da velocidade. Como já foi visto, para a aquisição do nível da bateria, foram-se buscar 3 sinais lógicos aos pinos 3, 4 e 5 do Atmega32 (ver Figura 3-17). De forma a transformar estes 3 sinais lógicos de 5V, num único sinal a variar entre 0V e 3.3V, utilizaram-se divisores resistivos de diferentes escalas para os diferentes sinais. Desta forma, um único sinal a ser lido pelo ADC do PIC32 fornece a informação sobre o estado da bateria.

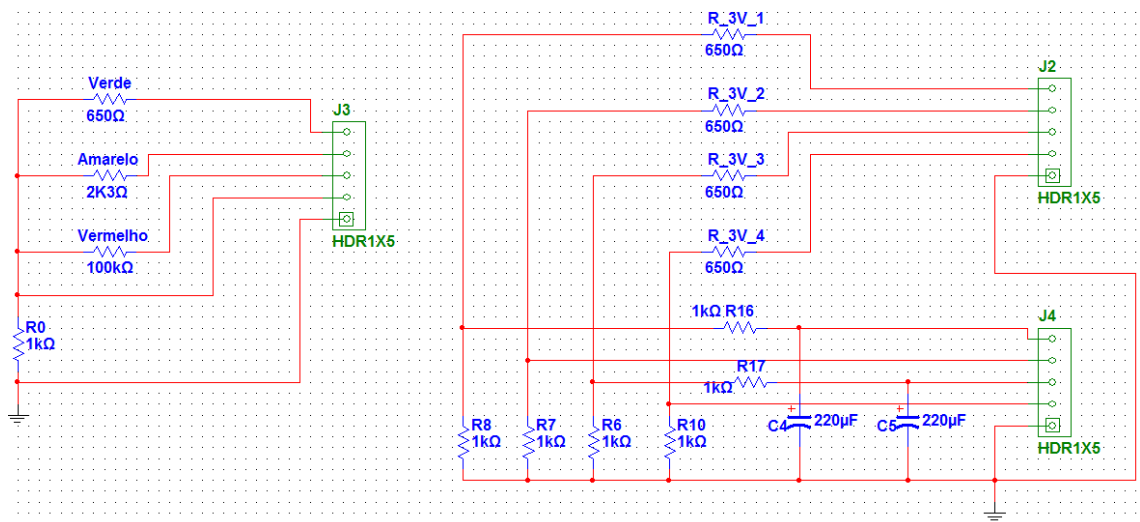


Figura 3-20 - Esquema elétrico para condicionamento dos sinais que permitem determinar o nível da bateria e a velocidade do robô.

O circuito do lado direito da Figura 3-20 mostra o condicionamento feito aos 4 sinais que permitem determinar a tensão aplicada aos motores do Segway, e assim, obter

aproximadamente a velocidade do robô. Como se pode ver, foi utilizado um divisor resistivo, assim como um filtro passa-baixo nos 2 sinais de PWM.

Por fim, a Figura 3-21, ilustra o circuito que amplifica em potência às saídas do PIC32, possibilitando um sinal de saída com 5V@1.5A, no lugar dos 3.3V@10mA que as portas do PIC32 fornecem.

Na Figura, pode ser visto também, marcado com o círculo vermelho, a parte eléctrica que permite efectuar o controlo de viragem do robô. No *Segway* original da Elektor Wheelie, a viragem é conseguida com o girar de um guiador, acoplado a um potenciômetro. Como tal, o sistema de controlo do *Segway*, vira o veículo em resposta à aplicação de uma tensão, que pode variar entre 0V e 5V. Como se pode ver pelo esquema, isso é conseguido pela utilização de um PWM gerado pelo PIC32 e por um filtro passa-baixo. O PWM gerado pelo PIC32 é transformado, pelo filtro passa-baixo, numa tensão entre 0V e 5V, a ser aplicada ao sistema de controlo de viragem do *Segway*.

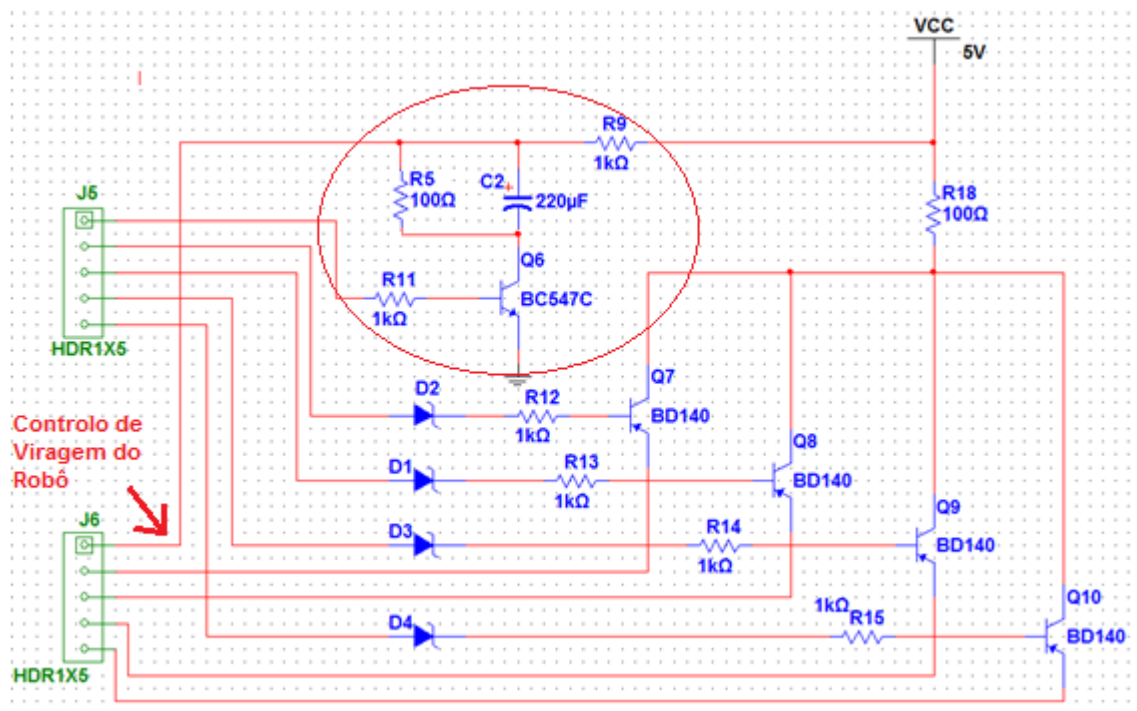


Figura 3-21 - Esquema eléctrico do circuito que permitiu dar potência aos sinais de saída do PIC32.

A Figura 3-22 e a Figura 3-23 pretendem demonstrar as ligações feitas à placa de controlo original do *Segway*. Foram soldados cabos directamente à placa, nos pinos de interesse já referidos. Nas ligações foram utilizados cabos compridos passados para fora pela abertura existente para a saída do guiador, que foi retirado. O facto de se passar todos os sinais de interesse para fora da estrutura, permitiu efectuar a partir daí, todo o trabalho sem haver a necessidade de voltar a desmontar o *Segway*.

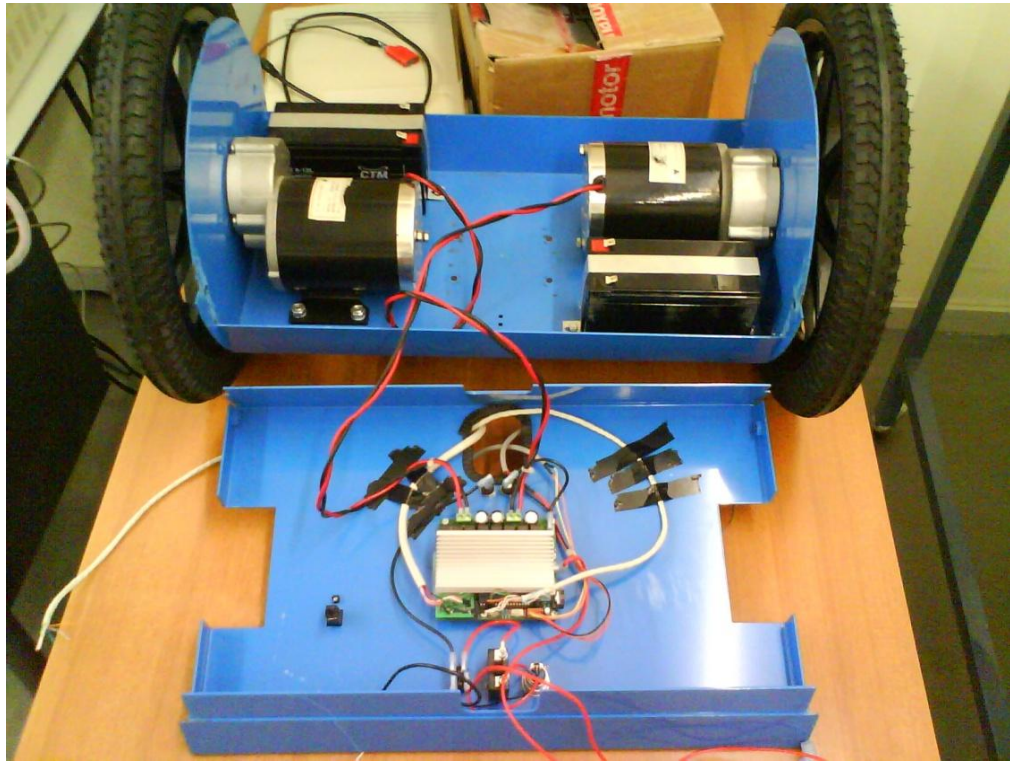


Figura 3-22 - Interior do Segway original.

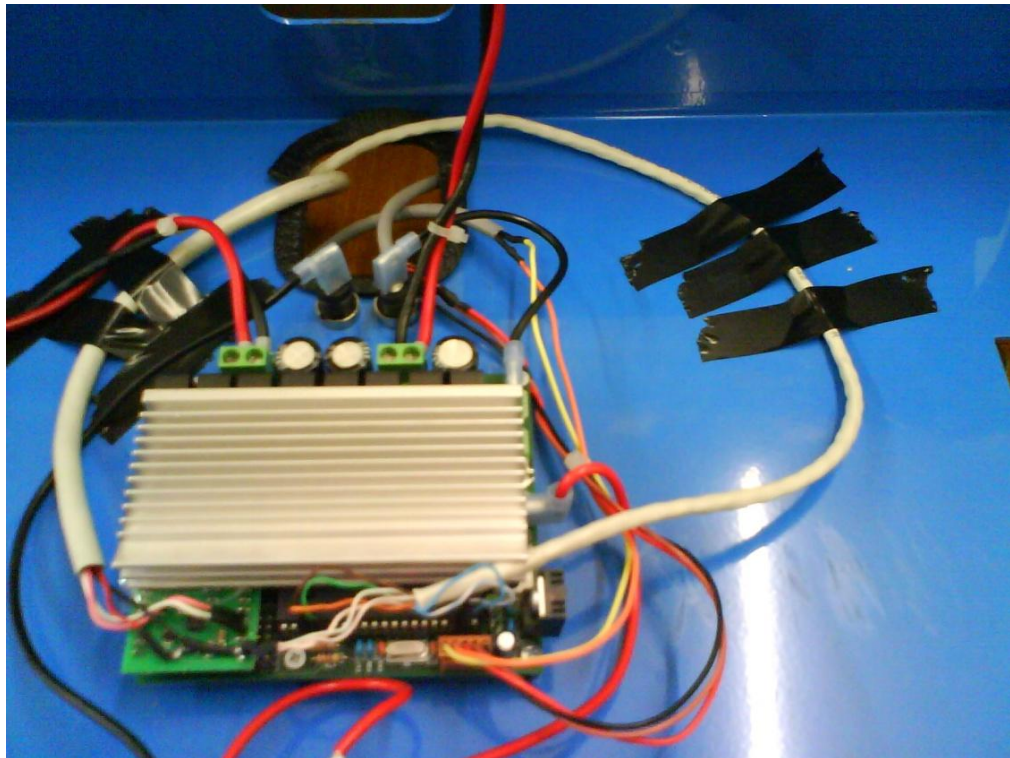


Figura 3-23 - Ligação de cabos à placa original do Segway para obtenção de sinais: ângulo de inclinação, velocidade e nível da bateria.

A Figura 3-24 mostra a primeira versão da placa que permite dar acesso aos IOs do PIC32 Ethernet Starter Kit, esta placa apresentava alguns defeitos que foram corrigidos numa segunda iteração. A Figura 3-25 mostra a PCB desenvolvida, ligada ao PIC32.

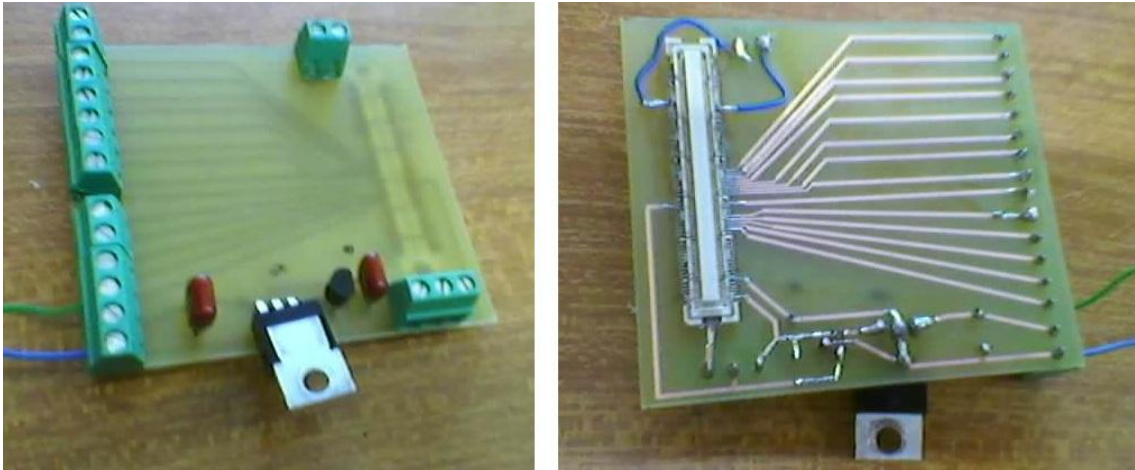


Figura 3-24 - Primeira versão da placa que permite dar acesso aos IOs do PIC32.

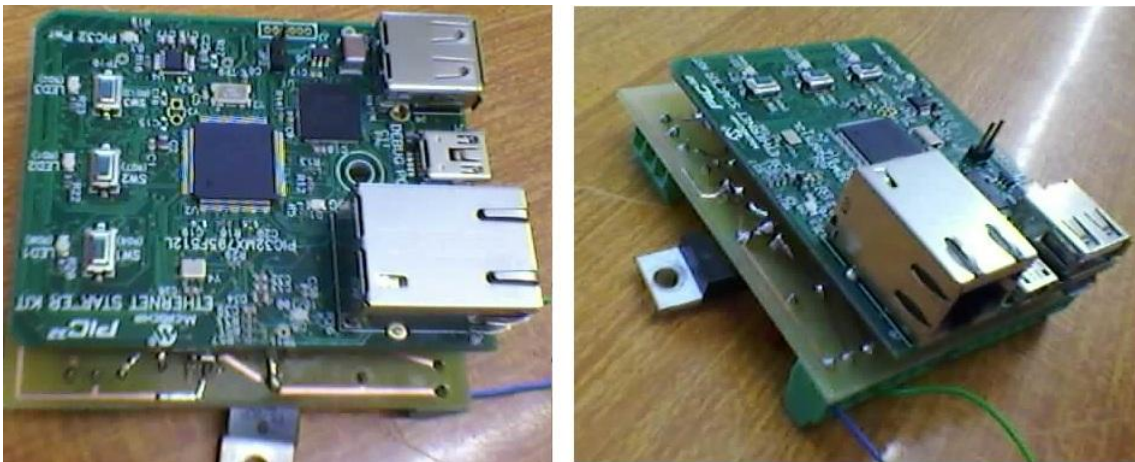


Figura 3-25 - PIC32 Ethernet Starter Kit ligado à placa desenvolvida.

Por fim, a Figura 3-26, mostra a versão final das duas PCBs desenvolvidas. Na Figura, as PCBs encontram-se já correctamente ligadas ao robô. Pode ser visto, que a placa que permite dar acesso aos I/Os do PIC32, está ligada ao PIC32 Ethernet Starter Kit, através do conector Hirose. Por baixo, encontra-se a placa responsável por todo o condicionamento de sinal, fazendo a ponte de ligação entre os sinais provenientes da placa de controlo do Segway e a placa do PIC32. Toda a electrónica e todo o processamento utilizado para o controlo do robô pela Internet encontra-se nas placas mostradas na Figura 3-26. As placas apresentadas

forneem também a alimentação necessária para o funcionamento do controlador do motor de passo, do *router* e da câmara IP.

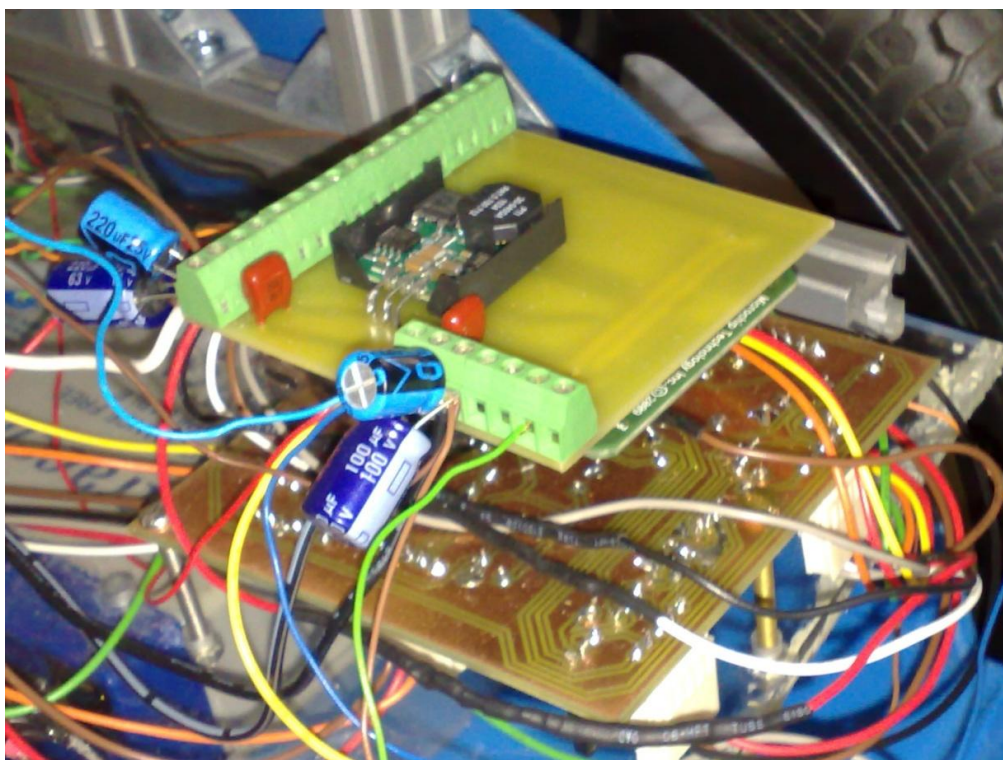


Figura 3-26 - Versão final das duas placas desenvolvidas - ligadas ao PIC32 e ao robô.

3.3.2 - Estrutura Mecânica do Robô

Foi necessário construir algumas estruturas mecânicas adicionais no *Segway*, de forma a suportar as baterias adicionais, a calha que desloca o peso, o motor, o controlador para o motor, as PCBs de controlo, o *router* e a estrutura de suporte para o manequim.

A primeira coisa a fazer foi retirar do *Segway* original a estrutura mecânica da barra vertical, o guiador, que permitia ao ocupante apoiar-se e controlar a viragem. De forma a minimizar os estragos na fuselagem do *Segway*, foram feitos apenas 8 furos na sua fuselagem, 4 de cada lado, onde se afixaram barras que suportam as baterias, assim como todo o resto da estrutura. Foi seguido mais ou menos a estrutura definida previamente e ilustrada na Figura 3-13, no início deste subcapítulo. A partir dos suportes criados para a bateria fez-se um suporte para uma placa de acrílico. Nesta fixou-se todo o material, como se pode ver pelas Figuras abaixo. A sequência das Figuras demonstram o processo evolutivo da construção, até se atingir o resultado final. Nas Figuras é possível ver o peso utilizado, para modificar o centro de massa do robô. Tinha-se projectado o robô para a utilização de uma massa com 6 Kg. No entanto, usou-se uma massa de ferro de 4 Kg. Foi presa com parafusos a uma placa de madeira, que por sua vez ficou afixada à estrutura da calha linear.

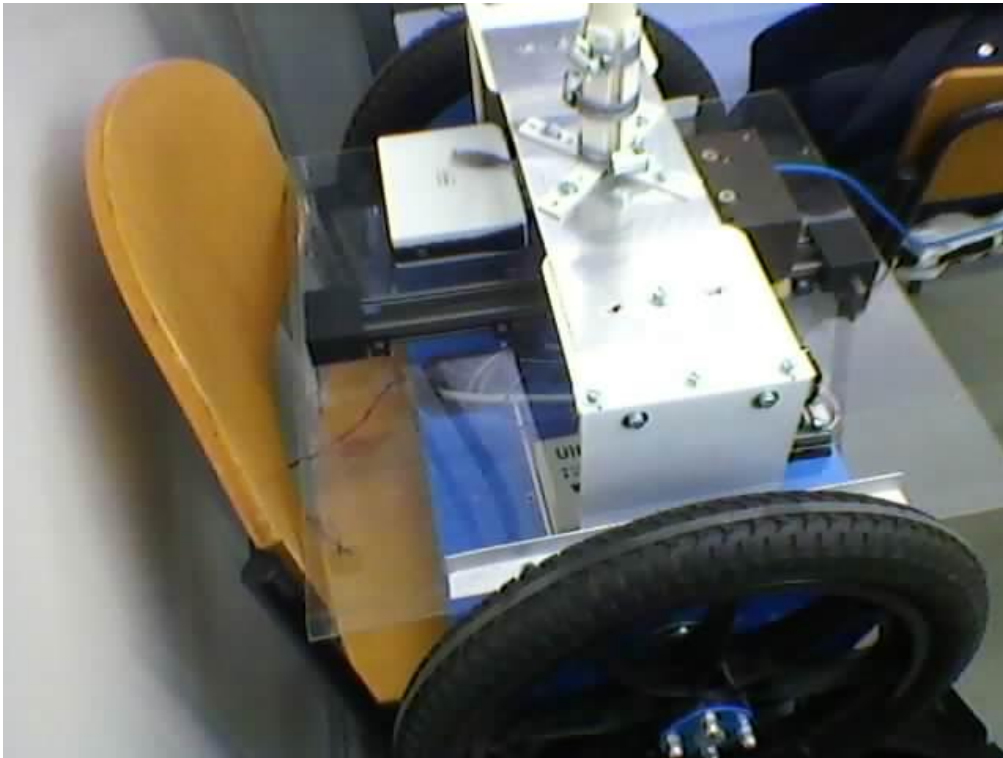


Figura 3-27 - Fase inicial da estrutura mecânica construída.

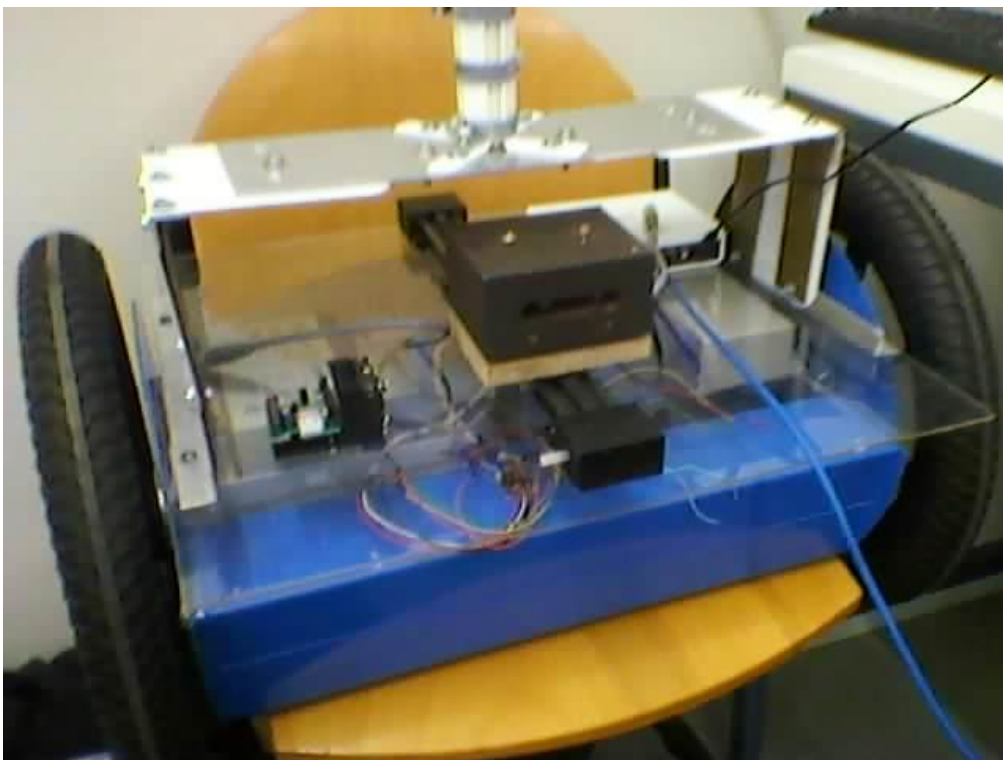


Figura 3-28 - A calha, o *router* e o controlador do motor, afixados ao acrílico.

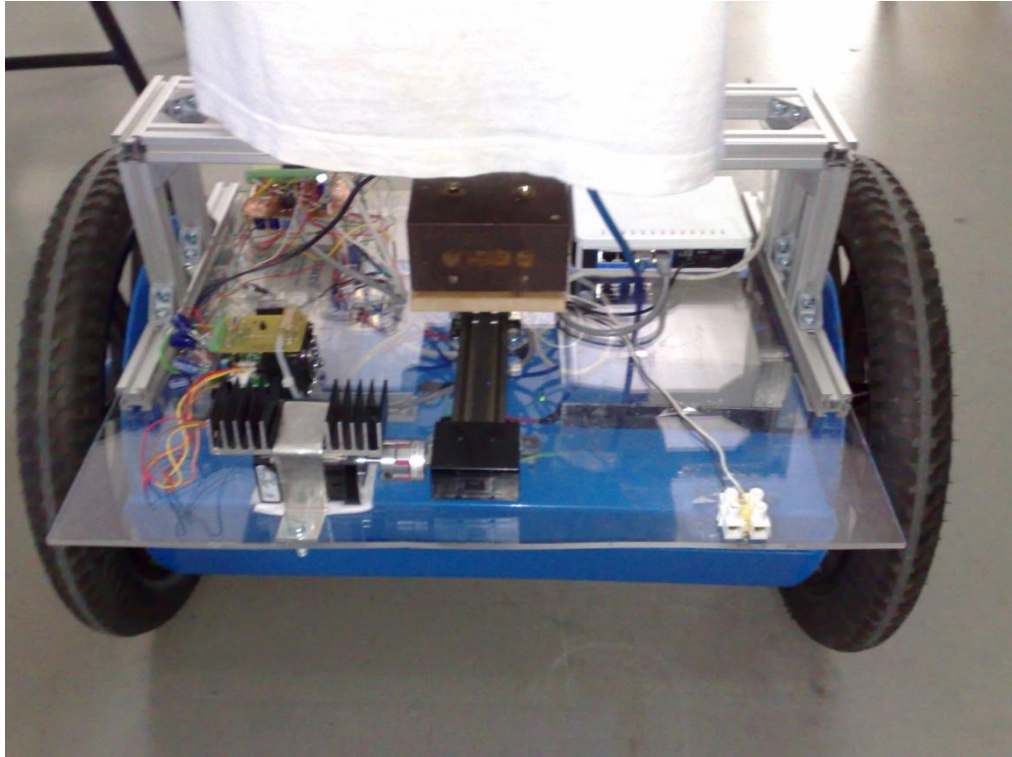


Figura 3-29 - Fase final da estrutura mecânica construída (foto nº1).

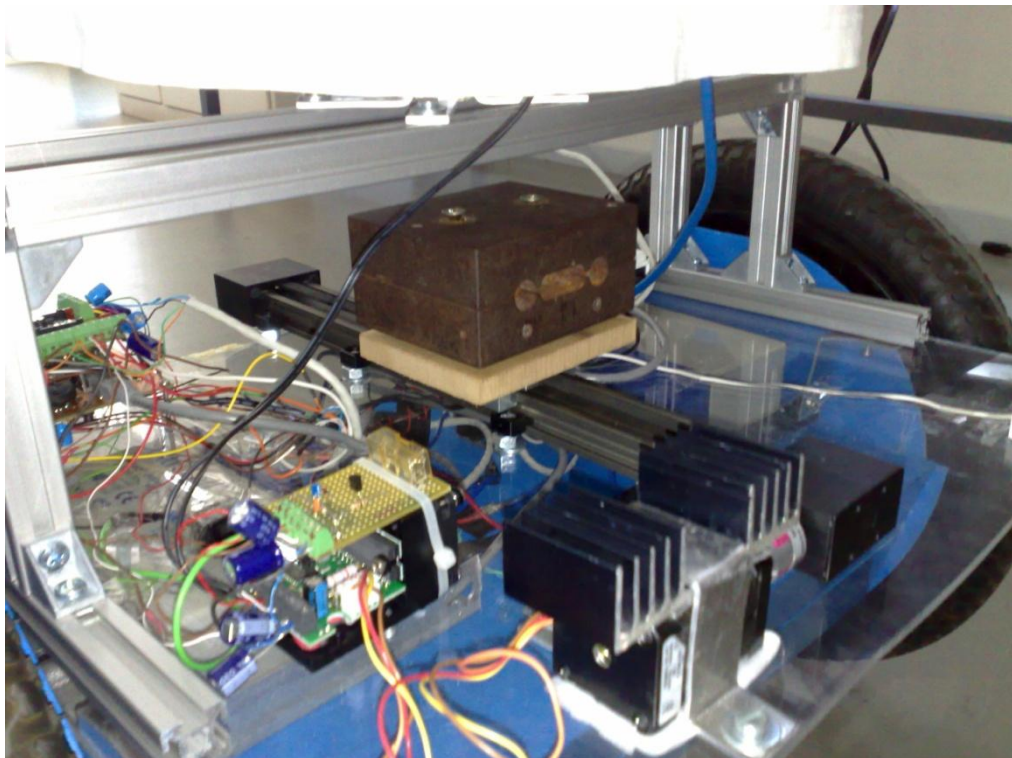


Figura 3-30 - Fase final da estrutura mecânica construída (foto nº2).

Na Figura 3-31, é possível ver em detalhe o motor de passo, acoplado à calha linear. Como se pode ver, foi adicionado um dissipador ao motor, por se ter verificado que o mesmo aquecia demasiado. Colocou-se também um amortecimento mecânico, por baixo do motor, para neutralizar as vibrações causadas pela rotação do mesmo.



Figura 3-31 - Motor de passo, acoplado à calha linear.



Figura 3-32 - Ultra-sons instalados na parte da frente do robô.

Na Figura 3-32, é possível ver os sensores de ultra-sons instalados na parte frontal do robô. Os sensores servem para ajudar o operador em manobras apertadas, assim como para impedir que o robô embata em obstáculos, mesmo que receba ordem para tal. Foi necessário o desenvolvimento de uma *Veroboard* (Figura 3-33), para permitir instalar os sensores na placa de acrílico. A mesma possui um filtro passa-baixo para a recepção do sinal do eco. Mais adiante, será explicado o porquê do seu uso.

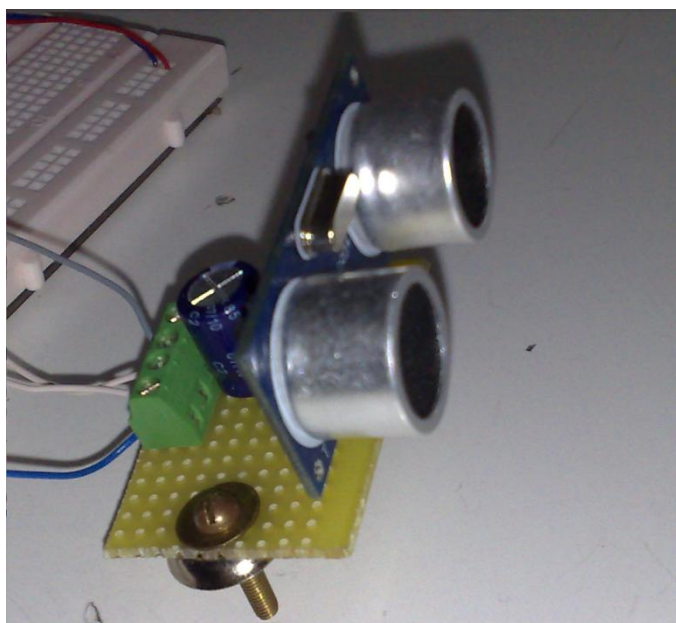


Figura 3-33 - *Veroboard* desenvolvida para Sensor de Ultra-sons.

A Figura 3-34 ilustra a forma como a câmara foi fixa no robô, através da utilização de um capacete, assim como a lente adicional que foi instalada. Após os primeiros testes da câmara adquirida no robô, constatou-se rapidamente que era quase impossível o controlo do mesmo a partir das imagens da câmara, em espaços apertados, como o interior de uma sala. Isto porque, a câmara adquirida apresentava uma imagem bastante ampliada e direccionada, com um ângulo de visão bastante reduzido. Um obstáculo mesmo encontrando-se bastante longe do robô, aparentava, na imagem, estar muito próximo. Para resolver o problema da câmara, adquiriu-se uma lente de grande angular, desenvolvida para telemóveis, que permite uma fácil instalação na câmara através de imans. A lente adquirida, revelou melhorias significativas, aumentando em 67 vezes a abertura da imagem, permitindo o correcto controlo do robô mesmo em espaços apertados, tendo como feedback as imagens captadas. Por fim, a Figura 3-35 e a Figura 3-36, mostram a versão final do robô desenvolvido.



Figura 3-34 - Câmera IP com lente de grande angular instalada (*wide angle lens*).



Figura 3-35 - Robô, num passeio pela FEUP.



Figura 3-36 - robô Vigilante.

3.4 - Sumário e Principais Conclusões

Neste capítulo foi analisado o hardware principal utilizado na construção do robô vigilante. Esta parte consumiu um tempo considerável neste projecto. Na compra do material, teve-se o cuidado de procurar apenas os componentes que satisfizessem os requisitos mínimos pretendidos, de forma a não encarecer demasiado o projecto. Relativamente ao Hardware desenvolvido, como as placas de circuito impresso e a estrutura mecânica, exigiram bastante tempo de projecto e construção. As PCBs tiveram que passar por uma segunda iteração, para corrigir alguns defeitos. No entanto o resultado final foi bastante satisfatório.

A Tabela 3-2 ilustra o material adquirido, indicando o preço de cada componente e o preço total do projecto no final. De notar que os preços aqui indicados estão ausentes dos custos de portes de envio. De referir ainda, que apesar de constar na tabela e de terem sido encomendados os sensores de ultra-sons e as colunas de som, estes componentes não foram adquiridos, devido a problemas com os respectivos fornecedores. Foram utilizados outro tipo de sensores de ultra-sons já existentes na Faculdade (Figura 3-33).

Tabela 3-2 - Material encomendado.

Material robô Vigilante	Referência	Preço (€)	Empresa	Contactos
Motor de Passo	NANOTEC - ST4118D1804	36,48	NANOTEC	BIBUS Portugal, Lda Rua 5 de Outubro, 5026 4465-079 S. Mamede de Infesta; Tel: +351 229 065 050
Controlador Motor Passo	NANOTEC - SMC42-2,0-1	78,04	FARNELL	Tel.: +34 93 475 88 04; Fax: +34 93 474 52 88; Email: vendaspt@farnell.com
Calha Linear	ZLW-0630-Basic-02-300mm	289,05	IGUS	igus®, Lda R Engº Ezequiel Campos, 239, 4100-231 Porto; Tel. +351 22 6109000; Fax +351 22 832 8321; pvilasboas@igus.pt
Micro-controlador	PIC32 Ethernet Start Kit	67,64	FARNELL	Tel.: +34 93 475 88 04; Fax: +34 93 474 52 88; Email: vendaspt@farnell.com
Wireless Bridge	DAP-1522	92,15	PIXMANIA	Via Catarina Shopping, Loja 4.09/10, PISO 4 (Restauração), Rua de Santa Catarina, 4000-443 Porto; porto@pixmania.com; Tel: 808203311
Câmara IP	ZAVIO F210A	193,90	ChipSite	Avenida D. Carlos I, Nº 36 2720-161 Amadora - Portugal; Telf: 21 499 60 00;
2 x Sensor Ultra-	Maxbotix LV-EZ1	58,91	Lusorobotica	LusoRobótica - Robótica em

Sons				Portugal; geral@lusorobotica.com
2 x Bateria Chumbo 12 V	Bateria 12 V, 7 Ah Extracell	32,52	AquarioNet	Rua da Alegria, 93-A,4000-042 Porto; Telf: 223394780; Fax: 222001379; Telm: 919551794,932082746,96614 7394; jbraga@aquario.pt
Colunas de som	Philips - SBA 1500	28,12	MINFO	Vendas On-line Minfo.pt; Telefone: 259 328 880; Email: vendas@minfo.pt
Connector Hirose PIC32	FX10A-120S/12-SV(71)	7,77	FARNELL	Tel.: +34 93 475 88 04; Fax: +34 93 474 52 88; Email: vendaspt@farnell.com
Manequim		20		Zona industrial da Varziela
Capacete		5		Póvoa de Varzim
Placa Acrílico	Maxmat	25		Póvoa de Varzim

Total Projecto (€)	847,58
-----------------------------------	--------

Capítulo 4

Adaptação de um Segway para uma Plataforma Robótica Móvel

4.1 - Introdução

O robô vigilante tem por base um veículo com auto-equilíbrio em duas rodas, semelhante a um Segway PT, que se desloca para frente, ou para trás, em resposta a uma inclinação da pessoa que transporta. O seu sistema de controlo tenta manter o veículo na posição vertical. Se este começar a pender para frente o sistema de controlo acelera o veículo nessa direcção, tentando compensar a queda, mantendo-o assim na vertical. Uma pessoa que se desloca em cima de um *Segway* está constantemente a forçar uma inclinação, fazendo com que o sistema de controlo a tente corrigir, adicionando assim velocidade ao veículo.

Para se colocar um sistema como o do Segway PT a ser controlado remotamente, funcionando como um robô móvel, existem essencialmente duas opções viáveis:

i) Desenvolver um sistema que altere o centro de massa do veículo, simulando assim a inclinação de uma pessoa em cima do mesmo;

ii) Desenvolver um sistema para alterar, ligeiramente, o ângulo de referência de controlo do *Segway*, alterando a referência vertical e fazendo-o, assim, acelerar na direcção pretendida.

Esta última abordagem funcionaria, no entanto, apenas com pequenos desvios do valor real do ângulo, fazendo com que o *Segway* se deslocasse muito devagar. Seria uma solução de fácil implementação mas má para a deslocação do robô, que estaria limitado em aceleração e velocidade. Uma terceira hipótese passaria ainda pelo desenvolvimento de um sistema de controlo de raiz, que permitisse não só o controlo do *Segway* na posição vertical, como também o controlo da sua posição e velocidade, tal como acontece no Segway RMP. Esta opção exige no entanto, o desenvolvimento de um sistema de controlo de grande complexidade, uma vez que os dois controladores pretendidos (controlo de posição e

estabilidade) são antinómicos. Trata-se de um problema semelhante ao clássico pêndulo invertido sobre um carro, onde se pretende efectuar o controlo de posição do carro e ao mesmo tempo manter o pêndulo equilibrado.

Tendo tudo isto em consideração, e dos robôs tomados como exemplo no estado da arte, optou-se pelo desenvolvimento de um controlador que simula o movimento da pessoa em cima do veículo. Com a análise do projecto PUMA surgiu a inspiração para colocar um motor linear em cima do *Segway*, suportando uma massa que, ao ser deslocada, altera o centro de massa do veículo, fazendo-o movimentar-se.

De forma a testar o conceito de alteração do centro de massa de um *Segway*, para conseguir assim controlar a sua posição e velocidade, foi desenvolvido um simulador tridimensional e realista do robô, recorrendo à plataforma SimTwo. Foi também desenvolvido um controlador de velocidade, baseado em lógica difusa, responsável por movimentar o peso na base do *Segway*, de forma a manter a velocidade pretendida para o robô. Após o desenvolvimento e teste do controlador de velocidade no simulador foi também analisada a sua implementação no robô real. Foi aqui visto também, como os sensores de ultra-sons foram utilizados para permitir ao robô evitar obstáculos.

4.2 - Simulação da dinâmica do robô recorrendo à plataforma SimTwo

A simulação do controlo de um sistema instável é uma tarefa bastante importante, uma vez que permite ajustar os parâmetros de um dado controlador, cometendo os erros na simulação e não no sistema real, o que poderia causar danos ao mesmo ou até mesmo a terceiros. Mesmo que não se tenha um modelo exacto do sistema é importante simular, para se ter uma ideia da resposta obtida. No entanto, quanto melhor modelado estiver o sistema, melhor se comportará o controlador quando este for aplicado ao sistema real.

Para a simulação do robô, foi utilizada a plataforma SimTwo. O SimTwo é um sistema de simulação realista, desenvolvido pelo Professor da FEUP Paulo Gomes da Costa, e que permite a implementação de vários tipos de robôs móveis, assim como manipuladores, quadrúpedes e até humanóides. No fundo é possível simular qualquer tipo de robô terrestre, que possa ser descrito como uma mistura de articulações rotativas e rodas clássicas/omnidirecionais, assim como veículos mais-leves-que-o-ar com ou sem hélices para propulsão.

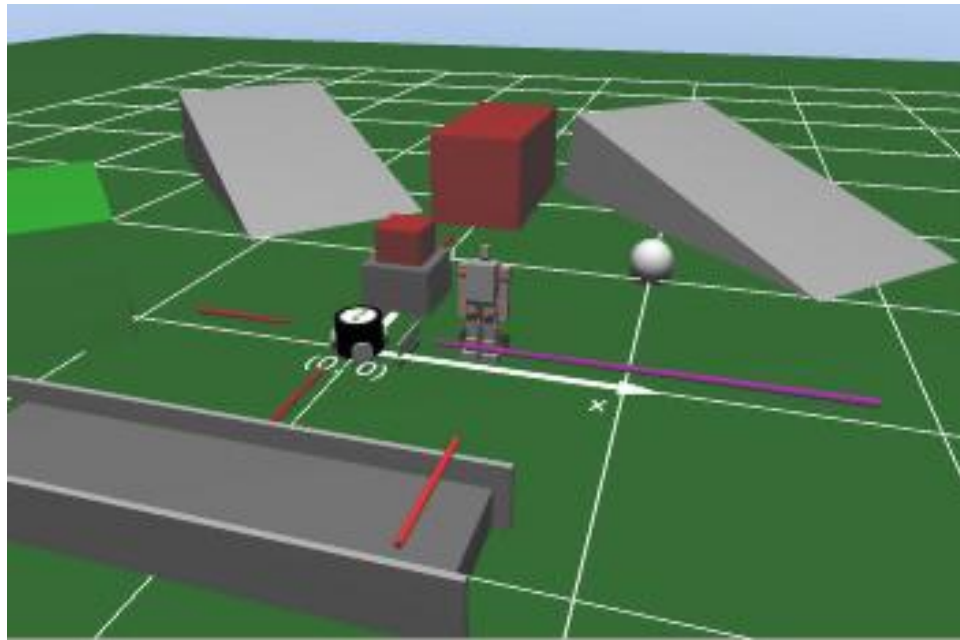


Figura 4-1 - Plataforma SimTwo [24].

O realismo da dinâmica implementada no SimTwo é conseguido decompondo o robô num sistema de corpos rígidos ligados entre si por vários tipos de articulações. A "mecânica" associada aos corpos é simulada numericamente considerando as suas propriedades físicas: forma, massa e momentos de inércia, atritos das superfícies e elasticidade. As articulações podem ter associado um sistema de accionamento ou um sistema de sensores. O sistema de accionamento pode ser constituído por um motor DC com caixa redutora. O modelo do motor DC contém vários elementos não lineares, como a saturação da tensão aplicada, limite de corrente e atrito de Coulomb [24].

O robô Vigilante desenvolvido no SimTwo pode ser visto na Figura 4-2. Foram tidas em consideração as massas e as dimensões reais do robô. Os parâmetros eléctricos que modelam os motores DC foram obtidos com ensaios ao Segway real. O bloco a vermelho mostrado na Figura 4-2, modela o peso que pode ser deslocado linearmente na base do Segway, permitindo assim a alteração do centro de massa do robô.

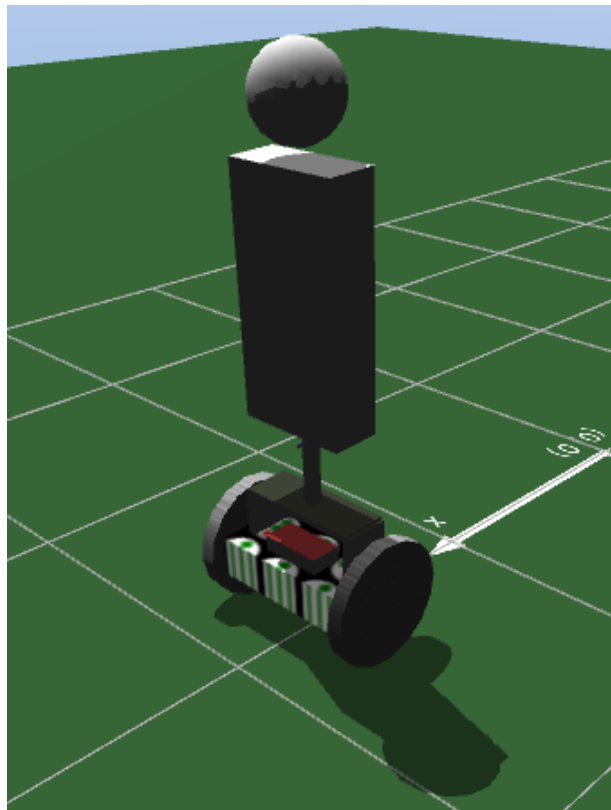


Figura 4-2 - Simulação do robô Vigilante.

4.3 - Desenvolvimento de um Controlador de Velocidade para o Segway e sua Implementação e teste no SimTwo

Apesar de, para o robô real, se pretender apenas o desenvolvimento de um controlador que possa controlar remotamente a velocidade do *Segway*, foi também necessário o desenvolvimento e implementação de um controlador responsável por manter o robô equilibrado em duas rodas. Isto porque, para testar o controlador de velocidade no simulador é necessário ter também um *Segway* simulado. Foi utilizado um controlador *Fuzzy*, para os dois controlos pretendidos. Optou-se por este tipo de controlador, no lugar dos tradicionais controladores PID, pelo facto de os sistemas a controlar serem não lineares e de difícil modelação matemática. Um controlador *Fuzzy* é projectado tendo em conta o conhecimento prévio de como o controlador deve operar, podendo esta informação ser passada de uma forma linguística, tornando assim, a implementação de controladores complexos mais simples e rápida, sem a necessidade de modelar matematicamente o sistema para determinar os parâmetros correctos para uma determinada resposta. Para ambos os controladores, foi utilizada a arquitectura de mandani, onde as entradas e as saídas dos controladores são conjuntos difusos. De seguida serão analisados os dois controladores desenvolvidos.

4.3.1 - Controlo responsável por manter o robô equilibrado em duas rodas (controlo do Segway)

As regras desenvolvidas para o controlador *Fuzzy* responsável por manter o robô equilibrado no simulador, podem ser vistas na Figura 4-3. À esquerda está a tabela de correspondência entradas/saída. À direita apresenta-se um *Segway*, que ilustra as orientações tomadas em consideração para o ângulo e para a velocidade a aplicar nos motores. O ângulo θ representa a inclinação do robô em relação a posição vertical (ângulo zero). A velocidade angular $\dot{\theta}$ é obtida derivando discretamente o ângulo entre dois ciclos de execução do programa no simulador, o que ocorre de 40 em 40 ms. As siglas utilizadas na Figura 4-3 para descrever as 9 regras *Fuzzy* correspondem a valores: MN - Muito Negativo; N - Negativo; Z - Zero; P - Positivo e MP - Muito Positivo.

Entradas	θ			
	Valores	N	Z	P
$\dot{\theta}$	N	MN	N	Z
	Z	N	Z	P
	P	Z	P	MP

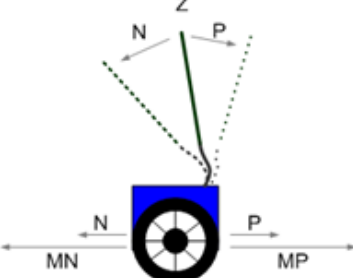


Figura 4-3 - Regras Fuzzy para o controlo de equilíbrio do robô no SimTwo.

As regras mostradas na Figura 4-3 pretendem representar os seguintes princípios, deduzíveis logicamente pela análise do funcionamento do sistema:

- Se o Segway se encontrar completamente imóvel na posição vertical (ângulo e velocidade angular nulas), os motores não devem actuar;
- Se o Segway se encontrar na posição vertical (ângulo nulo) e estiver a movimentar-se (velocidade angular não nula), os motores devem ser actuados ligeiramente no sentido de corrigir o previsível futuro erro do ângulo;
- Se o ângulo for Positivo/Negativo e a velocidade angular for também Positiva/Negativa, significa que o Segway está numa posição de queda que se irá acentuar ainda mais no futuro, de forma que os motores devem ser actuados à potência máxima no sentido de contrariar a queda;
- Se o ângulo for Positivo/Negativo e a velocidade angular Negativa/Positiva, os motores não devem actuar, uma vez que apesar de existir um erro no ângulo, prevê-se que esse erro diminua, devido à velocidade angular de sinal contrário existente.

Após a implementação das regras apresentadas na Figura 4-3, utilizando a ferramenta para criação de controladores *Fuzzy* disponível no Matlab, obteve-se a correspondência entre as entradas e a saída representada pela superfície mostrada na Figura 4-4. A saída varia entre -24 e 24 Volts, representando a tensão a ser aplicada aos motores. A saída do controlador pode ser exportada do Matlab para um ficheiro de texto. O ficheiro de texto é apresentado sob o formato de uma matriz 15x15, onde cada posição da matriz corresponde à saída do controlador para um determinado par de entradas. Para se exportar o controlador *Fuzzy* desenvolvido no Matlab para um ficheiro de texto é necessário executar os seguintes comandos:

```

fis = readfis('RoboVigilante.fis');
[x y z] = gensurf(fis);
save -ascii 'TabelaFuzzy.txt' z;

```

ao executar estes comandos, o controlador existente no ficheiro “RoboVigilante.fis” é exportado em forma de tabela para o ficheiro de texto “TabelaFuzzy.txt”.

O SimTwo contém uma função que permite passar directamente os dados contidos num ficheiro de texto para uma matriz, como tal, para incorporar o controlador *Fuzzy* desenvolvido no SimTwo basta fazer:

```
SegwayControl := Mload('TabelaFuzzy.txt');
```

onde SegwayControl é uma matriz 15x15, que contém as tensões a aplicar nos motores do Segway, para que este se mantenha equilibrado na vertical, consoante o ângulo e a velocidade angular do mesmo.

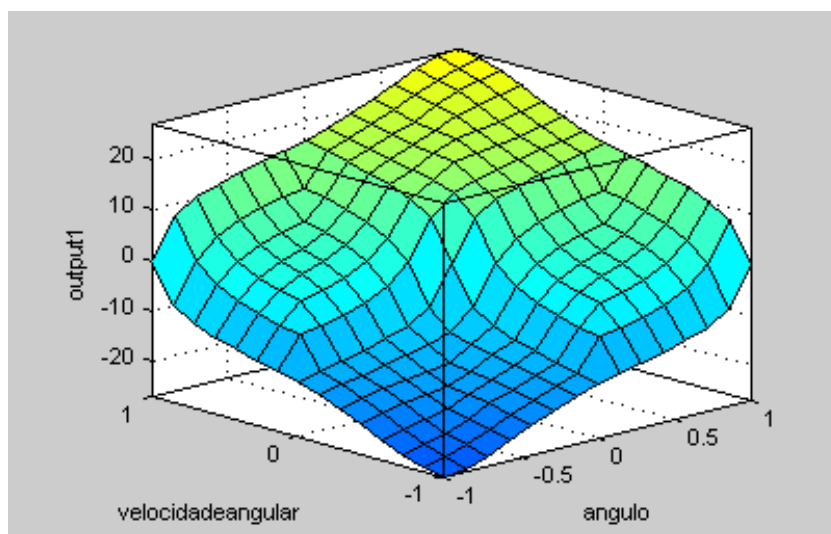


Figura 4-4 - Superfície *Fuzzy* responsável pelo equilíbrio do robô na posição vertical.

Os valores das entradas, ou seja, do ângulo e da sua derivada, foram normalizados para variarem de -1 a 1, de forma a facilitar a obtenção da saída do controlador a partir da matriz. Para se extrair o valor da matriz de controlo, para um determinado par de entradas que variam entre -1 a 1, é necessário transformar essa variação em valores inteiros entre 0 e 14 para corresponderem aos índices da matriz *SegwayControl*. Uma vez que o MatLab apenas disponibiliza uma tabela de correspondência com 225 dados (15x15), para os pares de entrada não disponíveis na tabela é necessário fazer interpolação ou arredondamento para o valor mais próximo contido na tabela. Por se ter demonstrado eficaz e simples de implementar, foi utilizado o sistema por arredondamento para o valor mais próximo.

De seguida é mostrado um excerto do código implementado no SimTwo. Trata-se do procedimento responsável pelo equilíbrio do robô na posição vertical, através da leitura dos dados contidos na matriz de controlo. Como se pode ver, primeiro os valores de entrada são normalizados para terem uma variação entre -1 e 1. Posteriormente esta variação é transformada numa variação entre 0 e 14, para corresponder a um índice da matriz que contém os dados de controlo.

```

procedure Segway_Control;
begin
    i_teta_sat := sat(i_teta/30 ,1); //normalização do ângulo (entre -1 e 1).
    i_w_sat := sat(i_w/5 ,1); //normalização da velocidade angular (entre -1 e 1).
    i_v := 0;
    Step := 2/14 := 0.1429; //intervalo entre os valores de -1 a 1 na tabela.
    Row := Round(i_w_sat/Step)+7; //velocidade angular a variar entre 0 e 14.
    Col := Round(i_teta_sat/Step)+7; //ângulo que agora varia entre 0 e 14.
    i_v := Mgetv(SegwayControl, Row, Col); //vai buscar a saída do controlador.
end;

```

Com esta abordagem, e após um período de testes e ajustes, foi possível colocar o controlo responsável por manter o robô equilibrado em duas rodas a funcionar correctamente no SimTwo.

4.3.2 - Controlo de velocidade do robô (simulação do comportamento de uma pessoa em cima do *Segway*)

As regras desenvolvidas para o controlador *Fuzzy*, responsável por controlar a velocidade do robô através da modificação do seu centro de massa, podem ser vistas na Figura 4-5. À esquerda está a tabela de correspondência entradas/saída e à direita um esquema de um *Segway*, para demonstrar as orientações tomadas em consideração. O bloco a vermelho representa o peso que pode ser deslocado linearmente na base do *Segway* para assim actuar

sob o seu centro de massa. Como se pode ver, como parâmetros de entrada para o controlador de velocidade do *Segway*, utilizou-se o erro da velocidade e o erro do ângulo.

Entradas	$e_v = V_{ref} - V$			
	Valores	N	Z	P
$e_\theta = -\theta$	N	MN	N	N
	Z	N	Z	P
	P	P	P	MP

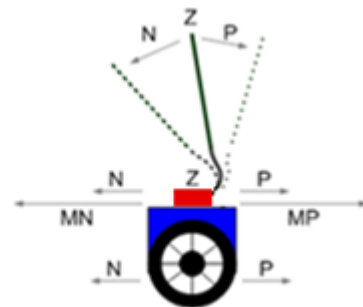


Figura 4-5 - Regras Fuzzy para o controlo de velocidade do robô

O controlador de velocidade foi desenvolvido de forma a comportar-se como uma pessoa em cima de um *Segway*. A pessoa inclina-se para frente para se deslocar. De cada vez que o faz adiciona aceleração ao veículo e, quanto mais se inclina, maior é a aceleração produzida. A aceleração gerada cria a força necessária para o equilíbrio, forçando a posição vertical. O condutor ao promover a aceleração do veículo adiciona velocidade, no entanto, não pode estar continuamente a produzir uma aceleração, uma vez que à medida que a velocidade aumenta a aceleração que o mesmo pode produzir diminui, acabando por ficar com menos força para manter o sistema equilibrado na vertical. O que acontece na realidade é que o condutor vai gerando gradualmente impulsos de aceleração. Ele inclina-se para a frente uma vez, adicionando aceleração ao veículo até que o mesmo volte à posição vertical, depois volta a inclinar-se para frente gerando mais aceleração. Ao fazer isto repetidamente vai adicionando velocidade. No entanto, se o mesmo ficar sempre a forçar uma inclinação para a frente, o veículo chega a um ponto em que não consegue restabelecer a posição vertical, fazendo-o cair. Como tal, o controlador de velocidade desenvolvido controla o centro de massa do veículo não apenas tomando em consideração o erro da velocidade, como também o ângulo que o veículo faz com a vertical. Por exemplo, se o *Segway* estiver parado e o quisermos colocar a circular a 15 km/h, o erro da velocidade vai ser grande o que vai fazer com que o controlador desloque o peso para uma das extremidades da calha linear, isto mudará o centro de massa do sistema e fará o *Segway* acelerar. No entanto, mesmo que a velocidade desejada ainda não tenha sido atingida, se o *Segway* estiver na iminência de cair, ou seja, quando o ângulo atingir um determinado valor, o peso é enviado para trás, para que o *Segway* restabeleça a sua posição vertical, voltando de seguida a mandar o peso para a frente, adicionando mais aceleração e aumentando assim, gradualmente, a velocidade com impulsos de aceleração.

Na Figura 4-6 em baixo, pode ser visto a superfície *Fuzzy* gerada, que relaciona as entradas (erro da velocidade e ângulo) com a saída (posição do peso na calha linear).

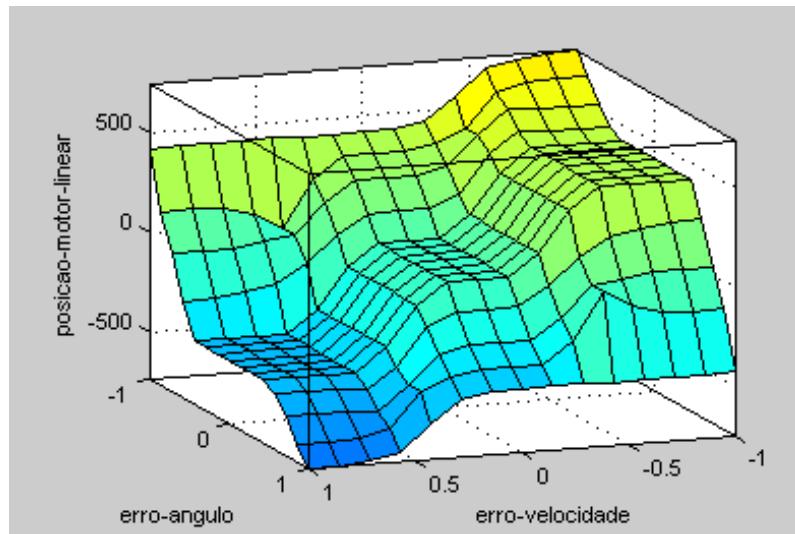


Figura 4-6 - Superfície *Fuzzy* responsável pelo controlo de velocidade do robô.

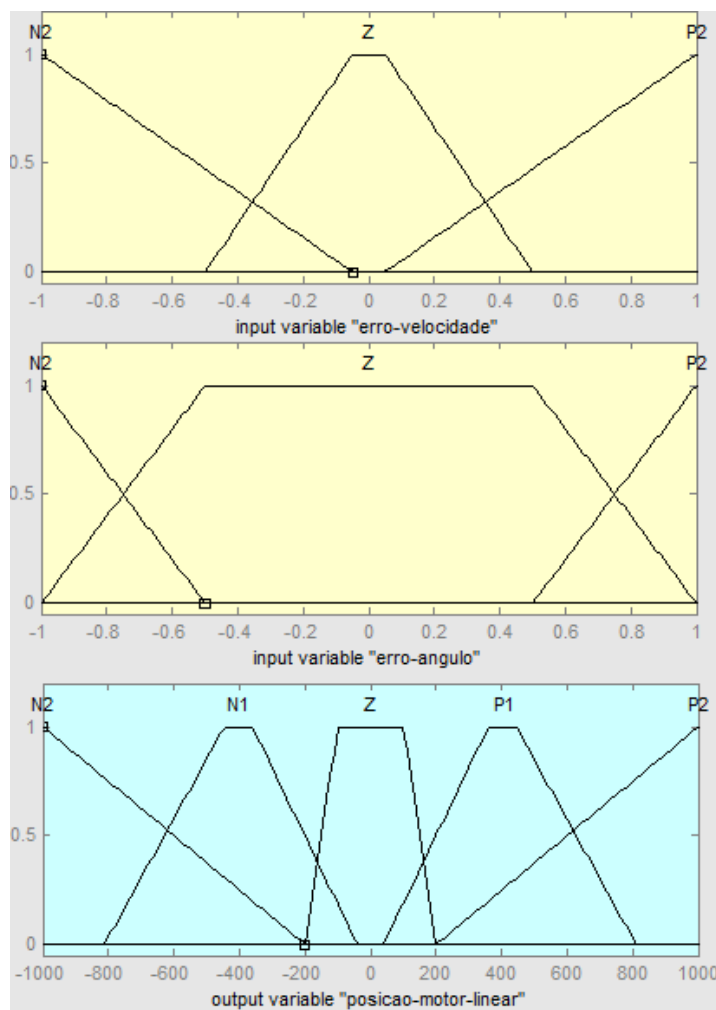


Figura 4-7 - Funções de pertença das entradas e da saída do controlador *Fuzzy* responsável pelo controlo de velocidade do robô

Como se pode ver na superfície obtida e pelas funções de pertinência ilustradas na Figura 4-7, definiu-se um intervalo considerável onde a variação do ângulo não causa qualquer influência no controlo de velocidade. Os valores dos parâmetros de entrada encontram-se normalizados, para apresentarem uma variação entre -1 e 1. Para o ângulo o valor 1 corresponde a 30 graus e para a velocidade 20 km/h. Fazendo esta correspondência, pode-se ver que o controlador apenas começa a considerar o valor do ângulo, quando este ultrapassa os 15 graus, uma vez que a esta inclinação a estabilidade do veículo começa a ficar comprometida. É como se o objectivo do controlador fosse alterado. Quando o ângulo ultrapassa os 15 graus, o controlador deixa de actuar sob o centro de massa do veículo não com o objectivo de variar a velocidade do mesmo, mas sim com o objectivo de restabelece-lo à posição vertical (ângulo zero).

O controlador de velocidade desenvolvido foi implementado no simulador do robô, da mesma forma que o controlador responsável por manter o robô equilibrado em duas rodas, analisado anteriormente, exportando os dados de controlo do Matlab para um ficheiro de texto, sendo posteriormente importado sob o formato de uma matriz no SimTwo. Apresenta-se, de seguida, o excerto de código responsável pela execução do controlador no SimTwo.

```

procedure LinearMotor_Control;
begin
    erro_teta := -i_teta;
    erro_v := v_ref - vel;
    erro_teta := sat(erro_teta/30 ,1); //normalização do ângulo (entre -1 e 1).
    erro_v := sat(erro_v/20 ,1); //normalização da velocidade (entre -1 e 1).
    Vi := 0;
    Step := 2/14 := 0.1429; //intervalo entre os valores -1 a 1 na tabela.
    Row := Round(erro_teta/Step)+7; //ângulo que agora varia entre 0 e 14.
    Col := Round(erro_v/Step)+7; //erro da velocidade a variar entre 0 e 14.
    Vi := Mgetv(LinearMotorControl, Row, Col); //saída do controlador.
end;

```

O controlo de velocidade foi implementado e testado com sucesso no simulador, tendo-se verificado que a abordagem escolhida, que toma em consideração também o ângulo do Segway no controlo de velocidade, foi a correcta. A monitorização do ângulo, revelou-se mesmo de extrema importância, tendo-se verificado que se o ganho do ângulo não estivesse bem ajustado, o peso permaneceria sempre num extremo da calha causando rapidamente a queda do robô.

4.4 - Implementação do Controlador de Velocidade no robô real

Após ter sido testado e aperfeiçoado no simulador do robô, o controlador *Fuzzy* responsável pelo controlo de velocidade, foi implementado no robô real. Para implementar o controlador de velocidade, foi primeiro necessário colocar o sistema que controla a posição do peso na calha a funcionar correctamente. Como tal, será aqui apresentado o código desenvolvido para o controlo de posição do motor de passo. Seguidamente será mostrado como o controlador *Fuzzy* desenvolvido no Matlab, pode ser incorporado e utilizado no Microcontrolador PIC32. Com o robô já em movimento, torna-se importante que o mesmo tenha a capacidade de evitar obstáculos. Assim, será também mostrado o código desenvolvido no PIC32, para colocar os sensores de ultra-sons a funcionar correctamente. Por fim, será apresentada a arquitectura geral utilizada para a execução de todo o código no PIC32, que garantiu que todos os subsistemas, que o mesmo tem de controlar, funcionassem correctamente.

4.4.1 - Controlo do motor passo a passo

A calha está ligada a um motor de passo que é controlado pelo driver SMC42 da Nanotec. O driver em questão permite o controlo do motor com um sinal de relógio, onde cada impulso corresponde a um passo dado pelo motor, assim como um sinal lógico para controlar a direcção de rotação. Na Figura 4-8 em baixo, é mostrado um excerto da *datasheet* do controlador, que indica os pinos disponíveis.

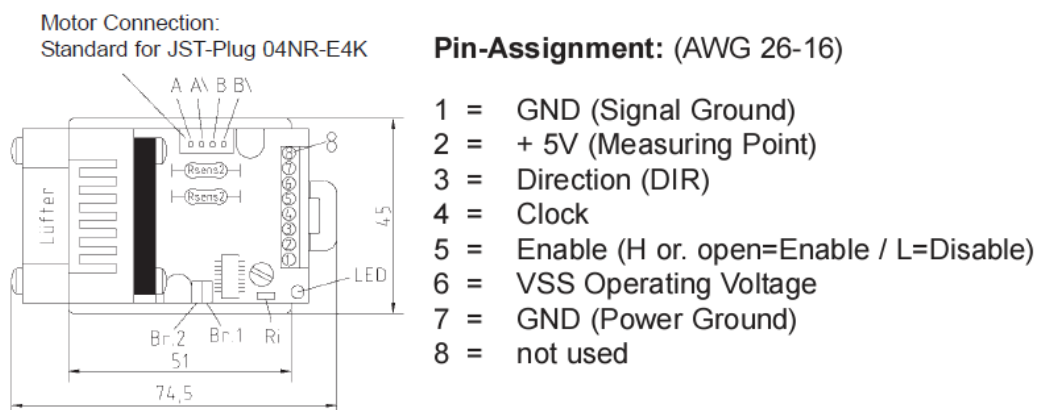


Figura 4-8 - Ligações eléctricas do controlador do motor passo a passo SMC42.

Foi necessário implementar no PIC32, uma função para gerar o sinal de relógio e o sinal da direcção, consoante a posição e a velocidade pretendidas para o motor. Também foi importante desenvolver uma função para controlar a aceleração do motor, para minimizar os

esforços mecânicos na calha e no peso a transportar. De seguida apresentam-se as funções implementadas, devidamente comentadas para facilidade de leitura.

```

void GoToPos(int Pd, float v) //desloca o peso para a posição Pd a uma velocidade v.
{
    int dir;

    Passos = abs(Pa - Pd); //Passos_necessários = abs(Posição_actual - Posição_destino).

    if(Pa < Pd) //Posição_actual < Posição_destino.
    {
        dir = 1;
        mPORTCClearBits(BIT_4); //coloca o sinal lógico da direcção a zero.
    }
    else
    {
        dir = -1;
        mPORTCSetBits(BIT_4); //coloca o sinal lógico da direcção a um.
    }

    if(Passos != 0)
    {
        if(InterruptDelay10us(-2.8*v+83)) //relaciona a velocidade pretendida com a
        { //largura do impulso do relógio.
            mPORTDToggleBits(BIT_0); //faz toggle ao sinal do relógio.
            toggles++; //conta os toggles gerados, 2 toggles
            //criam um impulso que faz o motor
            //andar um passo.
        }
        if(toggles >= 2) //após a geração de um impulso incrementa a posição actual.
        {
            Pa = Pa + dir; //incrementa ou decrementa a posição actual
            //consoante o sentido de rotação do motor.
            toggles = 0;
        }
    }
}

void AccelerationControl(int a_max) //quanto maior a_max mais gradual é a aceleração.
{
    if(Passos < a_max)
        a--; //se o motor está quase a chegar ao destino final começa a desacelerar.
    else
        if(a < a_max)
            a++; //senão começa a acelerar gradualmente até ao valor máximo definido.

    if(a < 1) //saturação do valor de a.
        a = 1;
    if(a > a_max)
        a = a_max;

    speed = v_max*a/a_max; //velocidade a aplicar na função GoToPos(int Pd, float v),
    //v_max é a velocidade máxima que se pretende que o
    //motor atinja, o valor desta variável é definido
    //no main().
}

```

A função *GoToPos(int Pd, float v)* faz deslocar o peso para a posição *Pd* a uma velocidade *v*. A posição *Pd* (Posição destino) pode ser qualquer valor entre -500 e 500. A posição 0 é o centro da calha, centro este definido no arranque do programa. As posições -500 e 500 são os extremos da calha. As funções *GoToPos(int Pd, float v)* e *AccelerationControl(int a_max)* são executadas sequencialmente no mesmo ciclo, que ocorre a uma frequência fixa de 5 kHz, através da utilização de interrupções com temporizadores:

```
//Timer4 interrupt handler.  
void __ISR(_TIMER_4_VECTOR, ipl2) Timer4Handler(void) //executado a 5 kHz.  
{  
    GoToPos(Pd, speed); //envia o peso para a posição Pd.  
    AccelerationControl(400); //aumenta e diminui gradualmente o speed do motor.  
}
```

4.4.2 - Importação do controlador Fuzzy desenvolvido para o PIC32

Como já foi visto, um controlador *Fuzzy* desenvolvido em Matlab pode ser exportado para um ficheiro de texto em formato de tabela. Viu-se ainda que a importação deste ficheiro de texto no SimTwo é simples, devido ao facto de o programa já disponibilizar uma função que lê directamente os dados contidos no ficheiro de texto e os transfere para uma Matriz. No entanto, no PIC32 não temos disponível essa função. A passagem dos dados contidos no ficheiro de texto (15x15 = 225 dados) para uma matriz em código C manualmente, consumiria muito tempo. Como tal utilizou-se um programa desenvolvido em C, que faz a leitura do ficheiro de texto e o transforma no formato de inicialização de uma matriz em C (`double Matriz[15][15] = {{dado1},{dado2},.....{dado225}};`). O programa em questão, foi desenvolvido e cedido pelo Estudante José Xavier [25], que desenvolveu um programa com o mesmo objectivo aqui pretendido. Para utilizar o programa, basta colocar o ficheiro de texto exportado do Matlab, na pasta onde se encontra o executável do programa (`matlab2matriz.exe`). O nome do ficheiro de texto deve ser alterado para "LookUpTable". De seguida basta executar o ficheiro "matlab2matriz.exe" na linha de comandos. Depois de localizar correctamente o directório de origem do executável, na linha de comandos, deve-se executar o programa com os seguintes argumentos de entrada:

```
matlab2matriz.exe nome_ficheiro_saida 15 15
```

criando-se assim um ficheiro de texto, com o nome escolhido, no directório do programa. O ficheiro de texto criado contém o controlador *Fuzzy* no formato de uma matriz em código C (`double FuzzyTable[15][15] = {{dado1},{dado2},.....{dado225}};`), sendo apenas necessário copiar o texto e acrescentá-lo no programa. Consegue-se assim, uma forma simples e rápida importar um controlador *Fuzzy*, desenvolvido no Matlab, em qualquer Microcontrolador programado em código C.

Com a matriz de controlo inicializada, desenvolveu-se uma função semelhante à desenvolvida no simulador, para executar o controlador. Foi também necessária a implementação de uma função para arredondar valores decimais, uma vez que foi utilizado o método de arredondamento para os parâmetros de entrada não disponíveis na tabela de correspondência *Fuzzy*. De seguida apresenta-se a implementação da função de arredondamento e da função que executa o controlador de velocidade.

```

int Round(float x) //arredonda valor decimal, para inteiro mais próximo.
{
    int y;
    float r;
    y = x;
    r = (x/10);

    if((r >= 0.5) && (y > 0))
        return (y+1);
    else
        if(r < 0.5 && y > 0)
            return y;
        else
            if(r <= -0.5 && y < 0)
                return (y-1);
            else
                if(r > -0.5 && y < 0)
                    return y;

    return 0;
}

void FuzzyControl(int teta_med, int vel_med) //executa controlador Fuzzy.
{
    float Step = 0.1429;

    double erro_vel, erro_teta;
    erro_teta = -teta_med;
    erro_vel = vel_ref - vel_med;

    erro_teta = Ka*erro_teta/30; //normalização dos parâmetros de
    erro_vel = Kv*erro_vel/20; //entrada do controlador Fuzzy.

    if(erro_teta > 1)
        erro_teta = 1;
    if(erro_teta < -1)
        erro_teta = -1;
    if(erro_vel > 1)
        erro_vel = 1;
    if(erro_vel < -1)
        erro_vel = -1;

    Row = Round(erro_teta/Step)+7;
    Col = Round(erro_vel/Step)+7;

    Pd = (int)FuzzyTable[Row][Col]; //Saída do controlador. Obtêm-se assim a Pd
    //(posição_destino) a utilizar na função
    //GoToPos(Pd, speed) que controla a posição do peso
}

```

4.4.3 - Implementação de sistema baseado em ultra-sons para evitar obstáculos

Um sistema baseado em ultra-sons para detectar e evitar obstáculos é muito importante num robô controlado à distância pela Internet. Primeiro porque pode ajudar o utilizador a controlar o robô, em espaços mais apertados, onde só as imagens da câmara não são suficientes. Por outro lado, permite evitar colisões no caso de haver alguma falha ou atraso nas comunicações. O sensor utilizado foi o HC-SR04 mostrado na Figura 4-9. Como se pode ver pela Figura, existem 4 pinos disponíveis, dois são para a alimentação (“VCC e GND”), um para o sinal de disparo (“Trig”), onde é possível emitir o ultra-som e um para a obtenção do eco captado pelo receptor (“Echo”).

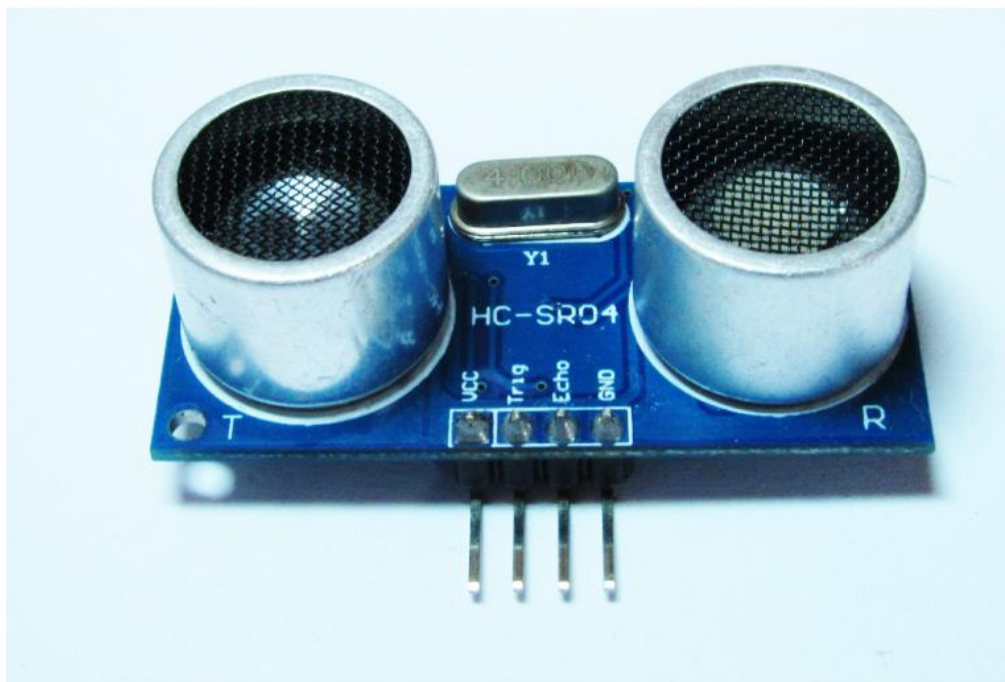


Figura 4-9 - Ultra-sons HC-SR04.

Aplicando um sinal com pelo menos $10\mu\text{s}$ no comando de disparo são gerados, automaticamente, 8 impulsos de 40kHz. O sinal gerado, ao embater num obstáculo, retorna como eco que, ao ser detectado pelo receptor, faz com que surja um sinal no pino “Echo”. O tempo que o sinal permanece ao nível alto é proporcional à distância entre o objecto detectado e o sensor. Na Figura 4-10 é possível ver o diagrama temporal, extraído da *datasheet* do sensor, que ilustra o descrito. A distância ao obstáculo pode ser calculada da seguinte forma:

$$Dist = \frac{velocidade_{som} \times tempo_{EcoNivelAlto}}{2} \quad (4.1)$$

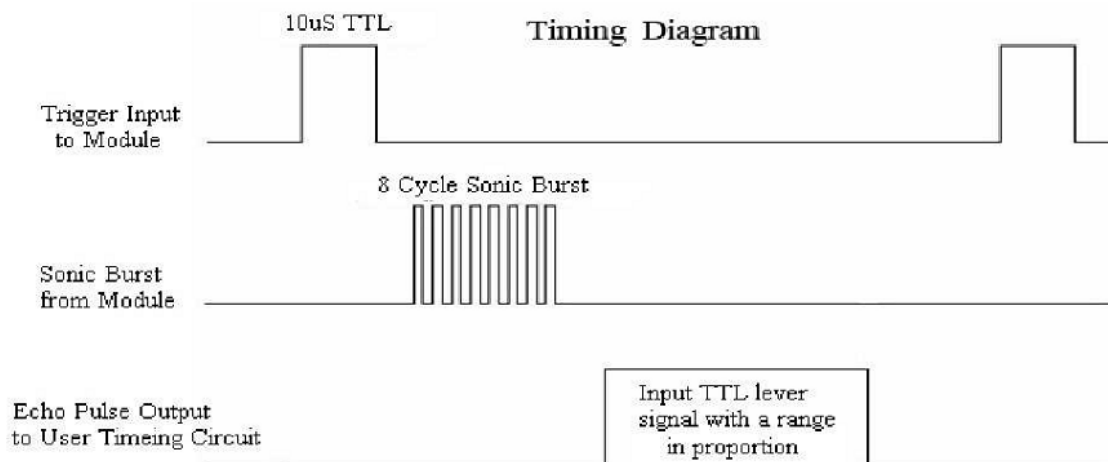


Figura 4-10 - Diagrama temporal de funcionamento do HC-SR04.

Através da expressão 4.1 é possível calcular com uma precisão de 5mm, distâncias entre 2 a 400cm. No entanto, para o conseguir é necessário medir com precisão o tempo que o pino “Echo” permanece ao nível alto. O PIC32 disponibiliza algumas entradas, designadas por “*Input Capture*”, que permitem a medição, por hardware, do tempo que um sinal permanece ao nível alto. No entanto, para um menor tempo de desenvolvimento da aplicação dos ultras-sons no robô, optou-se por uma abordagem diferente, possibilitando uma implementação mais rápida. O objectivo dos ultra-sons instalados no robô, não é a medição precisa da distância a que um obstáculo se encontra do mesmo. Deseja-se sim saber se existe um obstáculo próximo do sensor ou não. Como tal, aplicou-se um filtro passa-baixo ao sinal do eco. O sinal que surge como eco pode ser visto como um sinal de PWM, cujo *duty cycle* varia proporcionalmente com a distância ao obstáculo. Assim, aplicando um filtro passa-baixo, obtêm-se uma tensão contínua que varia, entre 0 e 5V, consoante a distância. Essa tensão pode depois ser lida utilizando um dos ADCs do PIC32, fornecendo assim uma medida de distância, o que permite detectar eventuais obstáculos.

O controlador desenvolvido no PIC32 para detecção de obstáculos foi o seguinte: se for detectado um obstáculo, pelos dois sensores, as ordens de controlo do operador são ignoradas e o robô recua; se for detectado obstáculo apenas num dos sensores o robô vira no sentido que permita o afastamento desse obstáculo.

4.4.4 - Arquitectura de todo o código desenvolvido no PIC32

O Microcontrolador PIC32 executa várias tarefas em simultâneo no robô. Foi a única unidade de processamento utilizada e ao seu cargo tem:

- Aquisição e filtragem de sinais eléctricos (ângulo, velocidade, nível bateria, etc);

- Controlo de posição, velocidade e aceleração do motor passo a passo;
- Operar como servidor da página Web que permite o controlo do robô pela Internet;
- Gerar os sinais de controlo para os ultra-sons e detectar através da medição do eco, se existe ou não um obstáculo à frente do robô;
- Gerar PWM para o controlo de viragem do robô;
- Executar o controlador *Fuzzy*, que permite o controlo de velocidade do robô.

A maioria destes subsistemas foram, inicialmente, implementados e testados isoladamente, pelo que, quando se colocou tudo a funcionar em simultâneo, se constatou que certos subsistemas não funcionavam correctamente. Verificaram-se principalmente problemas no controlo do motor de passo. Também ocorriam frequentemente atrasos, ou mesmo encravamento, da página Web de controlo do robô. Ainda, os sensores de ultra-som ocasionalmente deixavam de funcionar.

Devido às dificuldades encontradas inicialmente para colocar todos os subsistemas a funcionar correctamente, resolveu-se reestruturar todo o código desenvolvido. Assim pensou-se numa arquitectura baseada essencialmente em interrupções, com diferentes níveis de prioridade para permitir o processamento paralelo de todas as tarefas necessárias. Removeram-se ainda todos os atrasos existentes, substituindo-os por temporizadores baseados em interrupções.

A Figura 4-11 apresenta o algoritmo de controlo utilizado para que todo o código desenvolvido no PIC32 funcionasse correctamente.

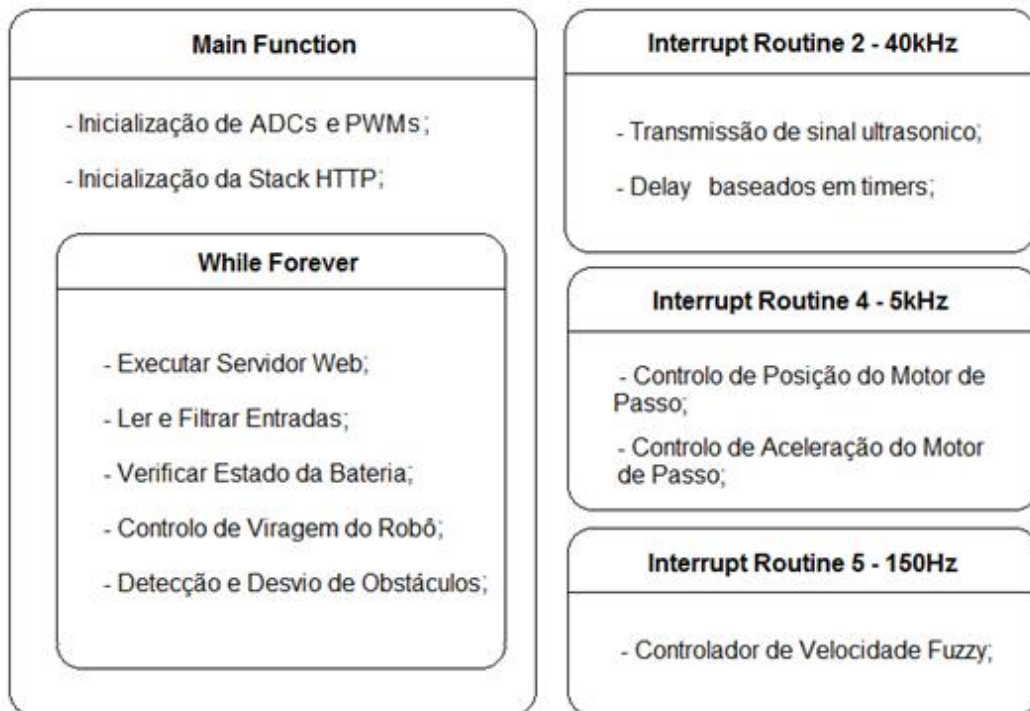


Figura 4-11 - Algoritmo de Controlo para o correcto funcionamento do robô Vigilante.

4.5 - Sumário e Principais Conclusões

Este capítulo apresentou a adaptação de um veículo com auto-equilíbrio em duas rodas, numa plataforma móvel robótica que pode ser controlada à distância.

Foi desenvolvido um simulador tridimensional e realista do robô, de forma a permitir testar o conceito de manipulação do seu centro de massa, através do deslocamento linear de um peso na sua base.

Foi implementado, e testado com sucesso no simulador, um controlador *Fuzzy* da velocidade do robô. O controlador desenvolvido, para além do erro da velocidade, toma em consideração também o ângulo que o mesmo faz com a vertical. O controlador actua sobre o robô, modificando o seu centro massa, conseguindo assim variar a sua velocidade. No entanto, o facto de actuar sob o centro de massa pode também pôr em risco a sua estabilidade, pelo que, para o desenvolvimento de um controlador de velocidade, através da estratégia aqui adoptada, torna-se importante a monitorização do ângulo que o veículo auto-equilibrado faz com a vertical. A importância da monitorização do ângulo para o controlador de velocidade desenvolvido pôde ser confirmada, nos testes efectuados no simulador, verificando-se que quando o ângulo não era correctamente obtido pelo controlador, devido a um incorrecto ajuste dos ganhos, o robô acabava sempre por perder o equilíbrio e cair.

Após ter sido comprovada em simulação a estratégia de controlo adoptada, e após o seu correcto ajuste, procedeu-se à implementação do controlador de velocidade no robô real.

Antes da implementação deste controlador, no PIC32, foi necessário o desenvolvimento do código que permitiu controlar o motor de passo. Desenvolveram-se funções para controlar a posição, a velocidade e a aceleração do motor de passo.

Apesar das limitações temporais associadas ao teste e aperfeiçoamento do controlo de velocidade implementado, pôde verificar-se que o mesmo estava a funcionar de acordo com o esperado e obtido na simulação. No entanto, o controlo estava um pouco instável, precisando ainda de alguns ajustes nos ganhos ou até mesmo no próprio controlador.

Relativamente ao sistema desenvolvido para evitar obstáculos o mesmo precisa ainda de alguns ajustes. Apesar de se ter apurado que, com o controlador desenvolvido, o robô conseguia desviar-se de alguns obstáculos, isso nem sempre acontecia. Assim é necessário melhorar a robustez deste controlador, com a introdução de mais sensores, ou mesmo com a implementação de um controlador mais robusto, que possa perceber em que ambiente o robô está e como deve actuar para evitar os obstáculos. Devido ao facto de o sistema de desvio de obstáculos implementado, não ter ficado robusto, acabou por se fazer algo mais simples, utilizando os sensores apenas para mandar parar o robô na presença de um obstáculo.

Capítulo 5

Controlo Remoto do Robô recorrendo a Comunicações Wireless

5.1 - Introdução

Pretende-se que o robô desenvolvido possa ser controlado remotamente, em todo o espaço interior e exterior da Faculdade, podendo assim efectuar missões de vigilância através da captação de imagens e som em tempo real. Uma vez que a Faculdade está coberta pela rede Wireless, em quase toda a sua extensão, a sua utilização torna-se um bom recurso para o controlo do robô de vigilância pretendido.

Assim, apresenta-se neste capítulo a operação do robô à distância, recorrendo à rede Wireless já existente e a um Microcontrolador PIC32, com interface Ethernet, uma Ponte Wireless e uma câmara IP. Como sinais de realimentação, associados ao controlo, usam-se imagens e som em tempo real. Mostra-se ainda a página Web desenvolvida, que permite ao utilizador visualizar as imagens captadas e enviar as ordens de controlo a partir de qualquer computador ligado à mesma rede que o robô.

5.2 - Comunicações Ethernet com o PIC32

O PIC32 Ethernet Starter Kit adquirido possibilita o alojamento de uma página Web na sua memória, funcionando como servidor da mesma. A informação de controlo passada através da página Web é processada directamente no PIC32, que executa no robô as ordens recebidas. A Microchip disponibiliza uma Stack TCP/IP implementada em código C, que pode ser executada no PIC32, assim como um conjunto de exemplos com funções já implementadas para os

diferentes protocolos de comunicação suportados. Como se pode ver na Figura 5-1, a Stack é dividida em múltiplas camadas a partir dos vários protocolos suportados.

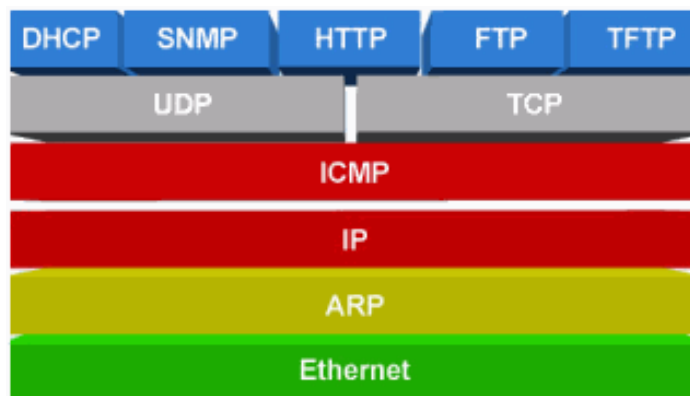


Figura 5-1 - Camadas da Stack TCP/IP desenvolvida pela Microchip para o PIC32 [26].

Para o controlo do robô Vigilante foi utilizado o protocolo de alto nível HTTP (*HyperText Transfer Protocol*). O HTTP é o protocolo utilizado para a troca de dados entre um *browser* e um servidor Web. Existem dois métodos para a troca de dados: o método GET e o método POST. O PIC32 suporta ambos os métodos, no entanto o método GET é mais fácil de processar, uma vez que a informação é recebida toda de uma vez. Com a informação toda em memória fica mais simples e rápida a procura dos dados recebidos. No entanto, a quantidade de dados que podem ser enviados utilizando este método, está limitada geralmente aos 100 bytes. O método POST tem uma implementação mais complexa e exige mais responsabilidade para o Microcontrolador. No entanto não tem limite para a transferência de dados.

Para controlar o robô foram utilizadas duas variáveis, X e Y, que permitem indicar o movimento pretendido, como se pode ver na tabela abaixo.

Tabela 5-1 - Atribuição de valores às variáveis de controlo do robô Vigilante.

		X		
		-1	0	1
Y	1	Esquerda e para Frente	Anda para Frente	Direita e para Frente
	0	Vira à Esquerda	Parado	Vira à Direita
	-1	Esquerda e para Trás	Anda para Trás	Direita e para Trás

Uma vez que não é necessária a transferência de grandes quantidades de dados, para o controlo do robô, e por ser mais simples e eficiente de executar no PIC32, utilizou-se o método GET para a troca de dados. No método GET os dados são passados juntamente com o endereço URL. Os mesmos são separados do endereço através da utilização de um ponto de interrogação (?). Assim, para, por exemplo, enviar a ordem de controlo para o robô andar

para frente e virar à esquerda é submetido o seguinte endereço URL: (http://segway/index.html?X=-1&Y=1). O método GET foi também utilizado para enviar o valor de velocidade pretendida para o robô, assim como para permitir a calibração do ângulo e o ajuste online dos parâmetros do controlador. Cada vez que é feito um pedido de submissão de dados no *browser*, a função HTTPExecuteGet() é executada no PIC32, podendo aí serem executadas rotinas para procurar as variáveis de interesse nos dados recebidos.

Para além do envio de dados de controlo para o robô, pretendia-se também o envio de dados do robô para o *browser*, para monitorizar parâmetros como a velocidade actual do robô, o ângulo que o mesmo faz com a vertical e o estado actual da bateria. Para tal utilizaram-se variáveis dinâmicas que são introduzidas directamente no código html da página Web. A variável dinâmica deve ser colocada entre tiles (~variavel~) no código html onde se pretenda que o seu valor apareça. Na Figura 5-2 é possível ver o exemplo de uma página desenvolvida para mostrar a velocidade, o ângulo e o estado da bateria do robô.

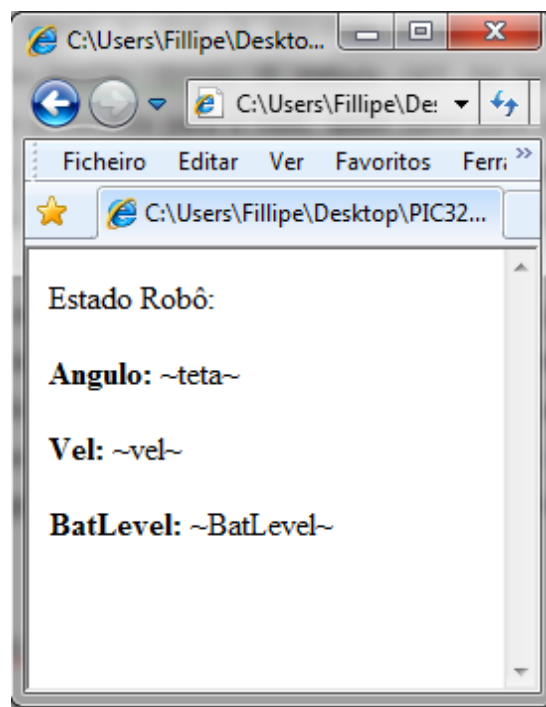


Figura 5-2 - Exemplo de utilização de variáveis dinâmicas.

Quando a página Web é executada no PIC32, as variáveis dinâmicas (~variavel~) são substituídas pelo valor atribuído à variável no PIC32. Para, por exemplo, atribuir o valor à variável dinâmica ~varName~ no PIC32 é necessário implementar a função HTTPPrint_varName(void). A função é executada sempre que o nome da variável é encontrado entre tiles no código html da página Web.

De seguida apresenta-se a implementação da função que atribui o valor da variável dinâmica ~vel~, que indica a velocidade do robô.

```

void HTTPPrint_vel(void)
{
    char velString[150];
    itoa2(vel, velString, 10); //passa variável inteira para uma string
    TCPPutString(sktHTTP, velString);
}

```

Para treino da utilização expedita da Stack TCP/IP e das funções já disponíveis, começou-se por compilar e executar um código de exemplo disponibilizado no site da Microchip ('TCPIP Demo App'). Através da manipulação desse código, foi possível entender como o mesmo funcionava, sendo depois alterado para conter a página de controlo desenvolvida para o robô.

5.3 - Comunicações Wireless

Uma vez que o PIC32 comunica usando uma interface Ethernet, utilizou-se uma Ponte Wireless (ver Figura 3-11) para estabelecer a ligação entre o PIC32 e a rede Wireless existente na Faculdade.

A Ponte Wireless adquirida permite dois modos de funcionamento: Ponte Wireless ou Ponto de Acesso. Devido ao facto de o equipamento também operar como Ponto de Acesso, podemos também chamá-lo de *router*. Para testar as comunicações entre o PC e o PIC32, o *router* foi inicialmente colocado a funcionar como Ponto de Acesso. Desta forma, foi possível efectuar a ligação sem fios entre o PC e o *router*, que, por sua vez se encontrava ligado por cabo Ethernet ao PIC32 e à câmara IP. Com o computador ligado directamente ao *router*, foi possível configurar correctamente a câmara IP e o PIC32, assim como desenvolver e testar a página Web de controlo do robô. No entanto, o funcionamento do *router* como Ponto de Acesso permite apenas o controlo remoto do robô a curtas distâncias (o *router* tem um alcance de $\approx 300\text{m}$).

Com o robô já a funcionar como se pretendia, o passo seguinte foi a correcta configuração do *router* para que o mesmo pudesse operar como Ponte Wireless e, assim, ligar-se à rede eduroam da Faculdade. A rede eduroam é uma infra-estrutura de acesso wireless com elevados patamares de segurança. O SSID da rede é escondido e a sua autenticação é baseada no sistema WPA-Enterprise, sistema normalmente utilizado em grandes empresas. O *router* adquirido foi desenvolvido para uso doméstico, verificando-se que não suportava o sistema WPA-Enterprise de autenticação. Recorreu-se à ajuda da Unidade de Infra-estruturas e Redes de Comunicações da FEUP, mais concretamente do Técnico Superior Fernando Romão, para ajudar a resolver o problema. A sua sugestão foi que a ligação fosse feita utilizando a rede eduroam-guest. A rede guest, não precisa de autenticação, cobre toda a Faculdade e permite uma ligação mais rápida na passagem entre pontos de acesso, uma vez que não é necessário efectuar a autenticação. Utilizando

a rede guest foi possível ligar o robô à rede da Faculdade e controlá-lo a partir de um PC também ligado à rede guest. Apesar de a Ponte Wireless adquirida conseguir ligar-se à rede eduroam-guest, verificou-se que a ligação tinha muitos problemas de conectividade. Analisado o problema verificou-se que tal se devia ao facto de existirem, na zona onde o robô foi testado outras redes desconhecidas pelos serviços técnicos. O facto de existirem, em salas ou laboratórios, *routers* a propagar sinal no mesmo canal que a rede da Faculdade, faz com que a Ponte Wireless adquirida não consiga estabelecer uma ligação estável, devido à sobreposição desta rede com outros sinais que apresentam uma potência de sinal maior, por os seus pontos de acesso estarem mais próximos.

O controlo do robô utilizando a rede guest, foi testado em diferentes zonas da FEUP (Edifício B, Edifício I, Zona Exterior em Frente à Biblioteca), de forma a tentar perceber se o problema estava apenas restrito ao Edifício I. Pôde-se verificar que apenas era possível estabelecer uma ligação estável, que permitisse controlar correctamente o robô pela rede guest, em espaços onde apenas existisse o sinal da rede eduroam e da rede eduroam-guest. Assim, como apenas em determinadas zonas do Edifício B e do Edifício I isso acontece, concluímos que o controlo do robô através da rede guest, utilizando o *router* adquirido, teria as limitações apontadas.

5.4 - Desenvolvimento de uma página Web para Controlo do robô e Visualização de Imagem

A página Web pretendida deveria possibilitar o envio de comandos para a movimentação do robô a partir do teclado do computador, utilizando o método GET para o envio dos dados. Pretendia-se também fazer o *streaming* do vídeo captado pela câmara IP na página de controlo do robô.

Nesta etapa do projecto, recorreu-se à ajuda do colega Bruno Moreira para o desenvolvimento da página Web final, devido à sua experiência em *Web Design*. A página Web foi desenvolvida utilizando *javascript* e pode ser vista na Figura 5-3. Ao centro da página encontra-se o vídeo captado pela câmara IP. Devido ao facto de o *plugin* utilizado pela câmara IP para mostrar o vídeo, apenas estar disponível para o Internet Explorer, a página de controlo desenvolvida apenas funciona correctamente neste *browser*. É possível controlar o robô a partir de qualquer outro browser, no entanto o vídeo captado apenas é mostrado no Internet Explorer.

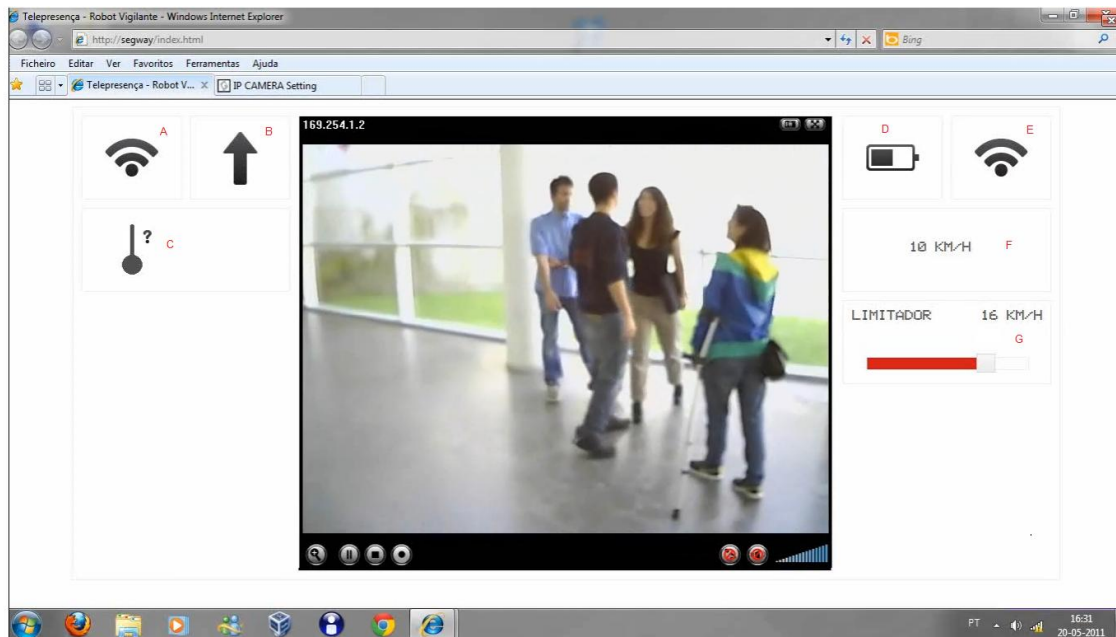


Figura 5-3 - Pagina Web de controlo do robô Vigilante (<http://segway/index.html>).

Apresentam-se, de seguida, as funcionalidades associadas à página (representadas pelas letras A a G, a vermelho na Figura 5.3):

- A e E - Estas secções foram desenvolvidas com o objectivo de auxiliar o operador no controlo do robô em espaços apertados. O símbolo mostrado em A aparece apenas quando é detectado um obstáculo pelo sensor de ultra-sons instalado do lado esquerdo do robô. O mesmo acontece para o símbolo E, para obstáculos que surgissem do lado direito. Por limitações temporais, esta função não está ainda implementada
- B - O robô é controlado pelas setas disponíveis no teclado do computador, com um click numa das setas, surge neste espaço a imagem da seta premida, de forma a dar *feedback* ao utilizador de que a ordem foi recebida pelo *browser*. Quando nenhuma tecla é premida nada é mostrado, ficando em branco;
- C - Este espaço está destinado para informar o utilizador do ângulo de inclinação do robô. Existe a representação de um *Segway* visto de perfil e na posição vertical. A ideia é alterar a inclinação do mesmo consoante a inclinação real do robô, assim como a indicação numérica do valor do ângulo. Por limitações temporais, esta função não está ainda implementada
- D - Aqui é possível ver o estado actual da bateria do robô, através de um indicador de três níveis (cheio, metade, vazio);
- F - Nesta secção é mostrada a velocidade actual do robô;
- G - Esta parte permite seleccionar a velocidade máxima pretendida para o robô a partir de um *slide*.

Desenvolveu-se também uma página (Figura 5-4) para, no início do programa, permitir a calibração do ângulo zero (posição vertical do robô) e para permitir o reajuste *online* dos ganhos do controlador *Fuzzy* desenvolvido (K_a - ganho do ângulo; K_v - ganho da velocidade; K_p - ganho da saída do controlador).

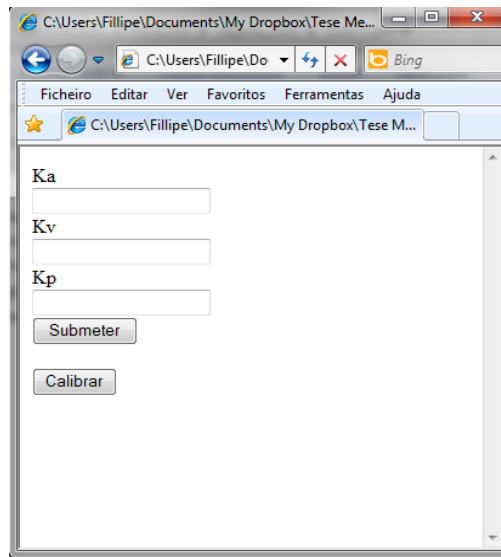


Figura 5-4 - Página para calibração do ângulo e ajuste dos ganhos do controlador Fuzzy (<http://segway/parametros.html>).

Para carregar as páginas desenvolvidas para o PIC32 é necessário convertê-las para código C, para poder depois ser compilado. A Microchip disponibiliza a aplicação MPFS Generator (Figura 5-5) que permite fazer essa conversão.

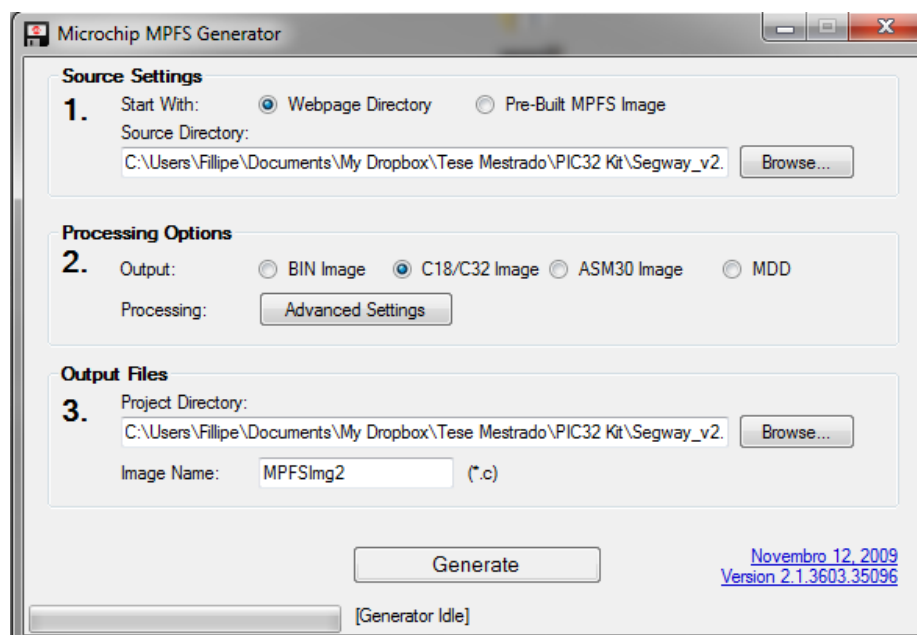


Figura 5-5 - Programa da Microchip que permite converter as páginas Web em código C.

5.5 - Sumário e Principais Conclusões

Mostrou-se, neste capítulo, o controlo remoto do robô através de uma rede wireless. Para tal utilizou-se um Microcontrolador e uma Ponte Wireless. O PIC32 e a câmara IP são ligados por cabo Ethernet à Ponte Wireless que, por sua vez, se liga à rede wireless existente. O PIC32 Ethernet Starter Kit possui todo o hardware necessário para ligações Ethernet. A Microchip fornece ainda um conjunto de aplicações e exemplos que ajudam a implementação de um servidor Web no Microcontrolador PIC32.

Foi visto que a Ponte Wireless adquirida (Figura 3-11 - Wireless Bridge D-Link DAP-1522 [22].) não suporta autenticação WPA-Enterprise, não sendo assim possível a ligação à rede eduroam. Foi assim utilizada a rede eduroam-guest para controlar o robô, constatando-se que a sua utilização tem mais vantagens para o controlo do robô. A rede guest não precisa de autenticação, podendo ligar-se à Ponte Wireless. O seu sinal está também propagado por toda a Faculdade e permite uma ligação mais rápida na passagem entre pontos de acesso, devido ao facto de não ser necessário efectuar a autenticação. Utilizando a rede guest foi possível ligar o robô à rede da Faculdade e controlá-lo a partir de um PC ligado à mesma rede. No entanto, surgiram também problemas na ligação entre a Ponte Wireless e a rede guest, verificando-se que a ligação estabelecida era muito instável em determinadas zonas. Contactados os serviços técnicos, concluiu-se que tal facto se deve à presença de outras redes a propagar sinal no mesmo canal que a rede da Faculdade. A compra de uma Ponte Wireless industrial ou, então, a utilização de um dos Pontos de Acesso utilizados pela Faculdade no robô, permitiria o estabelecimento de uma ligação mais estável.

Assim, a Ponte Wireless foi colocada a funcionar como Ponto de Acesso, o que permite que o robô seja controlado remotamente e com segurança até uma distância de $\approx 300\text{m}$.

Capítulo 6

Conclusões e Futuros Desenvolvimentos

6.1 - Conclusões gerais

O projecto aqui apresentado abordou o desenvolvimento de um robô de vigilância, de uma forma diferente, partindo de um veículo com auto-equilíbrio em duas rodas, semelhante a um Segway. Foram destacadas as vantagens da utilização de tal veículo como plataforma para robôs móveis, principalmente quando são robôs que devem interagir e circular em ambientes com pessoas. Isto porque um robô desenvolvido sob uma plataforma que tem a capacidade de se auto-equilibrar, pode usufruir de uma estatura semelhante à de um homem adulto e apresentar, mesmo assim, grande mobilidade quer em espaços interiores quer em exteriores.

O conceito de controlo de posição de um veículo semelhante a um Segway, através da alteração do seu centro de massa, foi testado em simulação e implementado com sucesso no robô real. Para alterar o centro de massa do veículo, foi utilizado um motor linear para fazer deslocar, de uma extremidade à outra da calha, uma massa de 4Kg.

Foi desenvolvido um controlador de velocidade, baseado em lógica difusa, para permitir que o operador do robô pudesse seleccionar a velocidade pretendida, sendo a alteração do centro de massa da responsabilidade do controlador desenvolvido. O controlador de velocidade foi testado com sucesso no simulador. No entanto, no robô real não apresentou o desempenho desejado. Inicialmente o controlador foi implementado sem o manequim, apresentando bons resultados. Com a introdução do manequim o sistema tornou-se mais instável e, como tal, mais difícil de controlar. Foram efectuadas as modificações necessárias, no controlador, na tentativa de melhorar o seu desempenho. No entanto, não foi possível dedicar o tempo necessário para colocar o controlador a funcionar como desejado. De referir que, mesmo sem o controlador de velocidade, é possível controlar correctamente o robô. O operador do robô pode ser o responsável por actuar remotamente sob o centro de massa do veículo. Em boa verdade é o que acontece quando um indivíduo se desloca em cima de um

Segway: o próprio indivíduo é o controlador de velocidade, actuando sob o centro de massa do sistema. O mesmo pode ser feito remotamente, actuando sob a massa de 4kg. Apesar de não ser necessário o controlador de velocidade é importante o seu desenvolvimento, para por exemplo, garantir que o robô fica parado, mesmo que o operador não esteja sob os seus comandos. Se o mesmo está programado para ficar parado, o sistema de controlo actua no sentido de o manter parado, mesmo que, por exemplo, alguém esteja a empurrar o robô. O controlador de velocidade também é importante no caso de se pretender colocar o robô a percorrer um determinado percurso autonomamente.

A última etapa deste projecto foi a implementação de um sistema, que permitiu o controlo remoto do robô utilizando a rede wireless da Faculdade, possibilitando obter imagens e som em tempo real.

Os objectivos foram cumpridos. No entanto pôde-se concluir que a Ponte Wireless adquirida, que permite a ligação do robô à rede wireless da Faculdade, não foi bem escolhida. Devido às características da rede na Faculdade, a ligação que é estabelecida entre a rede wireless e a Ponte Wireless adquirida é muito instável. A Ponte Wireless adquirida foi desenvolvida para uso doméstico e, segundo sugestão do Técnico Superior de Redes e Comunicações, Fernando Romão, dever-se-ia utilizar uma Ponte Wireless industrial, ou então usar um dos Pontos de Acesso da Faculdade para ser utilizado no robô. Isto garantiria uma ligação estável entre o robô e a rede.

Assim, podemos concluir que todos os objectivos traçados no início deste projecto foram cumpridos. Este projecto permitiu demonstrar, por um lado, as elevadas potencialidades da utilização de um veículo auto-equilibrado como plataforma robótica móvel e por outro, as potencialidades do Microcontrolador PIC32 para o desenvolvimento de aplicações Web.

6.2 - Futuros Desenvolvimentos

O facto de neste projecto se ter desenvolvido uma primeira versão do robô vigilante, implica que existam ainda inúmeras tarefas a serem realizadas e outras a serem melhoradas. Como tal, serão aqui feitas algumas sugestões para o melhoramento futuro do robô desenvolvido.

Uma vez que se verificou que a Ponte Wireless que foi adquirida, não foi a melhor para estabelecer a ligação entre o robô e a rede wireless da Faculdade, dever-se-iam encontrar alternativas, recorrendo a uma Ponte Wireless industrial ou pedindo à Faculdade um dos seus Pontos de Acesso.

Outro aspecto importante a melhorar, é o controlador de velocidade, que precisa de ser correctamente calibrado. Um aspecto que pode estar a causar alguma instabilidade ao controlador de velocidade, pode ser o facto de a velocidade estar a ser obtida de uma forma

estimada a partir da tensão aplicada aos motores. Verificou-se que se obtêm dados válidos para a velocidade pelo método adoptado. No entanto podem existir atrasos, ou desfasamentos que comprometam o desempenho do controlador. Como tal, sugere-se o desenvolvimento e a implementação de um *encoder* para as rodas do robô, de forma a obter melhores medidas para a velocidade.

Seria interessante também o desenvolvimento de uma estrutura mecânica à prova de água para colocar toda a electrónica, assim como uma estrutura mecânica para apoiar o robô e o impedir de cair, no caso de este ficar sem bateria.

Poder-se-iam ainda introduzir mais sensores de ultra-sons, a circundarem, o robô e desenvolver um controlador mais sofisticado que o actual, para permitir que o robô possa evitar eficazmente obstáculos.

Referências

- [1] H. Dreyfus, "Tele-presença Desincorporada e a Distância da Realidade". Disponível em <http://members.fortunecity.com/cibercultura/vol9/dreyfus.html>. Acedido em 09/11/2010.
- [2] Wikipedia, "Telepresence". Disponível em <http://en.wikipedia.org/wiki/Telepresence>. Acedido em 19/12/2010.
- [3] "Avatar robótico começa a chegar aos escritórios," *New Scientist*, 13/01/2011. Disponível em <http://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=robo-avatar-robotico&id=010180110113>. Acedido em 13/01/2011.
- [4] "Segway Simply Moving". Disponível em <http://www.segway.com/individual/models/i2.php>. Acedido em 08/06/2011.
- [5] "Segway RMP (Robotic Mobility Platform)." Disponível em <http://rmp.segway.com/>. Acedido em 08/06/2011.
- [6] "Science Fiction Meets Science Fact," 16/11/2009.
- [7] "Anybots QB Telepresence Robot Lets You Be At the Office ... Without Being There," 18/05/2010. Disponível em <http://spectrum.ieee.org/automaton/robotics/industrial-robots/051810-anybots-qb-new-telepresence-robot>. Acedido em 15/02/2011.
- [8] MundodasMarcas, "Segway", *Mundo das Marcas*, 5/5/2009. Disponível em <http://mundodasmarcas.blogspot.com/2008/06/segway.html>. Acedido em 14/02/2011.
- [9] "Marines Magazine". Disponível em <http://marinesmagazine.dodlive.mil/2010/10/21/rover-the-mobile-robotic-target-system/>. Acedido em 20/12/2010.

- [10] "MDS humanoid robot now available from Xitome", 21/06/2010. Disponível em <http://rmp.segway.com/rmp-200/>. Acedido em 02/02/2011.
- [11] "Tbot: The Self-Balancing Transformer Robot," 17/06/2010. Disponível em <http://www.shervinemami.co.cc/tbot.html>. Acedido em 18/12/2010.
- [12] "GM/Segway EN-V Project," 08/01/2010. Disponível em <http://www.flickr.com/photos/24641875@N07/sets/72157623526768197/>. Acedido em 16/02/2011.
- [13] "Willow Garage Creates Awesome Open Source Telepresence Robots," 04/02/2010. Disponível em <http://singularityhub.com/2010/02/04/willow-garage-creates-awesome-open-source-telepresence-robots-video/>. Acedido em 15/02/2011.
- [14] "Student Lyndon Baty Attends School In Texas Via Vgo Robot ". Disponível em http://www.huffingtonpost.com/2011/02/06/texas-student-lyndon-baty-vgo-robot_n_818884.html. Acedido em 15/02/2011.
- [15] "When My Avatar Went to Work". Disponível em <http://spectrum.ieee.org/robotics/industrial-robots/when-my-avatar-went-to-work>. Acedido em 15/02/2011.
- [16] "Anybot's QA: The Best Humanoid Robot Yet ". Disponível em <http://www.popularmechanics.com/technology/gadgets/4298623>. Acedido em 15/02/2011.
- [17] "Elektor Wheelie - self-balancing vehicle." Disponível em <https://www.elektor.com/projects/elektorwheelie.986808.lynkx>. Acedido em 08/06/2011
- [18] "Calha DryLin ZLW-0630-Basic-02". Disponível em http://www.igus.pt/_Product_Files/Download/pdf/66_gl2_e_dryl_sht_rz.pdf. Acedido em 28/03/2011.
- [19] "Nanotec Steep Motor ST4118D1804." Disponível em http://en.nanotec.com/steppermotor_st4118.html?highlight=857. Acedido em 08/06/2011.

- [20] "Nanotec SMC42-2,0-1 MICROSTEP DRIVER." Disponível em <http://pt.farnell.com/nanotec/smc42-2-0-1/driver-microstep-compact/dp/4743131>. Acedido em 08/06/2011.
- [21] "Microship PIC32 Ethernet Starter Kit." Disponível em http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2615&dDocName=en545713. Acedido em 08/06/2011.
- [22] "D-Link DAP-1522." Disponível em <http://www.dlink.pt>. Acedido em 08/06/2011.
- [23] "Câmara IP Zavio F210A." Disponível em http://www.chipsite.pt/detalhe_produto.php?pr=5144. Acedido em 08/06/2011.
- [24] Paulo Gomes da Costa, "Plataforma SimTwo." Disponível em <http://paginas.fe.up.pt/~paco/wiki/index.php?n=Main.SimTwo>. Acedido em 27/05/2011.
- [25] José Xavier, "matlab2matriz," 2010. Informação sobre autor disponível em http://paginas.fe.up.pt/~ee06091/?page_id=30. Acedido em 22/06/2011.
- [26] "Microchip Stack". Disponível em <http://www.microchip.com/tcpip>. Acedido em 27/06/2011.