

página em branco



Faculdade de Ciências e Tecnologia da Universidade de Coimbra
Departamento de Engenharia Electrotécnica e de Computadores

Curso: Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Título: Desenvolvimento de uma Plataforma de Todo-o-Terreno

Nome: Tiago Barros da Silva

Júri:

Presidente: Professor Doutor Rui Paulo Pinto da Rocha

Orientador: Professor Doutor Lino José Forte Marques

Vogal: Professor Doutor Paulo Jorge Carvalho Menezes

Data: Coimbra, Setembro 2015

Agradecimentos

Gostaria de agradecer, em primeiro lugar, ao meu orientador Professor Doutor Lino José Forte Marques, por me ter dado a oportunidade de desenvolver um projecto tão abrangente e de me ter acompanhado durante o mesmo, onde fui constantemente desafiado em termos de conhecimentos e competências.

Ao Instituto de Sistemas e Robótica da Universidade de Coimbra pela total disponibilidade de recursos e condições de trabalho no decorrer da dissertação.

Aos meus colegas do Instituto Sistemas e Robótica Ricardo Faria, David Santos, Bruno Gouveia, Gonçalo Cabrita, Luís Bica, Tiago Neves pelas dicas e conselhos quando as soluções nem sempre eram óbvias e o desespero tornava o caminho mais labiríntico. Pela boa amizade e companheirismo que sempre demonstraram tanto nos bons como nos maus momentos.

Ao Marco Mendes, Nuno Mendes e Tiago Claro pela amizade que sempre tiveram, tanto nos bons como nos maus momentos. Espero que a nossa amizade perdure eternamente.

A todos os professores e colegas que me inspiraram durante o percurso universitário de forma positiva.

E por fim, à minha família, em especial à minha mãe e aos meus irmãos, por terem aguentado as minhas impaciências e más disposições durante este período tão complicado.

O meu muito Obrigado.

Resumo

O objectivo principal deste projecto foi controlar um conjunto de movimentos de uma plataforma móvel de todo-o-terreno (ISR-TT), através de teleoperação. Para tal, foi necessário automatizar a plataforma, coadunando *hardware*, *firmware* e *software* e criando uma arquitectura com três camadas.

Foram concebidas duas caixas de controlo (eixo frontal e eixo traseiro), onde estão os microcontroladores, andares de potência e PCBs de interface. Foi concebido, também, uma caixa que concentra todas as ligações de potência e de comunicação. O *firmware* é responsável pelo controlo directo dos actuadores, para isso, executa ciclos de controlo de posição para os actuadores lineares (suspensão e direcção) e controlo de velocidade para as rodas. As referências são recebidas da camada a montante, usando um protocolo de comunicação para interagir com o *software*. Na camada hierarquicamente mais elevada é executado o *software*. O *software* foi desenvolvido, usando um *framework* designado por *Robot Operating System-ROS*. É nesta camada que estão implementadas as geometrias de condução e são recebidos os comandos do *Wii-Mote*, usado para controlar a plataforma remotamente.

Os resultados obtidos em termos de desempenho de controlo de posição e velocidade são bons, tendo, cada um, um erro máximo inferior a 1% . Em termos de condução, a plataforma consegue executar curvaturas com um raio mínimo de 3 metros.

Palavras Chaves: Plataforma móvel de todo-o-terreno, Teleoperação, *Firmware*, *Hardware*, *Software*.

Abstract

The main goal of this project was automation and teleoperation of an off-road platform with four steerable wheels and an active suspension. For this purpose, custom hardware, software and firmware was developed. The interface between the several microcontrollers, that integrate the platform, was achieved using several custom developed PCBs.

The firmware was developed to control six actuators and a custom made communication protocol was implemented to serve as interface between the lower and upper layer

The software was developed upon ROS, a framework as the capability to create a abstraction layer for further developments. In this layer, Ackerman and double Ackerman steering was implented.

The close-loop controllers have good performance. Each controller has a error below 1%. The platform can perform curves with radius below 3m.

Keywords: Teleoperation, off-road platform, hardware, firmware and software.

Lista de Acrónimos

ADC	-	Analogue-to-Digital Converter
AMR	-	Autonomous Mobile Robot
ASCII	-	American Standard Code for Information Interchange
BLDC	-	Brushless DC
DAC	-	Digital-to-Analogue Converter
DC	-	Direct Current
DoF	-	Degrees of Freedom
I ² C	-	Inter-Integrated Circuit
ICR	-	Instantaneous Center of Rotation
JPL	-	Jet Propulsion Laboratory
MER	-	Mars Exploration Rover
NASA	-	National Aeronautics and Space Administration
OSI	-	Open Systems Interconnection
PCB	-	Printed Circuit Board
PI	-	Proporcional Integral
PID	-	Proporcional Integral Derivativo
PWM	-	Pulse Width Modulation
ROS	-	Robot Operating System
RTOS	-	Real Time Operating Sistem
USAR	-	Unmanned Search and Rescue
SCI	-	Serial Communications Interface
SMD	-	Surface Mount Device
TTL	-	Transistor-Transistor Logic
UGV	-	Unmanned Ground Vehicle
USB	-	Universal Serial Bus

Conteúdo

Lista de Figuras	x
Lista de Tabelas	xiii
1 Introdução	1
1.1 Motivação e Objectivos	3
1.2 Contribuições	4
1.3 Estrutura da Dissertação	4
2 Estado da Arte	6
2.1 Plataformas para Terrenos Acidentados	6
2.1.1 Métricas de Mobilidade	8
2.1.2 Geometrias de Condução	9
2.2 Rodas	12
2.2.1 Roda Fixa	13
2.2.2 Roda Orientável	14
2.3 Actuadores	14
2.3.1 Motores DC Com Escovas	15
2.3.2 Motores DC Sem Escovas	18
2.4 Controladores	20
2.4.1 Controladores Digitais	22
2.4.2 Sintonização do Controlador	23

3	Plataforma ISR-TT	25
3.1	Arquitectura	26
3.2	Hardware	27
3.2.1	Locomoção	27
3.2.2	Direcção	28
3.2.3	Suspensão	29
3.2.4	Caixa de Controlo	30
3.2.5	Caixa de Potência e Comunicação	39
3.2.6	Comando Remoto	40
3.3	Firmware	42
3.3.1	Núcleo de Tempo Real	42
3.3.2	Tarefas para Controlo de Rodas	43
3.3.3	Tarefas para Controlo dos Motores Lineares	45
3.3.4	Tarefas de Comunicação	46
3.3.5	Protocolo de comunicação	46
3.4	Software	50
3.4.1	ROS	50
3.4.2	Hierarquia	51
3.4.3	Nó ISR-TT	51
3.4.4	Nó de Teleoperação	55
4	Testes e Resultados	56
4.1	Métricas de Avaliação	56
4.1.1	Locomoção	56
4.1.2	Direcção	57
4.1.3	Suspensão	59
4.2	Testes de condução	60
4.2.1	Slalom	60
4.2.2	Circunferência	61

5	Conclusões	62
5.1	Trabalho Futuro	63
	Apêndice A Desenvolvimento da Caixa de Controlo	67
A.1	Caixa	67
A.2	Conectores	67
A.3	PCB de Interface	69
A.3.1	Lista de Material	69
A.3.2	Circuito	69
	Apêndice B Firmware	72
B.1	Funcionamento	72
	Apêndice C Software	77
C.1	Matlab	77
C.2	Mapeamento	79

Lista de Figuras

1-1	Exemplos de UGVs objectivos diferentes.	2
1-2	Estado inicial da plataforma	3
1-3	Ilustração da plataforma em modo teleoperação	4
2-1	Comparação de mecanismos de suspensões	8
2-2	Condução de <i>Ackerman</i> [8]	11
2-3	Representação dos centros de curvatura para os dois eixos [2].	11
2-4	a) Roda normal; b) Roda <i>Castor</i> ; c) Roda <i>Swedish</i> ; d) Roda tipo bola ou esfera [21].	13
2-5	Representação dos parâmetros que modelam a cinemática da roda normal [21].	13
2-6	Circuito com os componentes básicos de um motor DC [19]	15
2-7	Ilustração do motor DC com escovas	15
2-8	Circuito eléctrico equivalente do motor DC com escovas [13]	16
2-10	Funcionamento do motor DC sem escovas	18
2-11	Saída dos sensores e sinais de comutação	19
2-12	Circuito de comando simplificado das 3 fases de um motor BLDC	20
2-13	Esquema de controlo de malha fechada digital	22
3-1	Plataforma ISR-TT	25
3-2	Arquitectura da plataforma	26
3-3	Actuadores linear da plataforma.	29
3-4	Actuadores linear da plataforma.	30
3-5	Caixa de controlo.	30

3-6	Actuadores a controlar	31
3-7	Microcontrolador STM32-E407.	32
3-8	PCB na fase desenvolvimento (a) e na fase final (b)	33
3-9	Cricuito electrico dos filtros passa-baixo.	34
3-10	Sinais dos sensores de efeito de Hall.	35
3-11	Filtro passa-baixo para converter PWM num valor analógico.	36
3-12	Sinal para referência de velocidade do BLDC.	37
3-13	Circuito eléctrico da conversão da comunicação de TTL para RS-485 e vise-versa.	38
3-14	Circuito de potência.	39
3-15	Comando Wii Mote.	40
3-16	Disposição dos actuadores na plataforma para o <i>firmware</i>	43
3-17	Diagrama de Blocos do funcionamento do <i>firmware</i> para as rodas. . .	43
3-18	Diagrama de Blocos do funcionamento do <i>firmware</i> para os actuadores lineares.	45
3-19	Diagrama de blocos da interação entre o <i>firmware</i> e <i>software</i>	46
3-20	Diagrama de blocos do funcionamento do protocolo de comunicação entre camadas.	47
3-21	Estrutura da mensagem do protocolo de comunicação.	47
3-22	Diagrama de blocos do funcionamento do software.	51
3-23	Representação dos parâmetros da condução de <i>Ackerman</i> [2].	52
3-24	Representação dos centros de curvatura para os dois eixos [2].	54
4-1	Gráfico do controlo de velocidade da roda.	57
4-2	Controlo de posição do actuador da direcção.	58
4-3	Controlo de posição do actuador da direcção com ampliação.	58
4-4	Controlo de posição do actuador da direcção.	59
4-5	Controlo de posição do actuador da direcção com ampliação.	60
4-6	Teste de slalom.	60
4-7	Teste da circunferência	61

A-1	Caixa de controlo.	68
A-2	Caixa de controlo com a perspectiva frontal.	68
A-3	Caixa de controlo perspectiva lateral.	69
A-4	Conectores do PCB.	70
C-1	Menu principal do software desenvolvido para Matlab.	78
C-2	Menu Manual do software Matlab.	78
C-3	Diagrama de blocos do funcionamento do software desenvolvido em Matlab.	79
C-4	Gráfico dos pontos do mapeamento do actuador M3 traseiro.	80
C-5	Gráfico dos pontos do mapeamento do actuador M6 traseiro.	80
C-6	Gráfico dos pontos do mapeamento do actuador M3 frontal.	81
C-7	Gráfico dos pontos do mapeamento do actuador M6 frontal.	81

Lista de Tabelas

2.1	Apresentação de três plataformas diferentes	7
2.2	Configurações das rodas para plataformas de quatro rodas [21]	9
2.3	Tabela dos parâmetros de Ziegler e Nichols [3]	24
3.1	Roda e as suas características eléctricas e mecânicas	27
3.2	Controlador da roda e as suas características	28
3.3	Tabela com as características do actuador linear da direcção	28
3.4	Tabela com as características do actuador linear da suspensão.	29
3.5	Tabela com entradas e saídas necessárias	32
3.6	Funcionalidades implementadas no Wii-Mote	41
3.7	Descrição dos campos da mensagem do protocolo de comunicação.	48
3.8	Funcionalidades implementadas	49
3.9	ID dos dispositivos	49
3.10	ID das tarefas	49
A.1	Lista de material da PCB.	71

Capítulo 1

Introdução

Actualmente, o desenvolvimento de veículos terrestres não tripulados (*Unmanned Ground Vehicle* - UGV) tem tido um aumento vertiginoso em aplicações de todo-o-terreno como por exemplo em missões de busca e salvamento, exploração planetária, militares ou florestais [11].

No domínio militar os UGVs são utilizados há vários anos em teatros de operação como no Iraque e Afeganistão, onde têm a missão de apoiar operações militares. As missões mais frequentes que estes dispositivos efectuam vão desde desarmamento de bombas ao patrulhamento. Em 2007 as Forças Armadas dos Estados Unidos contavam com aproximadamente 5000 robôs no Iraque e no Afeganistão, um aumento substancial comparando com os 150 em 2004. Esta instituição governamental norte-americana reporta que em 2005 foram desarmados mais de 1000 engenhos explosivos por UGV [6].

No domínio de busca e salvamento existem várias instituições e concursos que promovem o desenvolvimento de robôs não tripulados para busca e salvamento (*Unmanned Search and Rescue* - USAR Robots) com características específicas para as aplicações em cenários de desastre nomeadamente em terremotos, catástrofes naturais e catástrofes como em Chernobyl e Fukushima, onde a radiação radioactiva é nociva para o ser humano. Nestes cenários os UGVs têm a missão de supervisionar reactores e acidentes associados, assim como, realizar medições da radiação [10]. A nível europeu, NIFTi [14] e ICARUS [7] são alguns projectos com casos de sucesso.



(a) NIFTi UGV [14]



(b) Rover Curiosity da NASA [9]

Figura 1-1: Exemplos de UGVs objetivos diferentes.

Os UGVs, também são utilizados, de uma forma dominante, na exploração planetária. Neste domínio a NASA e outras instituições governamentais e privadas focam os seus recursos para a optimização de *rovers* (veículo espacial projectado para navegar sobre superfícies planetárias).

Em 1997 o *rover Sojourner* foi colocado na superfície de Marte. Em 2004 chegaram a Marte mais dois *rovers*, *Spirit* e *Opportunity*. Estes *rovers* tinham a missão de analisar a geologia da superfície, histórico climático e o potencial para a existência de vida no passado e no presente. O último *rover* a chegar à superfície de Marte foi *Curiosity* [9].

A missão principal de um UGV é de ser um intermediário, que, em primeira instância, recolhe dados para os operadores humanos ou providencia a capacidade de aproximar a instrumentação científica a alvos pretendidos. A capacidade de se adaptar ao terreno irregular é crucial para concretizar estas tarefas, para isso, é necessário maximizar a mobilidade. Um dos factores mais influenciadores na mobilidade é o mecanismo de locomoção. Existem diferentes sistemas de locomoção como: rodas, pernas, lagartas ou combinações das opções anteriores (chamados sistemas híbridos). A adaptação ao terreno pode ser passiva ou activa [20].

Porém, são as rodas o sistema de locomoção mais usado em plataformas moveis, devido à menor complexidade mecânica e de controlo.

Para desenvolver uma plataforma móvel, em termos de controlo de movimentos, é necessário coadunar diversos módulos tais como: *hardware*, *firmware* e *software*.

1.1 Motivação e Objectivos

Através dos exemplos anteriormente enunciados, constata-se que os robôs de todo-terreno têm vindo a ganhar notoriedade em várias áreas, sem os quais o ser humano, por exemplo, não poderia analisar a constituição de outros planetas ou até desactivar, de forma segura, engenhos explosivos. É devido a estes benefícios que o desenvolvimento de novas plataformas, cada vez mais adaptadas às tarefas propostas, é uma mais valia.

Pretende-se, com este projecto automatizar uma plataforma móvel com quatro rodas motrizes, direccionáveis e uma suspensão activa, por forma, a poder-se controlar os movimentos através de teleoperação. A plataforma, inicialmente, já possuía uma estrutura pré-definida, assim como, os actuadores que nela actuam. Estes são: actuadores lineares para a direcção e a suspensão e rodas com os motores DC embutidos no eixo de rotação das rodas.



Figura 1-2: Estado inicial da plataforma

Pretende-se, também, implementar uma camada de abstracção para facilitar a integração de módulos mais avançados para sistemas robóticos. Para este efeito, é necessário desenvolver uma arquitectura distribuída em três níveis (*hardware*, *firmware* e *software*), com o objectivo de criar um sistema modular.

Por fim, pretende-se teleoperar a plataforma através de um comando remoto, implementando diferentes geometrias de condução (*Ackerman* e *Ackerman* quatro rodas), provando a versatilidade que a plataforma consegue ter com as suas propriedades estruturais.

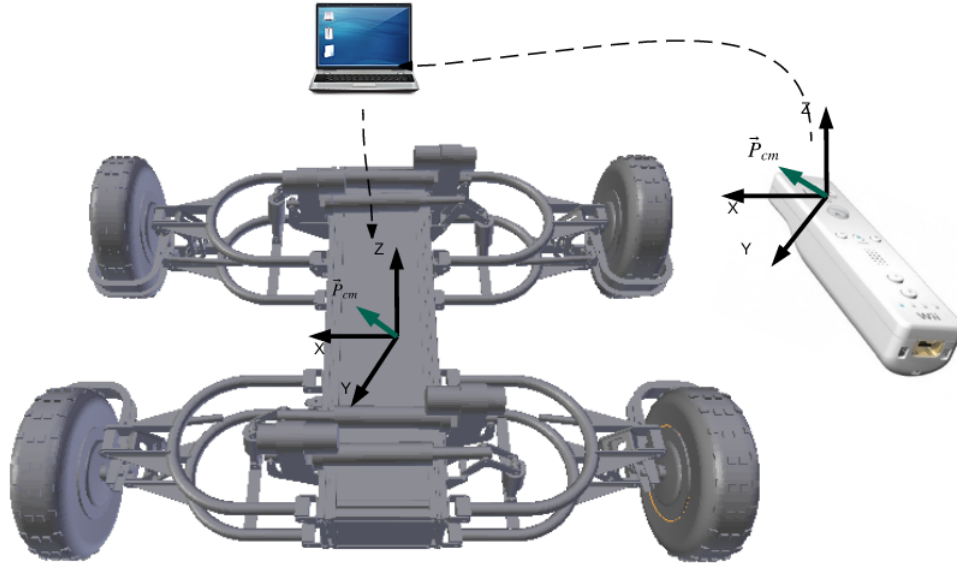


Figura 1-3: Ilustração da plataforma em modo teleoperação

1.2 Contribuições

Este projecto não tem como principal missão contribuir com resultados, técnicas ou métodos inovadores para a área da robótica móvel, pois todos os conceitos, técnicas ou métodos implementados são consolidados e amplamente usados. Pretende-se, sim, criar mais uma plataforma com características estruturais e electromecânicas bem definidas, por forma, a conseguir adaptar-se e movimentar-se, eficazmente, em terrenos pouco estruturados.

A principal contribuição deste projecto reside na construção de uma plataforma robusta, de forma, a comunidade científica poder usufruir das suas características para testar novos conceitos e algoritmos, estes, sim, inovadores, abstraindo-se das características electromecânicas de baixo-nível.

1.3 Estrutura da Dissertação

A presente dissertação está dividido em cinco capítulos (Introdução, Estado da Arte, Plataforma, Resultados e Conclusão). Os conteúdos desenvolvidos no âmbito deste projecto são das mais diversas áreas devido às características multidisciplinares

do próprio, tendo sido necessário sub-dividir o capítulo destinado ao desenvolvimento por áreas (*Hardware*, *Firmware* e *Software*).

- **Capítulo 1** - Neste capítulo é feita uma breve contextualização do tema da dissertação, apresentado algumas plataformas de todo-o-terreno. Ainda neste capítulo, também são apresentados os objectivos, contribuições deste projecto, assim como a divisão estrutural do documento.
- **Capítulo 2** - Este capítulo destina-se a apresentar os conceitos teóricos usados durante o desenvolvimento deste projecto.
- **Capítulo 3** - Este capítulo é referente aos desenvolvimento prático do projecto, estando sub-dividido em três secções (*Hardware*, *Firmware* e *Software*). Aqui são apresentados todos os componentes, métodos e conceitos usados e desenvolvidos para concretizar o projecto. .
- **Capítulo 4** - É neste capítulo que estão patentes os resultados relativos ao controlo de baixo e alto nível. São estes resultados que caracterizam a plataforma em termos de erro e precisão de afectar um determinado movimento ou manobra.
- **Capítulo 5** - O presente capítulo destina-se a expor as conclusões retiradas ao longo deste projecto e recomendações para melhorar a plataforma.

Capítulo 2

Estado da Arte

Neste capítulo são apresentadas algumas plataformas com características semelhantes à plataforma usada neste projecto (*ISR-TT*), de forma, a poder-se realçar vantagens e desvantagens.


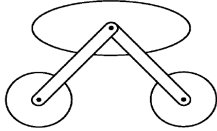


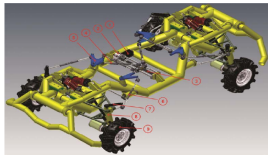
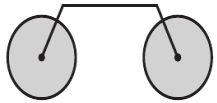
Existem enumeras métricas para qualificar numericamente as capacidades de um sistema, no entanto, nem todas são relevantes para uma aplicação específica. Sendo necessário, para este efeito, determinar as métricas que melhor avaliam o desempenho em função dos requisitos do projecto. São abordados indicadores de desempenho como: *trafficability*, *maneuverability* e *terrainability*. Ainda neste capítulo, são fundamentados os conceitos teóricos, mais relevantes, para desenvolver as três componentes deste projecto (*hardware*, *firmware* e *software*). Em termos de hardware os módulos usados tiveram que ser seleccionados em conformidade com o funcionamento dos actuadores existentes na plataforma.

2.1 Plataformas para Terrenos Acidentados

A definição de uma métrica para plataformas móveis, nem sempre é tarefa fácil, pois cada plataforma é concebida com requisitos e para fins diferentes. No entanto, é apresentado um conjunto de plataformas de todo-o-terreno com quatro rodas, com sistemas passivos e activos de adaptação, podendo, assim, comparar as componentes mais relevantes numa plataforma móvel, como a direcção, suspensão e tracção. Em

cada caso são enunciadas as características de locomoção e suspensão, assim como, é fornecido uma representação 2D da plataforma.

Tabela 2.1: Apresentação de três plataformas diferentes

<p>Nome: <i>NANO ROVER</i> Autor: JPL-USA Referência: [23] Condução: Diferencial Suspensão: Activa</p>		
<p>Nome: <i>NOMAD</i> Autor: Carnegie Mellon-USA Referência: [1] Condução: Síncrona Suspensão: Passiva</p>		
<p>Nome: <i>MNE-ROBEKO</i> Autor: University of Montenegro Referência: [22] Condução: <i>Ackerman</i> 4 rodas Suspensão: Passiva</p>		

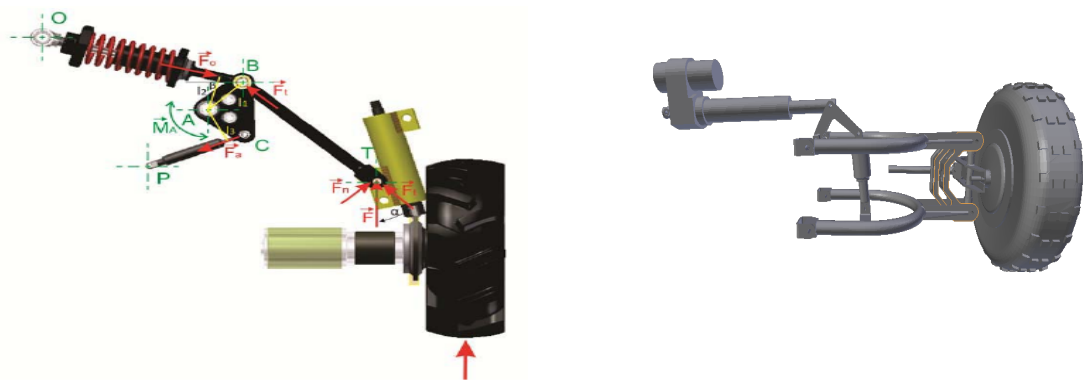
NANO ROVER é uma plataforma com quatro rodas motrizes e com dimensões reduzidas, sendo o comprimento 20 cm. Uma das características desta plataforma é a capacidade de conseguir, só através dos motores de tracção, posicionar a sua base verticalmente. Os motores DC são comandados através de sinais modulados por largura de impulso de 8 bits com *encoders* para efeitos de *feedback* e sensores de efeito de Hall para obter odometria [23].

NOMAD é uma plataforma dividida em duas partes simétricas (esquerda e direita), usando um eixo pivotante para fazer a ligação entre as duas partes. Este pivô, faz com que a plataforma se adapte ao terreno de forma passiva. Em termos de locomoção a plataforma possui quatro motores DC embutidos no eixo de rotação da roda. O controlo da direcção é conseguido através da orientação de duas rodas do mesmo lado, usando um actuador para cada lado. A estrutura tem a capacidade de encolher ou expandir através de um mecanismo de barras [1, p. 51].

MNE-ROBEKO é uma plataforma, estruturalmente, parecida com um carro. Tem quatro rodas direccionáveis e motrizes. A direcção é controlada através de um mecanismo centralizado que implementa condução de *Ackerman* de quatro rodas,

tendo um motor eléctrico e um *encoder* que controla a direcção das quatro rodas. A suspensão é composta por um conjunto de mola-amortecedor para absorver irregularidades em terrenos não estruturados [22].

As plataformas apresentadas na tabela 2.1 foram seleccionadas, tendo em conta as características únicas como o sistema de suspensão activo do *NANO ROVER*, que consegue alterar a configuração da base (altura e inclinação), usando apenas os motores de locomoção. Na plataforma *NOMAD*, é de realçar a capacidade de expansão (35% da posição encolhida) da sua estrutura, conseguindo aumentar 20% a sua estabilidade estática. A plataforma *MNE-ROBECO* é, estruturalmente, muito semelhante a *ISR-TT*. O mecanismo da suspensão, também, é bastante interessante, podendo ser objecto de análise para melhorar a suspensão da *ISR-TT*.



(a) Mecanismo da suspensão da plataforma *MNE-ROBECO* (b) Mecanismo da suspensão da plataforma *ISR-TT*

Figura 2-1: Comparação de mecanismos de suspensões

2.1.1 Métricas de Mobilidade

A mobilidade é o alicerce de qualquer plataforma móvel, sendo um dos requisitos mais importantes na projecção. A selecção e a configuração dos actuadores que actuam sobre a plataforma condicionam a sua mobilidade. O tipo suspensões (activas ou passivas) e a locomoção, são requisitos que têm que ser adequados às tarefas que vão executar e ao terreno onde vão operar.

Diferentes métricas ajudam a avaliar diferentes aspectos associados à locomoção.

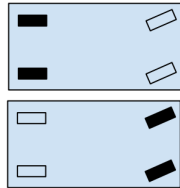


A *maneuverability*, *trafficability* e *terrainability* (aprofundados em [1, p. 20-42]) são as métricas mais usadas para avaliar plataformas com rodas.

- **Maneuverability** - “Ability to change its heading, avoid obstacles and navigate through cluttered environments” [1];
- **Trafficability** - “Ability to generate traction and overcome resistances” [1];
- **Terrainability** - “The locomotion’s ability to negotiate rough terrain features without compromising the vehicle’s stability and forward progress” [1].

2.1.2 Geometrias de Condução

Para plataformas com quatro rodas existem várias configurações de rodas possíveis, estas são apresentadas de forma resumida na tabela 2.2.

Tabela 2.2: Configurações das rodas para plataformas de quatro rodas [21]

Configuração	Descrição	Exemplo
	<p>Duas rodas motrizes na parte de trás ou na parte da frente, duas rodas direcionáveis à frente ou a trás respectivamente. As rodas têm que ter orientações diferentes para não haver deslizamento.</p>	<p>Condução de <i>Ackerman</i> e.g veículos</p>
	<p>Quatro rodas motrizes e direcionáveis</p>	<p>Condução Síncrona ou Omnidirecional e.g Hyperion (CMU)</p>
	<p>Quatro rodas motrizes</p>	<p>Condução Diferencial e.g Husky ISR</p>

A **condução Diferencial** consiste num conjunto de duas rodas motrizes independentes. A plataforma com esta configuração possui dois graus de liberdade e o centro de massa encontra-se no centro do eixo que liga as duas rodas.

A **condução Omidireccional** consiste num conjunto, com no mínimo duas rodas direccionáveis e com uma ou mais rodas livres, servindo só de suporte. Esta plataforma pode-se mover num plano com três graus de liberdade.

A **condução Síncrona** tem a vantagem de todas as rodas terem a mesma direcção. O número mínimo de motores necessários são dois, um acciona uma correia ou uma corrente que actua as rodas e o segundo para definir a direcção. A plataforma com esta configuração consegue atingir qualquer ponto num determinado plano, no entanto está limitado a dois graus de liberdade, tendo em conta, que não consegue rodar.

A **condução Ackerman** é amplamente usado no sector automóvel e é constituído por um eixo com rodas fixas e um eixo com rodas direccionáveis. Os robôs com esta configuração podem possuir várias rodas, no entanto, só pode ter um centro instantâneo de rotação (*Instantaneous Center of Rotation* - ICR). Para haver um movimento sem deslizamento ICR é a única solução. Este ponto é, onde os vectores do espaço nulo se intersectam. Assim sendo, a mobilidade depende das restrições que afectam a plataforma e não o numero de rodas nela existente. Uma plataforma com condução *Ackerman* possui três graus de liberdade, contudo não são independentes entre si. Esta configuração tem a desvantagem de possuir uma manobrabilidade reduzida, em que o raio do centro de curvatura é sempre maior que o comprimento da plataforma, no entanto tem a vantagem de atribuir estabilidade lateral para velocidades mais elevadas [21].

A **condução de Ackerman de quatro rodas**, possui 4 rodas direccionáveis. Esta configuração permite aumentar a manobrabilidade em relação à configuração *Ackerman* de duas rodas. A configuração *Ackerman* com quatro rodas direccionáveis permite ter um raio de curvatura mais reduzido, tendo mais manobrabilidade em espaços reduzidos [16]. Enquanto, que na condução de *Ackerman* com duas rodas direccionáveis, o conjunto de soluções para o ICR é a recta que intersecta as rodas traseiras, na configuração de *Ackerman* com quatro rodas direccionáveis as soluções possíveis do ICR é um plano, podendo ter várias soluções para o mesmo raio de curvatura.

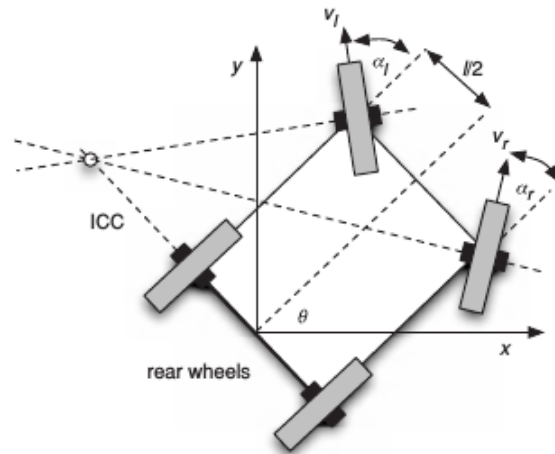


Figura 2-2: Condução de *Ackerman* [8]

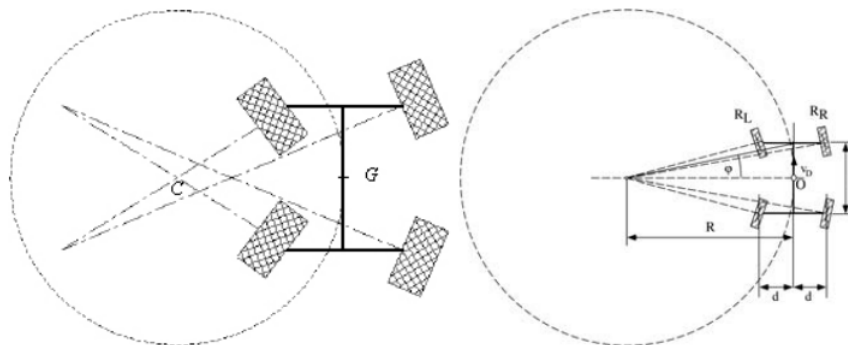


Figura 2-3: Representação dos centros de curvatura para os dois eixos [2].

2.2 Rodas

A escolha do tipo de roda e a sua configuração são requisitos importantes na concepção de plataformas, pois a mobilidade e a estabilidade dependem das restrições de mobilidade da mesma, por sua vez, dependem das restrições cinemáticas de cada roda.

Existem quatro classes principais de tipos de rodas, tendo cada uma a sua cinemática. O ambiente e as tarefas que a plataforma vai executar são factores preponderantes para a determinação do tipo de roda a usar.

Os quatro grupos principais de tipo de rodas são:

- **Roda Normal** - Até dois graus de liberdade, rotação em torno do eixo motriz e do ponto de contacto;
- **Roda *Castor*** - Dois graus de liberdade, rotação em torno de um offset em relação à junta de orientação;
- **Roda *Swedish*** - Três graus de liberdade, rotação em torno do eixo motriz, em torno dos rolos e do ponto de contacto;
- **Roda tipo esfera** - Omnidireccional;

Na figura 2-4 estão ilustradas as quatro classes anteriormente enunciadas.

Antes de deduzir as equações que regem as restrições cinemáticas da roda normal fixa e orientável, é necessário assumir as premissas de que o plano da roda permaneça sempre vertical, a existência de apenas um ponto de contacto entre a roda e a superfície e a não existência de deslizamento lateral. Nestas condições a roda efectua um movimento de rolamento puro através do ponto de contacto. Estas aproximações do modelo real facilitam a obtenção das equações, contudo não há perda generalidade.

Em ambientes não estruturados as rodas do tipo normal são as que conferem melhor estabilidade devido às restrições cinemáticas. Com as rodas normal é possível ter duas configurações: roda normal fixa e roda normal orientável. Enquanto, que a roda normal fixa só tem um grau de liberdade (figura 2-5a), a roda normal orientável tem dois graus de liberdade (figura 2-5b) [21].

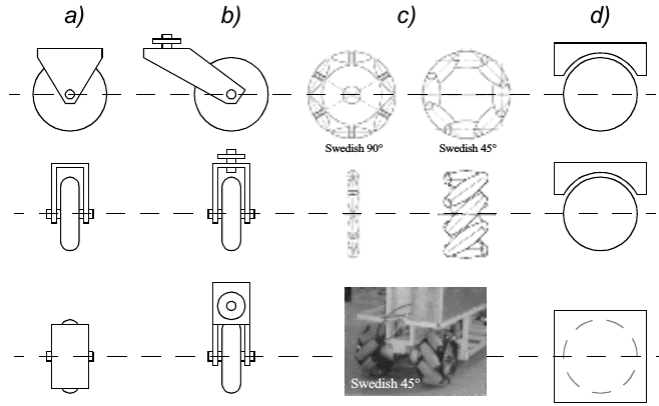


Figura 2-4: a) Roda normal; b) Roda *Castor*; c) Roda *Swedish*; d) Roda tipo bola ou esfera [21].

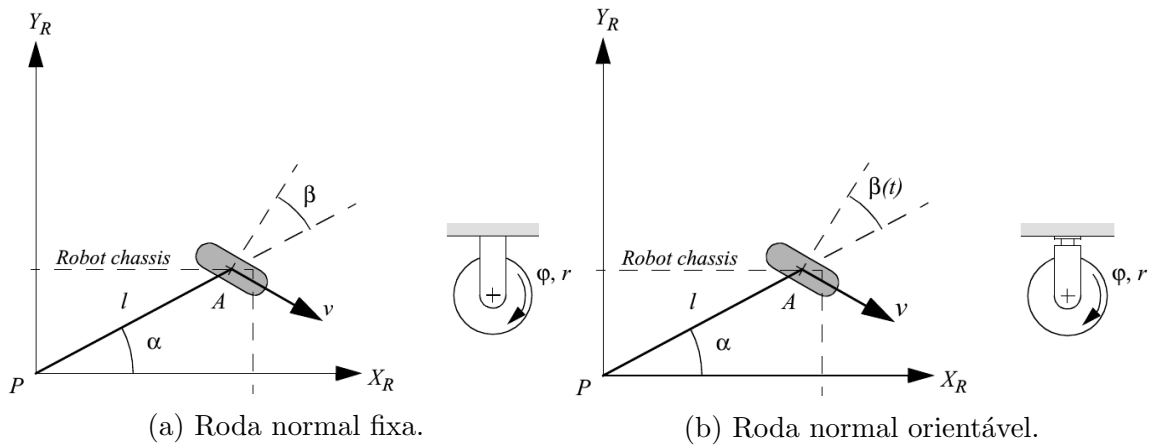


Figura 2-5: Representação dos parâmetros que modelam a cinemática da roda normal [21].

2.2.1 Roda Fixa

Na Figure 2-5a está representada uma roda normal fixa A cuja pose está referenciada no sistema de coordenadas $\{X_R, Y_R\}$ em coordenadas polares com uma distância l e um ângulo α em relação à origem. O ângulo β é, o ângulo entre o plano da roda e a plataforma. A roda possui um raio r , rodando em torno do eixo horizontal, sendo $\varphi(t)$ a posição da roda ao longo do tempo. Os termos $R(\theta)\dot{\xi}_I$ são usados para relacionar os parâmetros de movimento referenciados no sistema de coordenadas $\{X_I, Y_I\}$ no sistema de coordenadas $\{X_R, Y_R\}$.

As restrições de rolamento (*Rolling* eq. 2.1) e de deslizamento (*Sliding* eq. 2.2):

$$[\sin(\alpha + \beta) - \cos(\alpha + \beta)(-l) \cos(\beta)]R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0 \quad (2.1)$$

$$[\cos(\alpha + \beta) \sin(\alpha + \beta)l \sin(\beta)]R(\theta)\dot{\xi}_I = 0 \quad (2.2)$$

2.2.2 Roda Orientável

A diferença entre a roda normal orientável e fixa é a rotação em torno do eixo vertical que passa pelo centro da roda e o ponto de contacto com a superfície, conferindo à roda orientável um grau de liberdade adicional. As equações que definem as restrições da roda normal orientável são semelhantes às equações 2.1 e 2.2, com a excepção do termo β , que para o caso da roda orientável, varia com o tempo $\beta(t)$.

$$[\sin(\alpha + \beta(t)) - \cos(\alpha + \beta(t))(-l) \cos(\beta(t))]R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0 \quad (2.3)$$

$$[\cos(\alpha + \beta(t)) \sin(\alpha + \beta(t))l \sin(\beta(t))]R(\theta)\dot{\xi}_I = 0 \quad (2.4)$$

2.3 Actuadores

Motores eléctricos são frequentemente usados como elemento final em sistemas de controlo de posição ou de velocidade. Os motores eléctricos podem ser classificados em duas categorias principais: motores DC e motores AC. Os motores DC podem ser subdivididos em duas categorias: motores com escovas e sem escovas. Nos motores com escovas, o rotor tem enrolamentos e o estator pode ter um electroímã (enrolamentos) ou ímanes permanentes. Nos motores sem escovas a configuração é invertida, sendo o rotor a ter ímanes permanentes e o estator enrolamentos [3].

2.3.1 Motores DC Com Escovas

Dois circuitos básicos encontram-se em todos os motores DC: o rotor (a parte que roda ou *Armature*) e o estator (a parte estacionária ou *Field Winding*) [19]. Na figura 2-6 está representado um circuito básico de um motor DC.

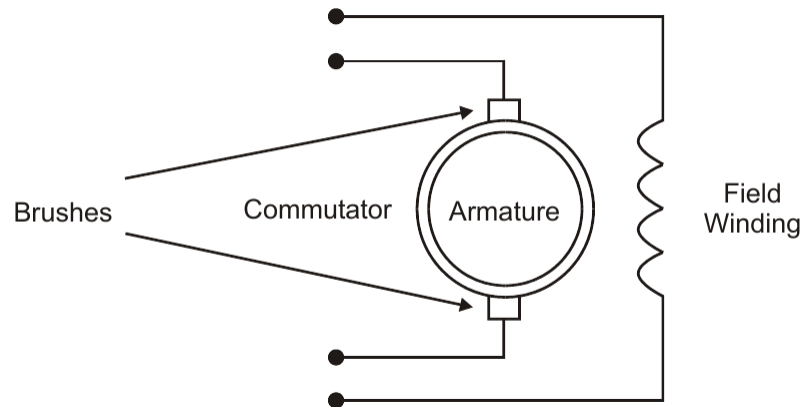


Figura 2-6: Circuito com os componentes básicos de um motor DC [19]

Como pode-se ver na figura 2-6 o rotor e o estator são dois circuitos separados, porém estão fisicamente próximos para haver interação magnética entre ambos [19].

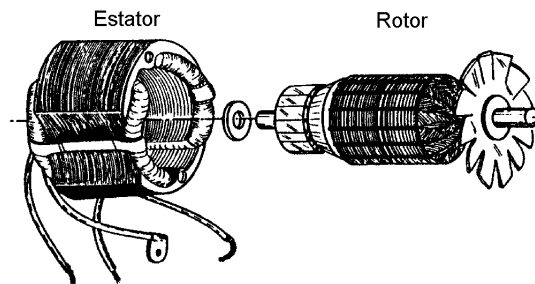


Figura 2-7: Ilustração do motor DC com escovas

O rotor possui um comutador (*commutator*), cuja finalidade é comutar a polaridade do campo magnético para as forças de repulsivas criarem movimento. O movimento é resultado da interação entre as forças repulsivas do rotor e o fluxo magnético do estator. A ligação física da tensão com o rotor é efectuada através das escovas, estas, estão em constante contacto com o comutador e são feitas de carbono. A polaridade dos enrolamentos tem que ser invertida num instante preciso para que haja continuidade no movimento [19].

O processo de construção do estator é semelhante ao do rotor. Este, é constituído por um núcleo laminado de ferro para maximizar o fluxo magnético criado pelos enrolamentos que o envolvem.

A direcção de rotação de um motor DC pode ser invertida, invertendo a corrente do rotor ou a corrente do estator.

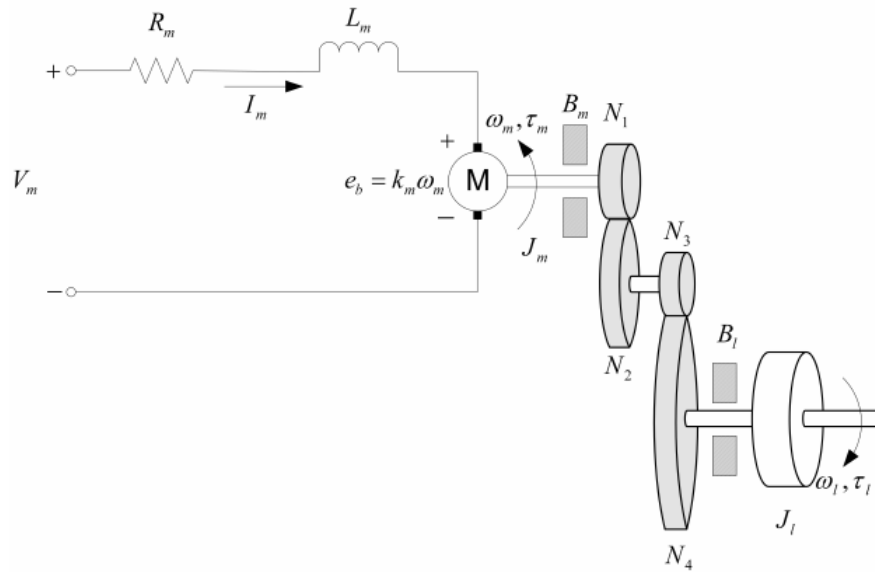


Figura 2-8: Circuito eléctrico equivalente do motor DC com escovas [13]

Enquanto, que o rotor está a rodar, uma força contra-electromotriz é induzida no rotor, devido ao facto, de o rotor estar mergulhado num fluxo magnético. A força contra-electromotriz e_m é proporcional à velocidade angular do rotor ω .

$$e_b = k_m \omega \quad (2.5)$$

Na figura 2-8 está representado o circuito eléctrico equivalente do motor DC com escovas, deste circuito consegue-se obter a relação entre a tensão de entrada V_m e a velocidade angular de saída ω_l

$$V_m(t) - R_m I_m(t) - L_m \frac{dI_m(t)}{dt} - k_m \omega_m(t) = 0 \quad (2.6)$$

Tendo em conta que a indutância do motor L_m é muito menor que a resistência R_m , obtém-se a seguinte igualdade:

$$I_m(t) = \frac{V_m(t) - k_m \omega(t)}{R_m} \quad (2.7)$$

Relacionando o momento de inércia equivalente J_{eq} e a fricção da viscosidade equivalente B_{eq} chega-se à expressão final que relaciona a velocidade angular com a tensão de entrada:

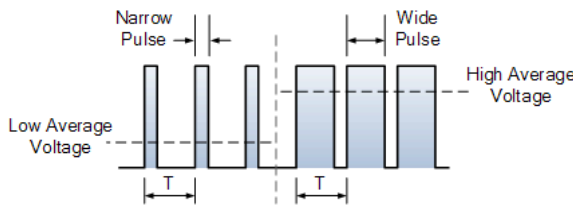
$$\left[\frac{d}{dt} \omega_l(t) \right] J_{eq} + B_{eq,v} \omega_l(t) = A_m V_m(t) \quad (2.8)$$

Da equação 2.8 conclui-se, que a velocidade angular é controlável, controlando a tensão de entrada.

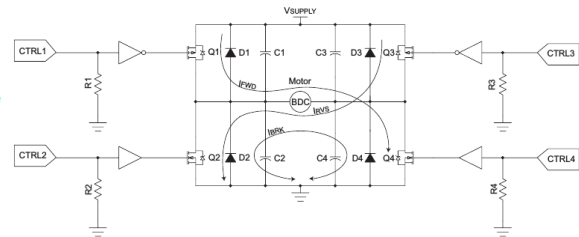
O desenvolvimento completo das equações anteriormente apresentadas encontra-se na referencia [13].

O controlo de velocidade é conseguido através da variação da tensão do rotor como a equação 2.8 comprova. O controlo da tensão, por norma, é obtido usando circuitos electrónicos e usando sinais modulados pela largura de pulso PWM. Estes sinais são aplicados na base dos transístores. Através da variação da largura do pulso (*duty cycle*) consegue-se variar a tensão média aplicada ao rotor, pois o circuito eléctrico do motor é um sistema passa baixo de primeira ordem como se pode verificar na equação 2.9.

$$\frac{I_m}{V - e_b} = \frac{1/L_m}{R_m/L_m + s} \quad (2.9)$$



(a) Sinal modulado em largura de pulso



(b) Circuito de comutação de sentido do motor DC com escovas

2.3.2 Motores DC Sem Escovas

Uma das desvantagens dos motores DC com escovas é a necessidade de ter um comutador e escovas para comutar a corrente. As escovas ao deslizarem nas lâminas do comutador produzem arcos eléctricos. Todos estes fenómenos têm como consequência, o desgaste do material e consequentemente a substituição dos mesmo periodicamente. Para evitar todos estas desvantagens foram criados os motores DC sem escovas BLDC [3]. O motor sem escovas é constituído por um rotor com ímanes permanentes e enrolamentos no estator. A energia eléctrica é convertida em energia mecânica através da força atractiva entre os ímanes permanentes do rotor e o campo magnético giratório induzido nos enrolamentos do estator [4].

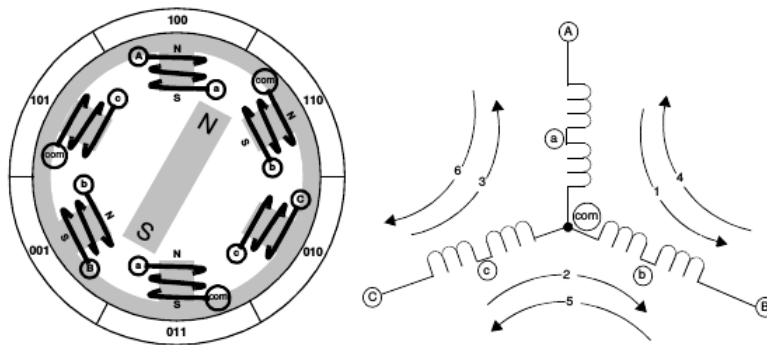


Figura 2-10: Funcionamento do motor DC sem escovas

A maioria dos BLDCs são constituídos por três fases ligadas em estrela. Um motor com esta configuração só necessita de activar duas fases de cada vez para fazer rodar o rotor. O binário é máximo quando o íman permanente do rotor está a 90 graus em relação ao pólo magnético do estator.

A comutação é efectuada através da detecção da posição do rotor, usando sensores de efeito de Hall. Após o sensor devolver a posição do rotor é activada a fase que produzir maior binário. O rotor roda unicamente 60 graus eléctricos por comutação. O circuito de corrente apropriado só é activado quando o rotor estiver a 120 graus do alinhamento do pólo do estator e é desactivado quando estiver a 60 graus do alinhamento, instante em que é activado o próximo circuito e o processo repete-

se. Os motores são equipados com 3 sensores de efeito de Hall, de forma que cada sensor devolva um sinal lógico positivo durante 180 graus eléctricos e um sinal lógico zero durante os restantes 180 graus. Os sensores estão desfasados 60 graus entre eles, desta forma, cada a saída de cada sensor corresponde ao sinal de cada circuito electromagnético. Na figura 2-11 está representado um diagrama temporal com a saída dos sensores e as comutações das fases.

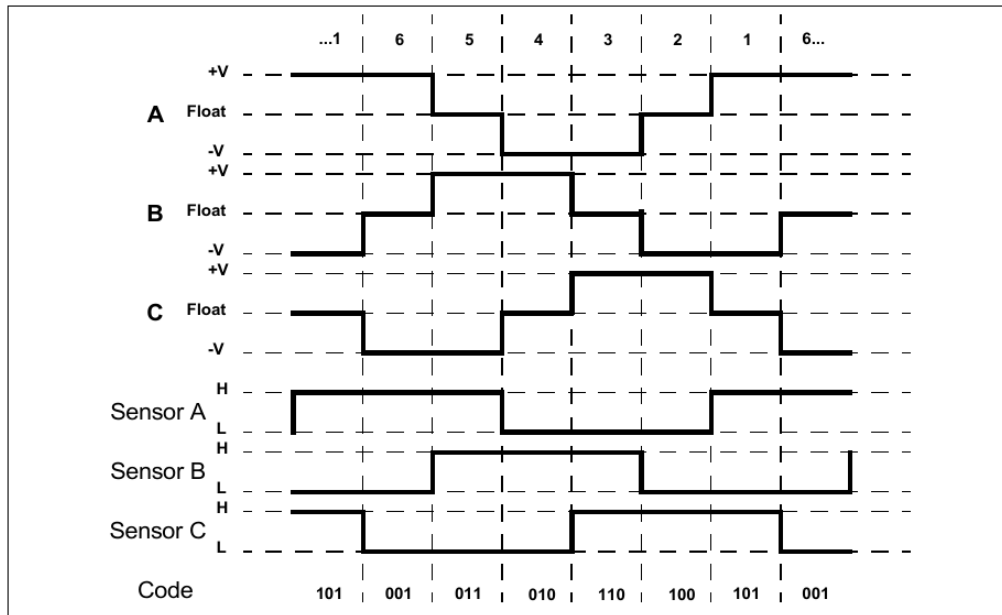


Figura 2-11: Saída dos sensores e sinais de comutação

Os números no topo do gráfico da figura 2-11 correspondem aos números da figura 2-10. Na figura 2-11 é perceptível a formação de um código binário de três bits único por parte das saídas dos sensores, correspondendo cada um deles às 6 fases de comutação. A numeração em torno da periferia do motor na figura 2-10 representa a codificação da posição dos sensores. O pólo norte do rotor aponta para o código em que o rotor se encontra posicionado. Os números representam os níveis lógicos das saídas dos sensores, onde o bit mais significativo é o sensor C e o menos significativo é o sensor A. Cada fase do motor pode ter cada um dos seguintes três estados : low, high ou floating. Um circuito de comutação simplificado está representado na figura 2-12.

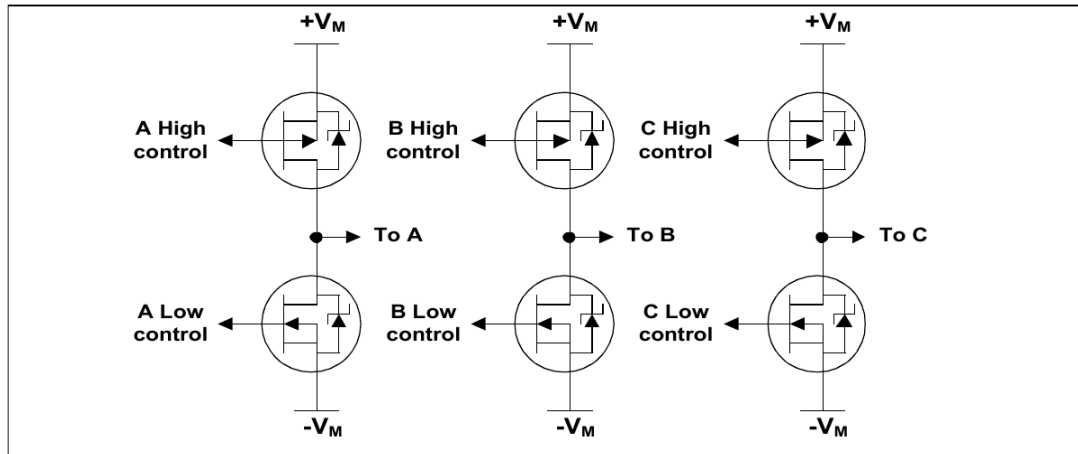


Figura 2-12: Circuito de comando simplificado das 3 fases de um motor BLDC

2.4 Controladores

Em **sistemas de controlo de malha fechada** são usados controladores para comparar, de forma contínua, a saída de um sistema com a uma referência. O erro é a diferença entre a saída e a referência e é usado para gerar uma acção correctiva com o objectivo de reduzir o mesmo.

Qualquer sistema controlado necessita de tempo para reagir a estímulos, provocando **atrasos** (*lag*) no sistema.

O **erro em regime final** (*steady-state error*) é a diferença entre a referência de entrada e a saída após as transições. Este parâmetro, é um valor qualitativo que pode ser usado para estabelecer uma métrica de desempenho entre controladores digitais.

Os controladores digitais podem reagir de forma diferente para o mesmo erro, devido aos diferentes modos de controlo (two-step, P, PI, PID).

O modo **two-step** reage ao sinal de forma binária, tendo só dois estados (ligado ou desligado).

O modo **proporcional** (**P**) produz uma acção de controlo proporcionalmente ao erro, ou seja, o sinal de correcção cresce proporcional com o erro.

O modo **derivativo** (**D**) produz uma acção de controlo proporcional à taxa de variação do erro. Se existir uma alteração repentina no erro, o controlador fornece um sinal correctivo proporcional a essa alteração. O controlo derivativo pode ser

considerado um controlador de antecipação, este, consegue antecipar a existência de um erro grande e corrigi-lo antes de se concretizar. O controlo derivativo nunca é usado sozinho, é sempre usado com controlo proporcional ou integral.

O modo **integral (I)** produz uma acção de controlo, que é proporcional ao integral do erro ao longo do tempo. Um erro constante produz, assim, um sinal de correcção crescente, persistindo, enquanto o erro existir.

Os modos anteriores (P,D e I) podem dar origem a **modos combinados**, proporcional derivativo (PD), proporcional integral (PI), proporcional integral derivativo (PID) [3].

Controlador Proporcional Derivativo :

$$u(t) = K_P e(t) + K_D \frac{d}{dt} e(t) \quad (2.10)$$

onde $u(t)$ é a saída do controlador em função do tempo, K_P é a constante de controlo proporcional, K_D constante de controlo derivativo e $e(t)$ é o erro [3].

Controlador Proporcional Integral :

$$u(t) = K_P e(t) + K_I \int e(t) dt \quad (2.11)$$

onde $u(t)$ é a saída do controlador em função do tempo, K_P é a constante de controlo proporcional, K_I constante de controlo integral.

Controlador Proporcional Integral Derivativo:

$$u(t) = K_P e(t) + K_I \int e(t) dt + K_D \frac{d}{dt} e(t) \quad (2.12)$$

onde $u(t)$ é a saída do controlador em função do tempo, K_P é a constante de controlo proporcional, K_I constante de controlo integral e K_D constante de controlo derivativo.

2.4.1 Controladores Digitais

A figura 2-13 representa um esquema genérico de um controlador de malha fechada digital implementado num microcontrolador. O microcontrolador recebe sinais provenientes de sensores, executa programas de controlo e gera uma acção de controlo correctiva. A leitura de estímulos físicos são feitos através de sensores, que por sua vez, convertem os estímulos em sinais analógicos. Os sinais analógicos são convertidos num formato digital pelo microcontrolador através de conversores analógico-para-digital (*Analógic-to-Digital Converters* - ADC).

O *clock* fornece uma base temporal ao microcontrolador. O instante em que as amostras do sinal analógico são geradas pelo ADC também é estabelecido pelo *clock*. Estas amostras são convertidas para um valor digital e comparadas com a referência, gerando um erro, com o qual, o microcontrolador gera um valor de saída em função do modo de controlo implementado. O sinal de saída é convertido, novamente, para uma formato analógica.

Para os valores amostrados reflectirem um sinal real e contínuo, a **frequência de amostragem** f_s tem que ser no mínimo o dobro da componente de frequência mais alta do sinal contínuo. T_s é o **período de amostragem** [3].

$$T_s = \frac{1}{f_s} \tag{2.13}$$

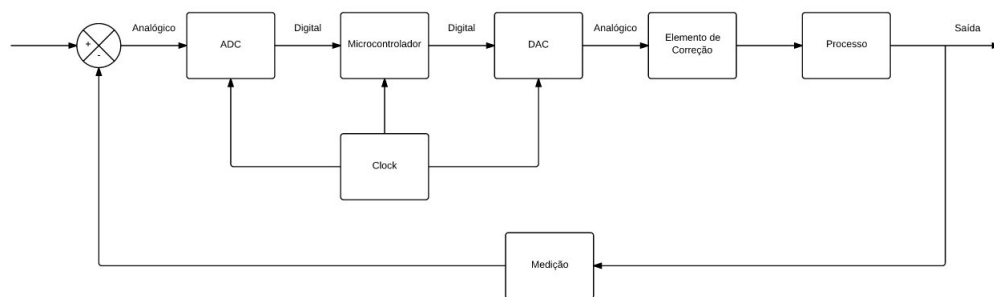


Figura 2-13: Esquema de controlo de malha fechada digital

Implementação

A implementação de um controlador digital, passa por criar um conjunto de instruções que efectuem as operações matemáticas necessárias (em função do controlador pretendido) para obter uma acção correctiva em função do erro. O processamento pode envolver entradas actuais, entradas passadas e saídas passadas [3].

A função de transferência de um controlador PID contínuo:

$$H(s) = K_P + \frac{1}{s}K_I + sK_D \quad (2.14)$$

A função de transferência de um controlador PI contínuo:

$$H(s) = K_P + \frac{1}{s}K_I \quad (2.15)$$

Multiplicando por s é equivalente a derivar. A derivação do erro no instante actual pode ser obtido através de $\frac{e[n]-e[n-1]}{T_s}$, em que $e[n]$ e $e[n-1]$ é a amostra do erro no instante actual e amostra anterior, respectivamente [3].

A divisão por s representa o integral. O integral pode ser calculado, somando o valor do integral de amostras anteriores Int_{prev} com o integral do erro no instante actual $\frac{\frac{1}{2}(e[n]+e[n-1])}{T_s}$ [3].

Equação para um controlador PID:

$$x[n] = K_P e[n] + K_I \left(\frac{(e[n] + e[n-1])T_s}{2} + Int_{prev} \right) + K_D \frac{e[n] - e[n-1]}{T_s} \quad (2.16)$$

Equação para um controlador PI:

$$x[n] = K_P e[n] + K_I \left(\frac{(e[n] + e[n-1])T_s}{2} + Int_{prev} \right) \quad (2.17)$$

2.4.2 Sintonização do Controlador

O termo **sintonização** é usado para definir o processo da selecção dos melhores parâmetros para o controlador. Para um controlador PI e PID significa, escolher as

constantas K_P, K_I e K_D , de forma, a ter o comportamento desejado. Para tal, existem vários métodos, porém, só é abordado o método Ziegler e Nicholas [3].

Com este método os valores das constantes do ganho integral e derivativo são reduzidos para o seu mínimo. A constante K_P é reduzido para valores pequenos e depois incrementada. Durante este processo pequenas perturbações são aplicadas até que a resposta do sistema seja oscilatória. O valor crítico da constante proporcional K_{Pc} é o valor que provoque a oscilação, enquanto que T_c é o período da oscilação [3]. A table 2.3 representa os parâmetros proposto por Ziegler e Nichols para os três modos de controlo.

Tabela 2.3: Tabela dos parâmetros de Ziegler e Nichols [3]

Modo de Controlo	K_P	T_I	T_D
P	$0.5K_{Pc}$		
PI	$0.45K_{Pc}$	$T_c/1.2$	
PID	$0.6K_{Pc}$	$T_c/2.0$	$T_c/8$

Capítulo 3

Plataforma ISR-TT

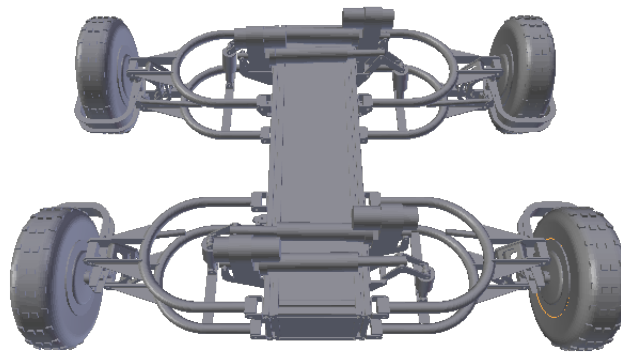


Figura 3-1: Plataforma ISR-TT

A plataforma denominada por ISR-TT, possui quatro rodas motrizes cujo actuator é um BLDC. A direcção e a suspensão são ajustadas por actuadores lineares. A plataforma possui uma distância entre eixos de 0.8 m e uma distância entre rodas de 1.4 m.

3.1 Arquitectura

A arquitectura da plataforma ISR-TT está dividida em três camadas. Começando de baixo para cima, a primeira camada é a camada Física (*hardware*). Nesta camada estão os componentes físicos, como os actuadores lineares, andares de potência (*motor drivers*), placas de circuitos impressos (*Printed Circuit Board - PCB*), microcontroladores, caixas, baterias etc.

A camada intermédia é denominada por Camada de Controlo Directo (*firmware*). A principal missão desta camada é implementar ciclos de controlo de malha fechada de posição e de velocidade, fazendo leituras dos respectivos actuadores.

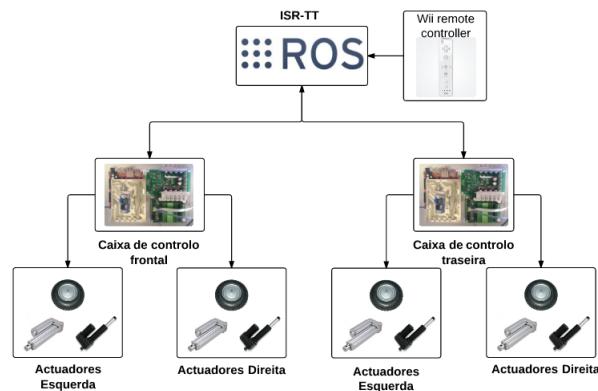


Figura 3-2: Arquitectura da plataforma

Na terceira camada são implementadas as tarefas cujos requisitos temporais não são tão críticos (*soft real-time tasks*), estas, são implementadas num *framework* denominado por ROS. Esta camada (*software*) está subdividida em dois nós (programas independentes). O nó hierarquicamente mais baixo implementa o *device driver*, este nó é denominado por *ISR-TT driver* e tem como função interagir com a camada inferior (*firmware*), de forma, a receber comandos genéricos de velocidade linear e angular e converte-los em unidades dos actuadores, em função da geometria de condução seleccionada.

Ainda na camada de alto nível, existe um segundo nó identificado por *Teleop node*, este, é responsável por receber valores de um comando remoto (Wii-Mote). O *Teleop-node*, também, recebe estados de botões do Wii-Mote.

3.2 Hardware

Nesta secção são expostos os actuadores, andares de potência, microcontroladores e outros componentes físicos usados na plataforma ISR-TT. São apresentadas as características, mais relevantes, de cada modulo.

3.2.1 Locomoção

A roda que foi usada nesta plataforma tem o motor embutido no seu eixo de rotação, também conhecida por *In-Wheel-Motor*. Para informação mais detalhada sobre a eficiência e o consumo energético pode se consultar a referência [5] da bibliografia.

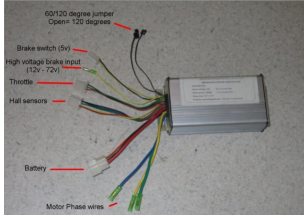


Potência	750 W
Tensão	48 V
Nº Fases	3
Nº Parares de pólos	20
Raio	200 mm
Feedback	3 Sensores de efeito de Hall
Velocidade max.	8 m/s

Tabela 3.1: Roda e as suas características eléctricas e mecânicas

Controlador do motor BLDC

Para controlar um motor BLDC é necessário controlar as fases em função da posição do rotor. Para este efeito, foi usado um controlador já existente. Este controlador oferece várias funcionalidade como o controlo de velocidade através de uma referência em tensão e ,também, permite a regeneração de energia durante a travagem. O sentido de rotação é definido por uma entrada digital.



Sinal de controlo	1V-5V
Sentido de rotação	I/O Digital
Tensão	48V
Feedback	3 sensores de efeito de Hall

Tabela 3.2: Controlador da roda e as suas características

3.2.2 Direcção

Actuador

A orientação de cada roda é efectuada através de uma actuador linear. O actuador linear é constituído por um motor DC com escovas, um êmbolo e um potenciómetro para fazer *feedback* da posição absoluta. Na tabela 3.4 estão patentes as características eléctricas e mecânicas destes actuadores.

Tabela 3.3: Tabela com as características do actuador linear da direcção

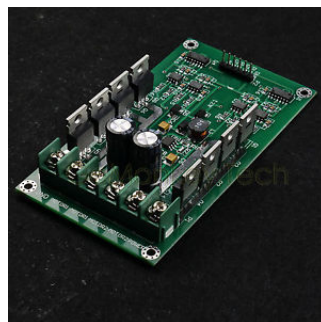
As características eléctricas são:	As características mecânicas são:
Potência = 76.8 W Tensão = 24 V Corrente máxima = 3.2 A Potenciómetro: 10 k Ω Resistência Interna = 497 Ω Indutância Interna = 1.8366 mH	Comprimento do cilindro = 200 mm Máxima carga estática = 2500 N Máxima carga dinâmica = 240 N Velocidade máxima = 22 mm/s

Andar de potência

O andar de potência é o interface entre o microcontrolador e o actuador. O andar de potência, tem o objectivo de adequar o sinal de comando para a gama de funcionamento do actuador. O sinal de entrada é modulado em PWM com uma gama de tensão de 0V a 5V, no entanto é tolerante aos 3.3V. Este actuador tem a capacidade de actuar dois actuadores independentemente. Em termos de potência, este dispositivo, suporta uma tensão máxima de 36V e 10A em regime nominal, podendo ir aos 30A pico para cada actuador.



(a) Actuador linear da direcção.



(b) Andar de potência do actuador da direcção.

Figura 3-3: Actuadores linear da plataforma.

3.2.3 Suspensão

Actuador

A altura da plataforma é regulada pela suspensão. Esta é constituída por uma actuador linear e um amortecedor, patente na figura 2-1b. Tal como o actuador linear da direcção, este, também é constituído por um motor DC com escovas, um embolo e um potenciómetro para efeitos de *feedback*.

Tabela 3.4: Tabela com as características do actuador linear da suspensão.

Características eléctricas:	Características mecânicas:
Potência = 168 W	Comprimento do cilindro = 200 mm
Tensão = 24 V	Máxima carga estática = 4500 N
Corrente máxima = 7 A	Máxima carga dinâmica = 1500 N
Potenciómetro = 10 k Ω	Velocidade máxima = 21 mm/s
Resistência Interna = 1.828 Ω	
Indutância Interna = 10.52 μ H	

Andar de potência

Assim, como o andar de potência do actuador da direcção, este também tem a missão de adequar o sinal de controlo para gama de funcionamento do actuador e tem a capacidade, igualmente, de actuar dois actuadores. O principio de funcionamento é o mesmo, tendo um sinal de entrada modulado em PWM e uma gama de funcionamento de saída de 12 V a 48 V e suporta 60 A de corrente máxima.



(a) Actuador linear da suspensão.



(b) Andar de potência da suspensão.

Figura 3-4: Actuadores linear da plataforma.

3.2.4 Caixa de Controlo

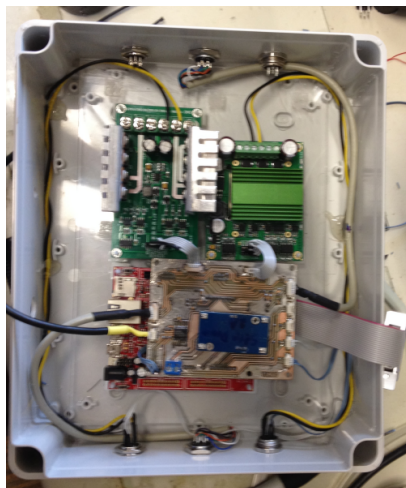


Figura 3-5: Caixa de controlo.

A caixa de controlo foi concebida com o objectivo de tornar o *hardware* mais modular e robusto. Dentro da caixa de controlo foi colocado um microcontrolador, PCB de interface, os *drivers* dos motores e foram fixados conectores com as características eléctricas adequadas para cada actuador. Neste secção serão, porém, abordado unicamente os factores que influenciaram a escolha do microcontrolador e alguns detalhes sobre a PCB de interface.

Microcontrolador

As tarefas de controlo de malha fechada de posição e de velocidade são executadas num microcontrolador. Existem dois microcontrolador na plataforma (eixo frontal e

eixo traseiro). Cada microcontrolador controla dois quadrantes (quadrante esquerdo e quadrante direito). Cada quadrante é constituído por uma roda, um actuador para a direcção e um actuador para suspensão. Tendo, cada microcontrolador, controlar, no total, duas rodas, dois actuadores para a direcção e dois actuadores para a suspensão, fazendo no total 6 actuadores.

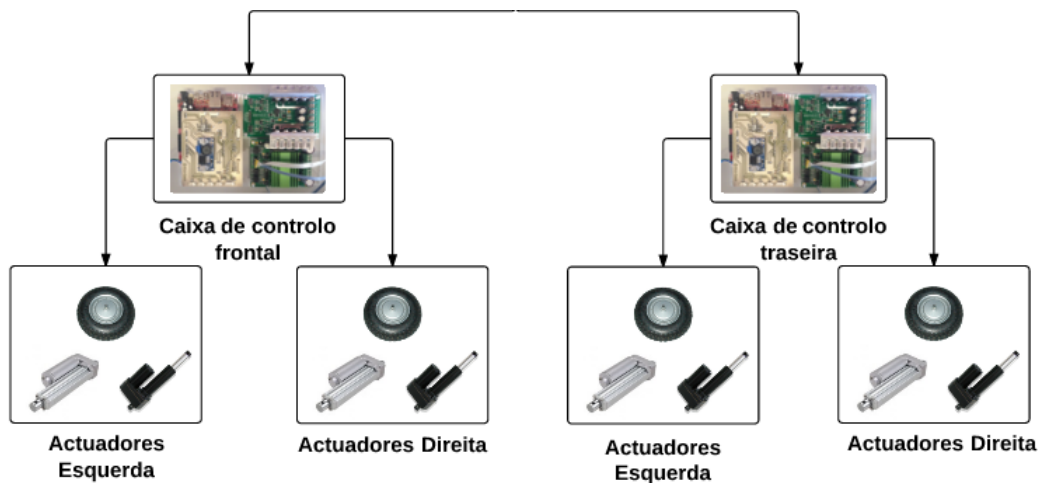


Figura 3-6: Actuadores a controlar

- **PWM** -Cada andar de potência é actuado por um sinal PWM, logo são necessárias seis saídas PWM;
- **Saídas Digitais** - O sentido dos actuadores (rodas e actuadores lineares) são definidos através de sinais digitais. O controlador da roda necessita de um sinal digital. Os andares de potência dos actuadores lineares da direcção necessitam de um sinal digital por actuador. Os andares de potência das suspensões necessitam de dois sinais digitais por actuador. A *transceiver* para converter a comunicação de TTL para RS-485, também necessita de um sinal digital para escrever ou ler na linha. Como existem dois actuadores lineares de direcção, dois actuadores lineares para suspensão e duas roda para cada actuador, são necessárias 9 saídas digitais;
- **Interrupções Externas** - Para cada sensor de efeito de Hall é atribuída uma entrada configurada como interrupção externa. Como cada microcontrolador

controla duas rodas e cada roda têm três sensores de efeito de Hall (um para cada fase), são necessárias seis entradas configuradas como interrupções externas.

- **ADC** - O *feedback* de posição dos actuadores lineares (d direcção e suspensão) é feito através de um sinal analógico, como cada microcontrolador controla quatro actuadores lineares (dois da direcção e dois da suspensão) são necessárias, ao total, quatro entradas configuradas em modo ADC.
- **TTL** - O microcontrolador necessita de comunicar para enviar e receber dados, para isso usa uma entrada para receber (RX) e uma saída para enviar (TX) dados.

Na tabela 3.5 são apresentados as entradas e saídas necessárias para controlar os actuadores.

Tabela 3.5: Tabela com entradas e saídas necessárias

PWM	Saídas Digitais	Interrupções Externas	ADC	TTL
6	9	6	4	1

Tendo em conta as entradas e saídas necessárias, a informação, exemplos de códigos existente e o custo, optou-se por uma placa de desenvolvimento da *OLIMEX* [17], com um microcontrolador *STM32F407ZGT6* da família 32bit ARM CortexTM-M, fabricado pela *STMicrocontrollers*. Para mais informações pode-se consultar o manual da placa de desenvolvimento [17].



Figura 3-7: Microcontrolador STM32-E407.

PCB de Interface

A PCB foi desenvolvida com os objectivos de robustecer as ligações entre componentes e acondicionar os sinais de entrada e de saída através de electrónica adicional. A PCB patente na figura 3-8b possui, sobre tudo, conectores para fazer interface entre o microcontrolador e os actuadores, porém, também possui filtros passa-baixo para filtrar o ruídos electromagnético, possui um *transceiver* SN75176AP [12] da Texas Instruments para converter a comunicação de TTL para RS-485 e um conversor ajustável DC/DC para reduzir a tensão de entrada de 24 V para 5 V.

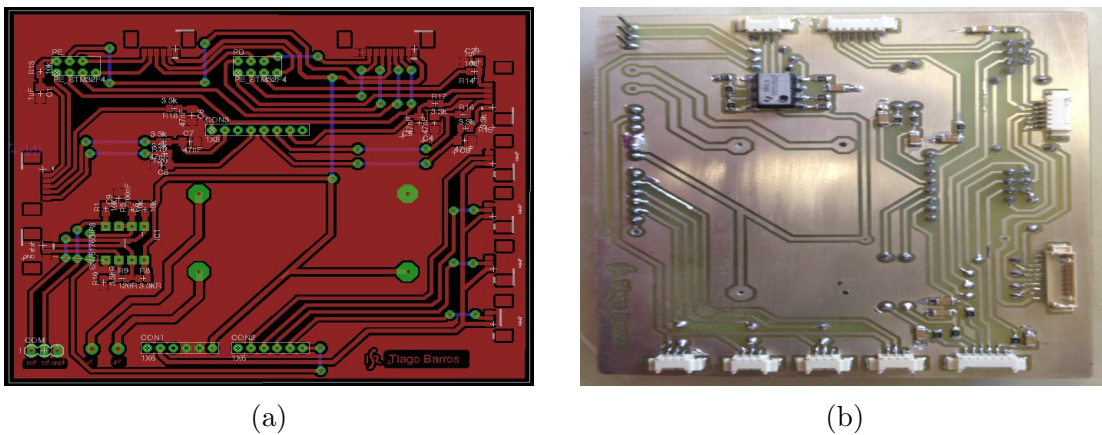


Figura 3-8: PCB na fase desenvolvimento (a) e na fase final (b)

Filtro Passa-Baixo - Sensores de Efeito de Hall

Os sinais do oriundos dos sensores de efeito de Hall são induzidos com ruído de alta frequência devido às comutações dos BLDCs.

A frequência do ruído é $f_{ruído} > 10 kHz$, com a seguinte equação obtém-se a frequência de corte f_{corte} :

$$f_{corte} = \frac{f_{ruído}}{10} \quad (3.1)$$

$$R = \frac{1}{2\pi \cdot f_{corte} \cdot C} \quad (3.2)$$

A equação 3.2 tem duas incógnitas (R,C) é necessário, portanto, atribuir um

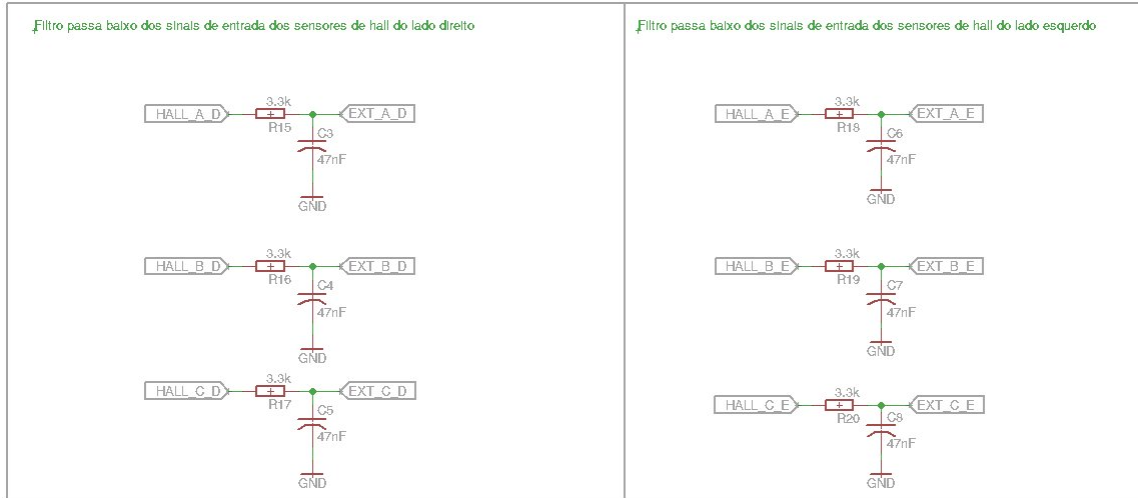


Figura 3-9: Cricuito electrico dos filtros passa-baixo.

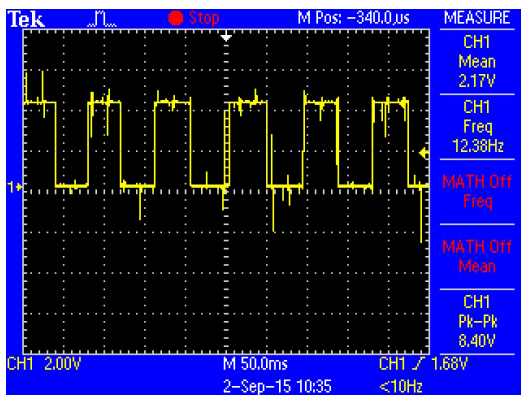
valor a uma das incógnitas e calcular a segunda usando a equação 3.2. Os valores dos condensadores são mais limitados. É preferível definir uma capacidade $C = 47 \text{ nF}$ e calcular a resistência, usando a resistência com o valor mais próximo.

$$R = \frac{1}{2\pi \cdot 10^3 \cdot 47 \cdot 10^{-9}} = 3.3 \text{ k}\Omega \quad (3.3)$$

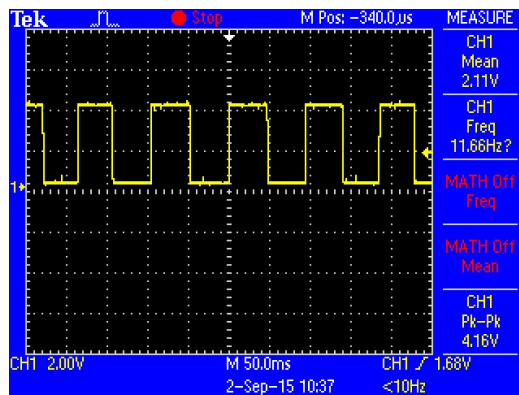
A resistência usada foi $3.3 \text{ k}\Omega$

Nas figuras 3-10a e 3-10b pode-se constatar a melhoria causada pelos filtros, onde a primeira figura representa o sinal dos sensores de efeito de Hall sem filtro e a segunda representa, os mesmos, com filtro.

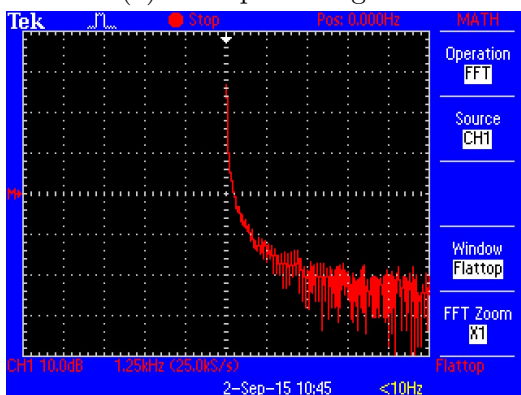
Nas figuras 3-10c e 3-10d estão representadas as Transformadas Rápidas de Fourier (FFT) dos sinais dos sensores de efeito de Hall sem e com filtro passa-baixo, respectivamente. Da primeira para a segunda figura é notório que as componentes de alta frequência são mais atenuadas.



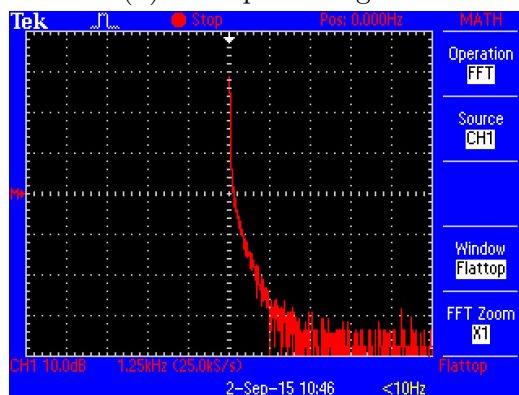
(a) Sinal pré filtragem.



(b) Sinal pós filtragem.



(c) FFT pré filtragem.



(d) FFT pós filtragem.

Figura 3-10: Sinais dos sensores de efeito de Hall.

Filtro Passa-Baixo - Referência de Velocidade

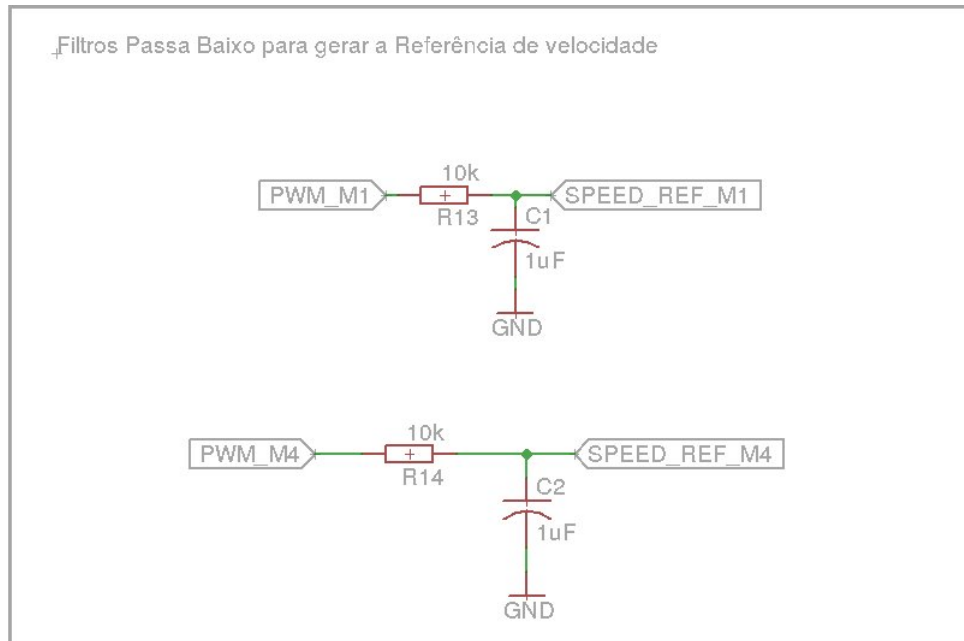


Figura 3-11: Filtro passa-baixo para converter PWM num valor analógico.

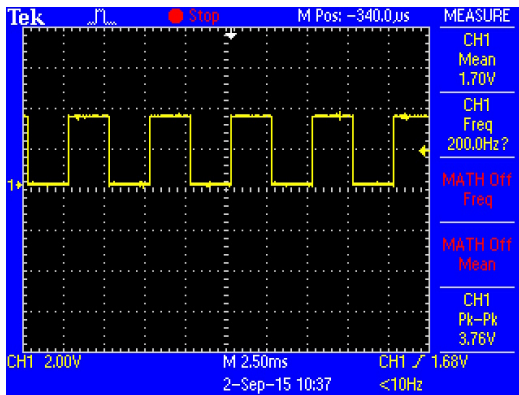
Usando a equação 3.2 com $f_{corte} = 15 \text{ Hz}$ e $C = 1 \cdot 10^{-6} \text{ F}$

$$R = \frac{1}{2\pi \cdot 15 \cdot 1 \cdot 10^{-6}} = 10.6 \text{ k}\Omega \quad (3.4)$$

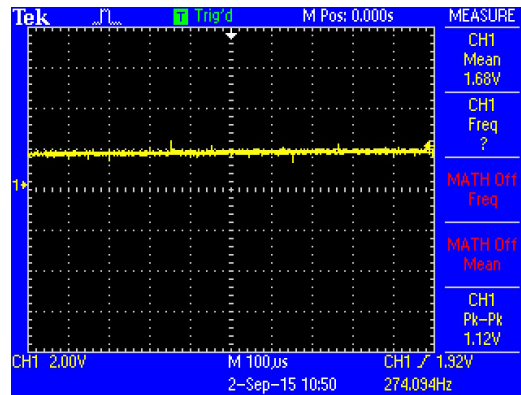
O valor da resistência usado foi $10 \text{ k}\Omega$

Nas figuras 3-12a e 3-12b estão representados os sinais antes do filtro e depois do filtro respectivamente.

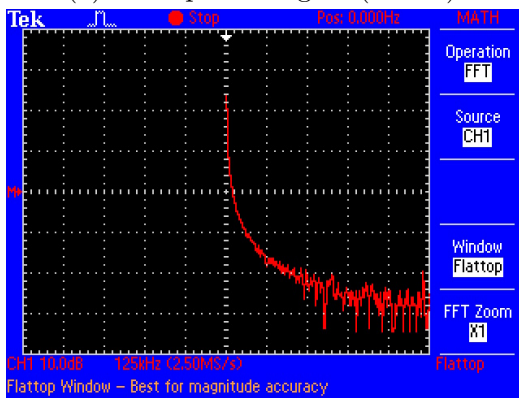
As figuras 3-12a e 3-12b representam o sinal PWM gerado pelo microcontrolador e o mesmo sinal filtrado por um filtro passa-baixo, respectivamente. As figuras 3-12a e 3-12b ilustram as transformadas de Fourier do sinal PWM e o mesmo sinal filtrado. Nestas figuras é observável que no segundo caso só existe uma componente DC.



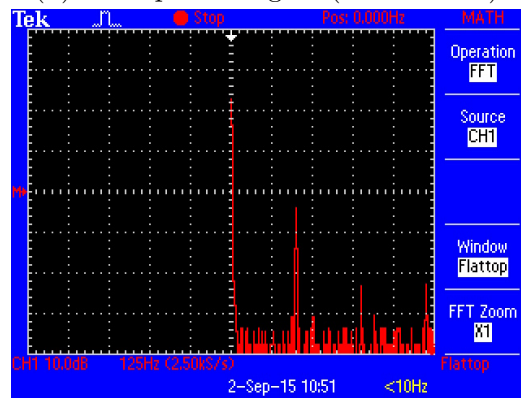
(a) Sinal pré filtragem (PWM).



(b) Sinal pós filtragem (valor médio).



(c) FFT do sinal pré filtragem.



(d) FFT do sinal pós filtragem.

Figura 3-12: Sinal para referência de velocidade do BLDC.

Conversor TTL para RS-485

As designações TTL e RS-485 referem-se aos níveis de tensão em que estes métodos de comunicação operam. Estes dois métodos de comunicação são considerados métodos de comunicação série. A comunicação TTL é *full-duplex*, isto é, pode-se escrever e ler simultaneamente. Em termos de tensão a comunicação opera com níveis de tensão entre os 0V e os 5V, o que a torna susceptível a ruído. Em termos de tipologia é uma comunicação *single-point*, isto significa só pode estar ligado fisicamente com um dispositivo. Este tipo de comunicação é feito para distâncias curtas.

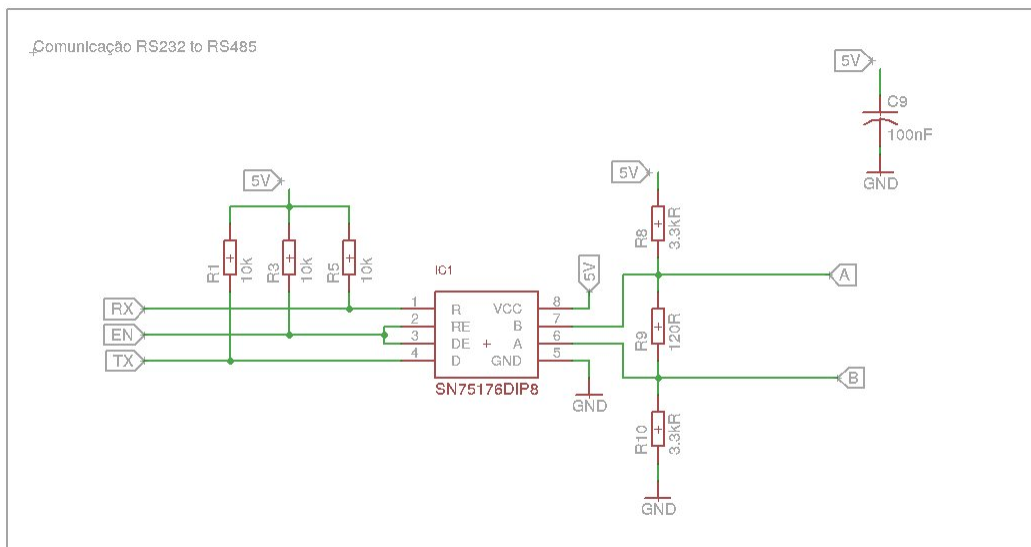


Figura 3-13: Circuito eléctrico da conversão da comunicação de TTL para RS-485 e vise-versa.

A comunicação RS-485 é *half-duplex* (com a configuração dois fios) isto significa, e ao contrário da comunicação TTL, só pode escrever ou ler alternadamente. Em termos físicos os sinais são diferenciais sendo a gama de tensão (neste caso) entre 5V e -5V, o que a torna mais imune a ruído. A tipologia é multi-ponto, ou seja, podem estar vários dispositivos ligados à mesma rede, no entanto, só pode escrever um de cada vez. Este tipo comunicação, devido às suas características, é muito usado na indústria onde existe muito ruído electromagnético [15].

3.2.5 Caixa de Potência e Comunicação

Na caixa de potência e comunicação, concentram-se todas as ligações de potência, que alimentam as caixas de controlo e as rodas. É, também, nesta caixa, que existe um botão de emergência para cortar a alimentação de todos os dispositivos em caso de emergência.

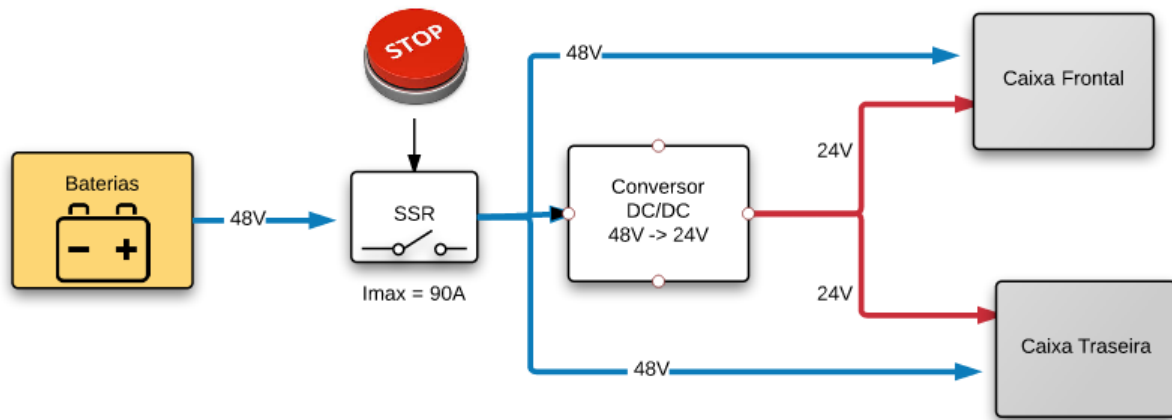


Figura 3-14: Circuito de potência.

O conversor DC/DC tem como objectivo baixar o nível de tensão de 48 V para 24 V. Esta solução não é a mais adequada, pois existem perdas de conversão na ordem de 80% no conversor.

A solução ideal seria usar uma linha de alimentação única, ou seja, alimentar os anadres de potência com 48 V e usar apenas 50% da gama de funcionamento dos sinais de PWM que os controlam, por forma, a ter uma alimentação média de, no máximo, 24 V nos actuadores.

Na caixa existe um conversor RS-485 para USB para fazer de interface entre o computador e os microcontroladores. A tipologia recomendada para a comunicação RS-485 é *daisy-chain* [15], no entanto, optou-se por usar uma tipologia estrela, devido à curta distância entre caixas e a disposição das mesmas.

3.2.6 Comando Remoto

Um comando remoto é uma ferramenta importante para qualquer robô móvel. O comando remoto escolhido para este projecto foi o Wii-Mote, desenvolvido pela Nintendo para a consola Wii. Este dispositivo é bastante interessante, devido ao facto de ter uma porta de expansão que permite a ligação a outros acessórios.







O Wii-Mote incorpora diversas tecnologias, tais como acelerómetros e uma matriz óptica (*optical array*) que permite a detecção de movimento e posição. Este comando também possui botões e um *joystick*. Este conjunto de opções de controlo permite a implementação de varias funcionalidades.



Figura 3-15: Comando Wii Mote.

O comando comunica com o PC através de *Bluetooth*. Para ler os valores do comando é usado um *device driver* implementado em ROS. As setas controlam o ângulo da direcção, enquanto que as setas verticais controlam o sentido de locomoção. Os botões 1 e 2 têm a função de definir a geometria de condução. O botão 1 faz com que a plataforma conduza com a condução *Ackerman* e o botão 2 implementa a condução *Ackerman* de quatro rodas. Os botões + e - aumentam e diminuem a velocidade de locomoção. O botão *B* tem que ser pressionado sempre que se quiser mover a plataforma ou aumentar a velocidade das rodas.

Tabela 3.6: Funcionalidades implementadas no Wii-Mote

	Ângulo de viragem
	Sentido de marcha
	Condições iniciais (rodas alinhadas e velocidade definida por defeito)
	Ajusto de velocidade
	Condução <i>Ackerman</i>
	Condução <i>Ackerman</i> quatro rodas

3.3 Firmware

O *firmware* é código, concebidas para um dispositivo com a missão de interagir com *hardware*. Para este projecto foi desenvolvido um *firmware* para controlar seis actuadores e comunicar com o *software*. Para desenvolver *firmware* recorreu-se a um núcleo de tempo-real (*Real-Time Operating System* - RTOS) denominado por ChibiOs. Foi desenvolvido um protocolo de comunicação, assim como, implementado funções periódicas de controlo de posição e velocidade. As especificações do projecto requereram a utilização de periféricos como ADC, PWM, Interrupções Externas, comunicação série e saídas digitais.

3.3.1 Núcleo de Tempo Real

Os sistemas operativos de tempo real têm como objectivo garantir que o sistema tenha uma resposta a um acontecimento externo dentro de um intervalo de tempo limitado e previamente especificado.

Qualquer núcleo de sistema operativo multi-tarefa deve disponibilizar 3 funções:

- **Escalonamento de processos ou tarefas (Scheduling)**; determina qual a tarefa seguinte que tomará conta do CPU (será executada)
- **Lançamento ou despacho de processos ou tarefas (Dispatching)** Faz a comutação de contexto e outras operações necessárias para retirar uma tarefa de execução e substituir por outra.
- **Comunicação (e sincronização) entre processos ou tarefas**

Um sistema operativo pode ser representado por camadas de abstracção desde o hardware até ao interface com as aplicações.

Optou-se por um núcleo que já fornecesse suporte de software, em termos de interface para os periféricos da placa de desenvolvimento da OLIMEX e que, desta forma, oferecesse uma camada de abstracção que acelerasse o desenvolvimento do *firmware*. A documentação, exemplos de códigos disponíveis e a condição de ser *freeware* foram factores preponderantes na escolha.

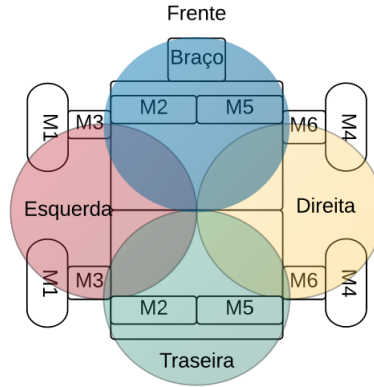


Figura 3-16: Disposição dos actuadores na plataforma para o *firmware*.

3.3.2 Tarefas para Controlo de Rodas

As tarefas M1 e M4 são responsáveis pelo controlo de velocidade das rodas esquerda e direita respectivamente.

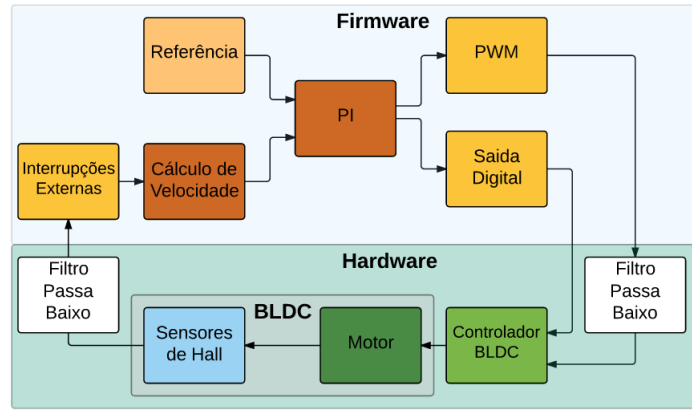


Figura 3-17: Diagrama de Blocos do funcionamento do *firmware* para as rodas.

Na figura 3-17 está representado o esquema de funcionamento destas tarefas. Sempre que ocorre um evento numa das entradas configuradas como interrupções externas uma função *callback* é invocada. Nesta função é medida a diferença temporal entre o instante actual e a ultima chamada Δt . Sabendo a resolução em radianos entre dois sinais consecutivos do sensores de efeito de Hall $\Delta\alpha$, consegue-se calcular a velocidade angular instantânea da roda ω_r .

$$\omega_r = \frac{\Delta\alpha}{\Delta t} \quad (3.5)$$

Para conseguir calcular uma boa estimativa da velocidade $\Delta t < P_{tarefa}$, ou seja, o tempo entre interrupções tem que ser sempre menor que o período da tarefa para obter pelo menos uma leitura por ciclo. Daqui deduzimos a seguinte condição:

$$\omega_{min} = \frac{\Delta\alpha}{P_{tarefa}} \quad (3.6)$$

$$\omega_{min} < \omega_r < \omega_{max} \quad (3.7)$$

Sendo $\alpha = 0.104 \text{radianos}$ e $P_{tarefa} = 250 \text{ms}$, substituindo os valores na equação 3.6 obtém-se:

$$\omega_{min} = \frac{0.104}{0.25} = 0.416 \text{ [rad/s]} \quad (3.8)$$

Tendo a velocidade angular é trivial calcular a velocidade linear da roda.

$$v_r = R \cdot \omega_r \quad (3.9)$$

Sendo $R = 200 \text{mm}$ o raio da roda e substituindo na equação 3.9 obtém-se o seguinte resultado para a velocidade linear mínima:

$$v_{min} = 0.20 \cdot 0.416 = 0.08 \text{ [m/s]} \quad (3.10)$$

O valor obtido da igualdade 3.10 é um valor teórico, pois na realidade a velocidade mínima que as rodas conseguem rodar, de forma contínua, é 0.2 m/s. Em termos de velocidade linear máxima, definiu-se 1 m/s.

Os ganhos para o controlador PI foram afinados por tentativa e erro. Os valores dos ganhos são $K_P = 1$ e $K_I = 0.5$

O controlador das rodas disponibiliza uma entrada analógica em tensão que serve como referência para o controlo de velocidade. A gama de funcionamento desta entrada é de 1 V a 5 V, em que 1 V representa a velocidade mínima e 5 V a velocidade máxima. O *firmware* faz o controlo desta tensão através do *duty cycle* do PWM. Na PCB de interface é feito a filtragem do sinal, de forma, só a componente DC estar

à entrada do controlador das rodas. O sentido de rotação é definido através de uma saída digital em modo NPN colector aberto.

3.3.3 Tarefas para Controlo dos Motores Lineares

As tarefas **M2**, **M3**, **M5** e **M6** controlam a posição da suspensão, direcção do lado esquerdo e direito respectivamente. A figura 3-18 ilustra o esquema de controlo. Estas tarefas implementam a mediana antes de calcular o sinal de controlo, com o objectivo de eliminar ruído que deteriora a precisão do controlo de posição.

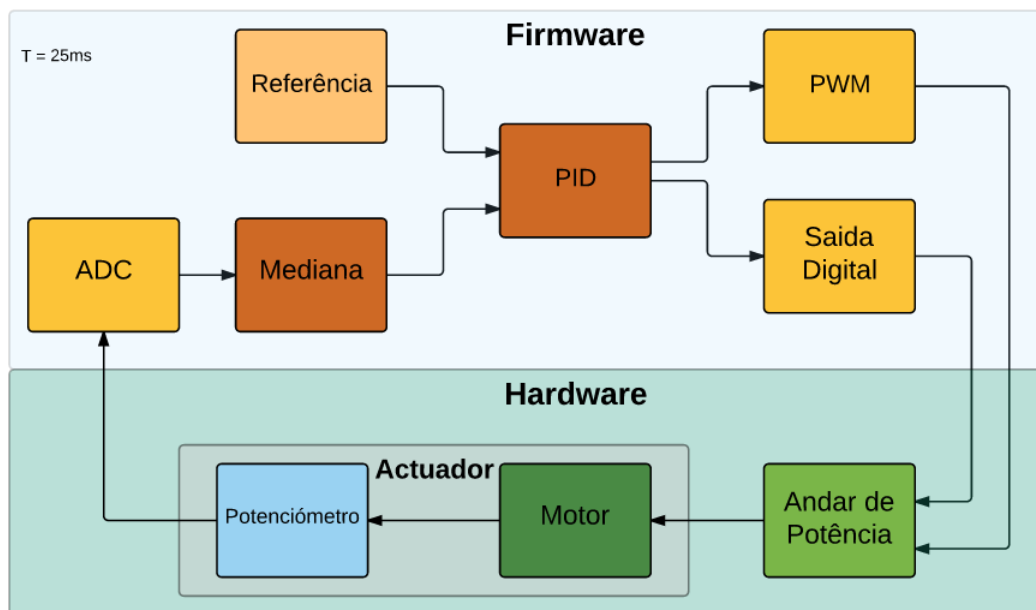


Figura 3-18: Diagrama de Blocos do funcionamento do *firmware* para os actuadores lineares.

A figura 3-16 representa a disposição dos actuadores com a nomenclatura usada no *firmware*.

Os ganhos do controlador PID, também foram afinados manualmente sendo $K_P = 20$, $K_I = 1$ e $K_D = 0.001$

Os andares de potência dos motores lineares das suspensão necessitam de um sinal de PWM para definir a velocidade e duas entrada digitais para definir o sentido. Os andares de potência dos motores lineares da direcção necessitam de um sinal PWM para definir a velocidade e um sinal digital para definir o sentido.

3.3.4 Tarefas de Comunicação

A tarefa *msg_t Recv(void * arg)* recebe mensagens oriundas da camada a montante, decodifica-as em conformidade com o protocolo utilizado(ver apêndice B).

A tarefa *msg_t Send(void * arg)* encapsula as mensagens com os parâmetros adequados em função da origem e envia-as para a camada a montante.

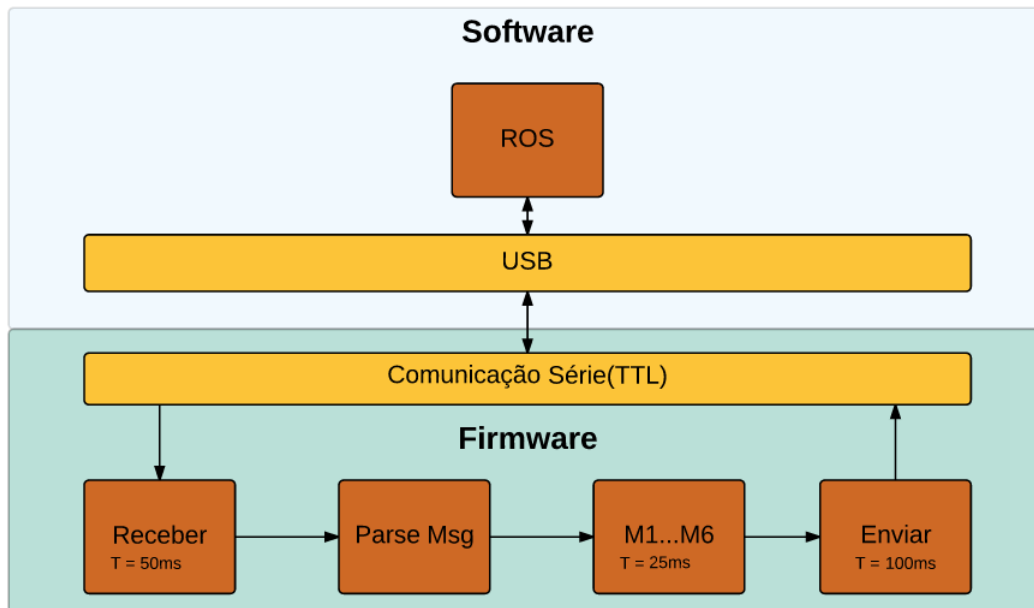


Figura 3-19: Diagrama de blocos da interação entre o *firmware* e *software*.

As duas funções descritas anteriormente, são funções de recepção e de envio de mensagens respectivamente. A figura 3-19 apresenta funcionamento da forma como estas duas funções interagem com a camada superior.

3.3.5 Protocolo de comunicação

O protocolo é uma ferramenta que permite que o *firmware* consiga interagir com o *software* e, desta forma, trocar informação e configurar parâmetros em tempo real.

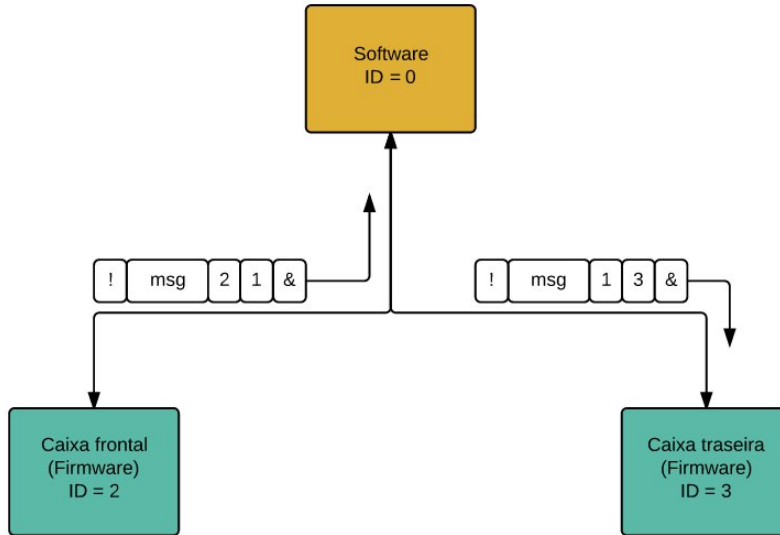


Figura 3-20: Diagrama de blocos do funcionamento do protocolo de comunicação entre camadas.

Estrutura

A estrutura da mensagem foi concebida com o intuito, de facilmente, ser possível adicionar funcionalidades. A mensagem começa com um caracter de inicialização “&” e acaba com um caracter de terminação “!”, estes foram escolhidos de forma a estarem o mais distante possível dos caracteres numéricos. Desta forma, facilmente é detectável o erro na mensagem. O primeiro campo destina-se a identificar o destinatário “D”. O segundo campo identifica o remetente “SR”. Com esta informação o destinatário consegue responder com o parâmetro pretendido. O terceiro campo identifica a tarefa “T” em causa, estas podem ser M1, M2, M3, M4, M5 ou M6. O quarto campo define a função que será executada ou foi executada “C”. O quinto campo indica o estado (ligada ou desligada), da tarefa em causa, “S”. O sexto campo tem funcionalidades diferentes em função do comando em causa “Value”. O protocolo está implementado em ASCII.

&	D	SR	T	C	S	V	!
0	1	2	3	4-5	6	7-14	15

Figura 3-21: Estrutura da mensagem do protocolo de comunicação.

Funcionalidades

As funcionalidades implementadas, tanto no *firmware* como no *software*, foram concebidas, de forma, a irem ao encontro dos requisitos necessários para testar os actuadores, individualmente e de forma integrada.

Ao longo do projecto foram implementadas novas funcionalidades, de forma, a otimizar os recursos e diminuir o tráfego na rede. Inicialmente, tencionou-se usar uma mensagem para o controlo de cada actuador individualmente, porém, na fase final conclui-se que este procedimento levaria a uma elevada carga de processamento nos microcontroladores. Por essa razão, implementou-se o controlo de quatro actuadores simultaneamente através de uma mensagem. Este procedimento reduz substancialmente o tráfego e aumenta a sincronização no início do controlo dos actuadores.

Os grupos foram agregados em quatro elementos do mesmo tipo, isto é, os quatro actuadores das suspensões são accionados através da mesma mensagem, os quatro actuadores da direcção foram agregados na mesma mensagem, assim como os BLDCs das rodas são controlados com uma única mensagem. Este procedimento é útil, principalmente, na teleoperação. Quando os actuadores são accionados através do *software* desenvolvido em matlab os actuadores são controlados com mensagens individualmente.

Tabela 3.7: Descrição dos campos da mensagem do protocolo de comunicação.

Posição	Símbolo	Designação
0	&	Caractere de iniciação
1	D	ID do destinatário
2	SR	ID do Emissor
3	T	ID da tarefa
4-5	C	Comando
6	S	Estado
7-14	Value	Caracteres úteis
15	!	Caractere de terminação

Tabela 3.8: Funcionalidades implementadas

0	Start/Stop
1	Targuet Value
2	Data Request
3	Gains
4	Who is there?
5	Error
6	Start/Stop all the threads
7	Control Mode

Tabela 3.9: ID dos dispositivos

0	Broadcast
1	PC
2	Caixa de controlo frontal
3	Caixa de controlo traseira

Tabela 3.10: ID das tarefas

0	Main
1	M1
2	M2
3	M3
4	M4
5	M5
6	M6

3.4 Software

Nesta secção é descrito o desenvolvimento da camada hierarquicamente mais elevada (*software*) e os módulos que a integram.

A principal missão desta camada é receber valores de velocidades linear e angular, assim, como, a altura da plataforma de um comando remoto e converter-los, em função da geometria de condução (Ackerman ou Ackerman 4 rodas), em unidades dos respectivos actuadores. Após a conversão, estes valores são enviados para os microcontroladores.

A implementação desta camada foi inteiramente desenvolvida usando um *framework* denominado por ROS. O comando remoto usado, foi o Wii-Mote da Nitendo.

Nesta camada o requisitos de tempo real não são tão rígidos comparativamente ao *firmware*, sendo as tarefas aqui executadas denominadas como *soft real-time tasks*.

3.4.1 ROS

Em projectos com alguma dimensão o *software* tem que ser bem estruturado, por forma, a ser facilmente mantido e actualizável. Idealmente, o *software* deveria ser reutilizável noutros projectos para não ser necessário “reinventar” a roda repetidamente. Os *frameworks* são concebidas para oferecer ao programador flexibilidade, eficiência, escalabilidade, robustez, características de tempo-real, interoperabilidade, portabilidade e reutilização de código [2].

O ROS é um *framework* amplamente usado na robótica, cuja filosofia principal é a criação de pedaços de código que funcionem noutros robôs com alterações mínimas. Esta filosofia, permite que sejam criadas funcionalidades genéricas, sejam partilhadas entre utilizadores e reutilizadas noutros robôs. ROS disponibiliza funcionalidades como abstracção do *hardware*, controlo de dispositivos de baixo-nível, partilha de mensagens entre processos. A arquitectura é baseada numa tipologia *graph* centralizada e o processamento é feito em nós. As bibliotecas de *software* são desenvolvidas em sistemas Unix [18].

3.4.2 Hierarquia

A hierarquia escolhida para esta camada, à semelhança das outras camadas, foi a de processamento distribuído por tarefas/nós que executam tarefas simples com objectivos bem definidos. Assim, consegue-se criar um sistema complexo, porém, modular e versátil na reutilização do código.

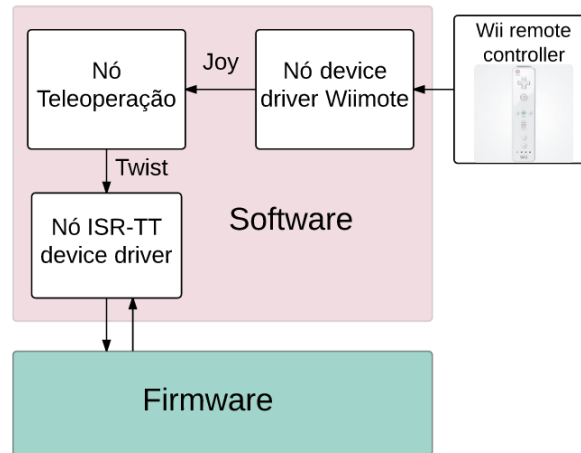


Figura 3-22: Diagrama de blocos do funcionamento do software.

O nó *ISR-TT device driver* e *device driver Wiimote* são os nós que estão hierarquicamente mais baixos, devido ao facto de interagirem directamente com *hardware*. O nó *Teleoperação* é a tarefa hierarquicamente mais elevado.

3.4.3 Nó ISR-TT

O nó *ISR-TT device driver* tem como objectivo receber velocidades lineares, velocidades angulares e altura do nó *Teleoperação* em relação ao centro de massa da plataforma e converte-las em conformidade com a geometria de condução em velocidades e posição para as rodas e actuadores lineares respectivamente. A plataforma *ISR-TT* tem a capacidade de executar várias geometrias de condução, no entanto só foram implementadas as geometrias *Ackerman* e *Ackerman* de quatro rodas. Para além das geometrias de condução este nó mapeia os ângulos da direcção das rodas nas unidades dos actuadores, que neste caso têm uma gama de funcionamento entre 0 e 255, onde 0 representa o êmbolo encolhido e 255 o embolo completamente esticado.

Este mapeamento é feito através de uma função de transferência calculada através de um processo de calibração executado para cada roda. Mais pormenores sobre o processo de calibração estão no apêndice C.2.

Condução de *Ackerman*

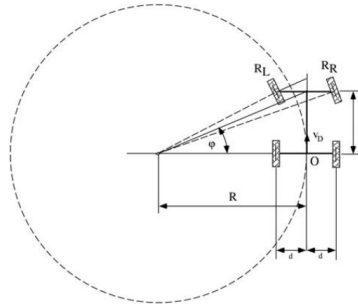


Figura 3-23: Representação dos parâmetros da condução de *Ackerman* [2].

Para implementar a condução de *Ackerman* é necessário obter os ângulos $\varphi_{x,y}$ para $(x, y) = \{(f, e), (f, d)\}$, isto é os ângulos para as rodas frontais da esquerda e da direita. É necessário, também, obter as velocidades $v_{x,y}$ para $(x, y) = \{(f, e), (f, d), (t, e), (t, d)\}$

Através da figura 3-23 conseguimos deduzir as equações necessárias para obter os ângulos e as velocidades.

Sabendo, que as variáveis de controlo são as velocidades linear v e angular ω deduzimos as seguintes equações:

$$h = \frac{v}{\omega} \quad (3.11)$$

$$\varphi = \arcsin\left(\frac{l}{h}\right) \quad (3.12)$$

$$R = \frac{l}{\tan(\varphi)} \quad (3.13)$$

Através das equações anteriores é recuperado o ângulo da viragem φ e o raio ao centro de curvatura R , onde h é a distância entre o centro de curvatura e o ponto

central do eixo frontal. Com estas variáveis conseguem-se obter as equações para o ângulo da roda esquerda $\varphi_{f,e}$ e direita $\varphi_{f,d}$

$$\varphi_{f,e} = \arctan\left(\frac{L}{R-d}\right) \quad (3.14)$$

$$\varphi_{f,d} = \arctan\left(\frac{L}{R+d}\right) \quad (3.15)$$

Sabendo os ângulos de cada roda é trivial deduzir as equações para obter as velocidades para cada roda.

$$h_e^2 = l^2 + (R-d)^2 \leftrightarrow h_e = \sqrt{l^2 + (R-d)^2} \quad (3.16)$$

$$h_d^2 = l^2 + (R+d)^2 \leftrightarrow h_d = \sqrt{l^2 + (R+d)^2} \quad (3.17)$$

$$v_{f,e} = h_e \omega = \frac{\sqrt{l^2 + (R-d)^2} \cdot |\tan(\varphi)|}{l} \cdot v \quad (3.18)$$

$$v_{f,d} = h_d \omega = \frac{\sqrt{l^2 + (R+d)^2} \cdot |\tan(\varphi)|}{l} \cdot v \quad (3.19)$$

$$v_{t,e} = (R-d)\omega = \frac{R-d}{R}v \quad (3.20)$$

$$v_{t,d} = (R+d)\omega = \frac{R+d}{R}v \quad (3.21)$$

Condução de *Ackerman* de quatro rodas

Na condução de *Ackerman* de quatro rodas, todas as rodas são orientáveis. Ao contrário da condução de *Ackerman* simples, onde o conjunto de soluções representa a recta que passa pelo eixo traseiro. Na condução de *Ackerman* de quatro rodas o conjunto de soluções representa uma área, por forma, a simplificar o problema, restringiu-se o conjunto de soluções para a recta que intersecta o centro da plataforma.

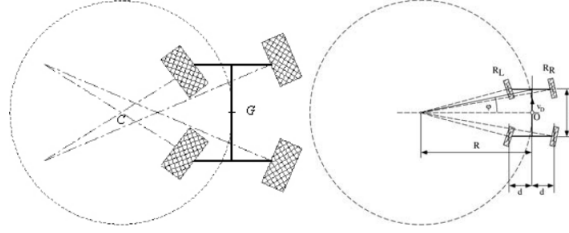


Figura 3-24: Representação dos centros de curvatura para os dois eixos [2].

Da figura 3-24 deduz-se as seguintes equações para calcular os ângulos das quatro rodas:

$$\varphi_{f,e} = \arctan\left(\frac{l/2}{R-d}\right) \quad (3.22)$$

$$\varphi_{f,d} = \arctan\left(\frac{l/2}{R+d}\right) \quad (3.23)$$

$$\varphi_{t,e} = \frac{\pi}{2} - \arctan\left(\frac{l/2}{R-d}\right) \quad (3.24)$$

$$\varphi_{t,d} = \frac{\pi}{2} - \arctan\left(\frac{l/2}{R+d}\right) \quad (3.25)$$

As velocidades para as respectivas rodas são obtidas com as seguintes equações:

$$v_{f,e} = \frac{\sqrt[2]{\frac{l^2}{4} + \left(\frac{R}{2} - d\right)^2} \cdot 2|\tan(\varphi)|}{l} \cdot v \quad (3.26)$$

$$v_{f,d} = \frac{\sqrt[2]{\frac{l^2}{4} + \left(\frac{R}{2} + d\right)^2} \cdot 2|\tan(\varphi)|}{l} \cdot v \quad (3.27)$$

$$v_{t,e} = \frac{\sqrt[2]{\frac{l^2}{4} + \left(\frac{R}{2} - d\right)^2} \cdot 2|\tan(\varphi)|}{l} \cdot v \quad (3.28)$$

$$v_{t,d} = \frac{\sqrt[2]{\frac{l^2}{4} + \left(\frac{R}{2} + d\right)^2} \cdot 2|\tan(\varphi)|}{l} \cdot v \quad (3.29)$$

Mapeamento

O método usado para converter os ângulos das rodas para a escala dos actuadores, consistiu em fixar uma bússola electrónica em cima da roda e fazer deslocações equidistantes com o actuador. Para cada deslocação do actuador foi registado o valor angular da bússola. Após de registar um conjunto de valores considerado como representativos para modular o movimentos da direcção, foram criados gráficos (Apendice C.2) . Os gráficos mostram que as funções são de natureza linear. As respectivas equações foram geradas a partir dos gráficos. Estas funções foram à posteriori implementadas no *software*. De seguida é apresentado a formulação matemática.

$$Z_{x,y} = f_{x,y}(\alpha_{x,y}) \quad (3.30)$$

$$x = \{d, e\} \vee y = \{f, t\}$$

Em que o parâmetro x representa as rodas esquerda (e) e direita (d) e y representa as rodas frontais (f) e traseiras (t). $\varphi_{x,y}$ é o ângulo de cada roda medido em radianos e $Z_{x,y}$ representa a posição de cada actuador.

3.4.4 Nó de Teleoperação

O nó *Teleoperação* subscreve a um tópico *Joy* que disponibiliza os estados e valores do comando Wii-Mote. Após receber os valores, estes, são convertidos numa escala física. Sendo a velocidade linear v em [m/s] e o ângulo de viragem φ em radianos. Com estes valores é calculado a velocidade angular ω

$$R = \frac{l}{\tan \varphi} \quad (3.31)$$

$$\omega = \frac{v}{R} \quad (3.32)$$

Tendo a velocidade linear e angular, o nó publica estes valores num tópico genérico chamado *cmd_vel(geometry_msgs/Twist)*.

Capítulo 4

Testes e Resultados

4.1 Métricas de Avaliação

O desempenho de uma camada depende directamente da camada a jusante e essa depende da camada que a suporta e assim sucessivamente até à camada física, que neste caso são os actuadores. Toda a estrutura de uma plataforma móvel, depende do desempenho dos actuadores ,portanto, é crucial ter indicadores qualitativos do desempenhos em termos de erro (*steady state error*) das rodas, da direcção e da suspensão.

4.1.1 Locomoção

Na figura 4-1 é apresentado um gráfico relativamente ao controlo em malha fechada da velocidade da roda do ISR-TT. No eixo vertical a escala está em metros por segundo (m/s), enquanto que no eixo horizontal a escala está numa base temporal, cuja unidade é segundos (s).

Neste gráfico é perceptível que o controlador da roda apresenta dificuldades em seguir a referência de entrada, apresentando erros na ordem dos 0.3 m/s. Numa escala de 0 a 1 m/s o erro representa 30 %. Esta variação no comportamento da roda é devido a vários factores, nomeadamente à baixa resolução na detecção da posição e a imperfeições físicas (e.g falta de calibração) na própria roda em rotação, estes

distúrbios são visualmente perceptíveis. Outro aspecto que não passa despercebido no gráfico é a perda de sinal para baixas velocidades, isto leva a que a velocidade mínima seja definida acima dos 0.2 m/s.

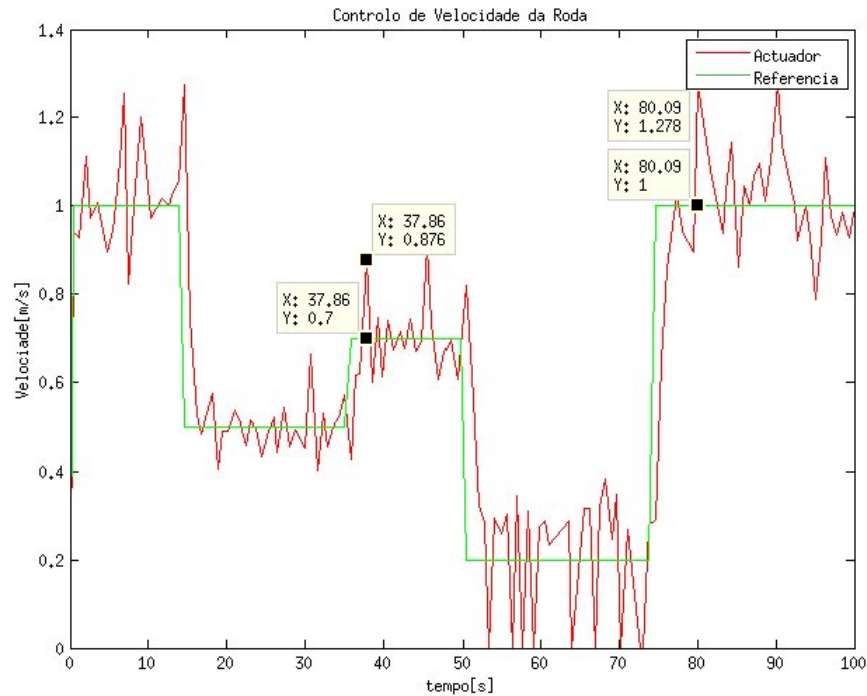


Figura 4-1: Gráfico do controlo de velocidade da roda.

4.1.2 Direcção

Na figura 4-2 é apresentado o gráfico do controlo de posição em malha fechada dos actuadores da direcção em vazio. No eixo vertical é apresentado a escala do actuador com uma gama de funcionamento [0,255], onde 0 representa o êmbolo totalmente encolhido e 255 representa o êmbolo totalmente esticado. Porém no gráfico a escala está entre 0 e 255/1000. O eixo horizontal está dividido em segundos.

Na figura 4-2 verifica se que o controlador tem uma boa resposta em regime final, porém o controlador não consegue acompanhar a referência em regime de transição. É preciso referenciar que o instante em que a referência transita para outro valor no gráfico, representa o instante em que o valor é enviado para o microcontrolador, por isso existe um atraso entre o instante do gráfico e o instante em que o controlador

recebe a referência. Contudo, este atraso não é crítico, já que estamos perante um sistema lento.

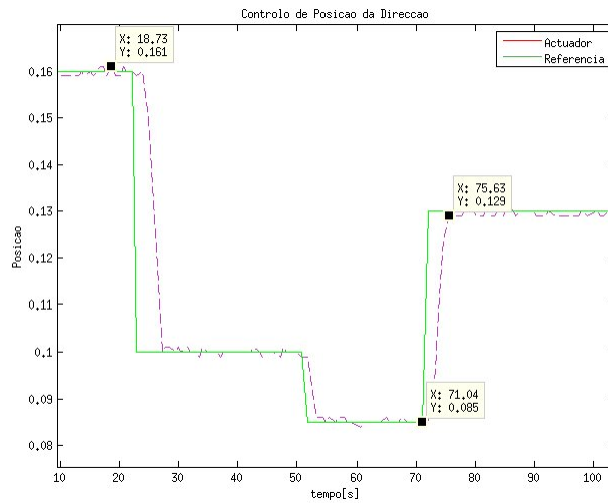


Figura 4-2: Controlo de posição do actuador da direcção.

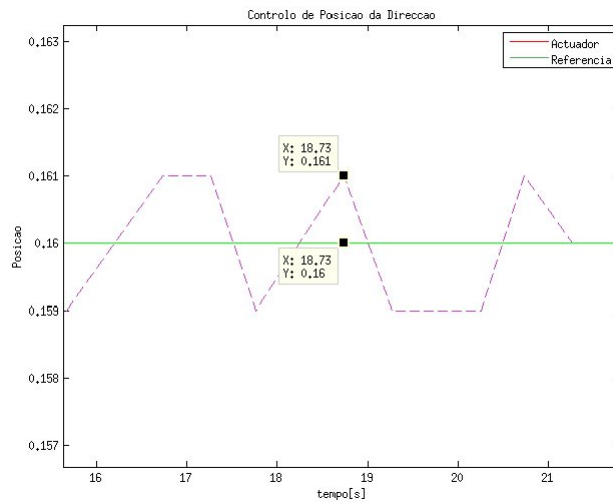


Figura 4-3: Controlo de posição do actuador da direcção com ampliação.

A figura 4-3 é uma ampliação do ponto (18,73,0,161), e através deste gráfico pode-se inferir que o erro em torno da referência é 0,3%.

$$erro = \frac{161 - 160}{255} = \frac{1}{255} = 0,3\% \quad (4.1)$$

4.1.3 Suspensão

Na figura 4-4 é apresentado o gráfico do controlo de posição em malha fechada dos actuadores da suspensão. No eixo vertical é apresentado a escala do actuador com uma gama de funcionamento [0,255], onde 0 representa o êmbolo totalmente encolhido e 255 representa o êmbolo totalmente esticado. Porém no gráfico a escala está entre 0 e 255/1000. O eixo horizontal está dividido em segundos.

Na 4-4 verifica se que o controlador tem uma boa resposta em regime final, porém o controlador não consegue acompanhar a referência em regime de transição. É preciso referenciar que também aqui o instante em que a referência transita para outro valor no gráfico, não representa o instante que o controlador recebe a referência.

A figura 4-5 é uma ampliação do ponto (65.4 , 0.05), e através deste gráfico pode-se inferir que o erro em torno da referência é 0.7%.

$$erro = \frac{52 - 50}{255} = \frac{2}{255} = 0.7\% \quad (4.2)$$

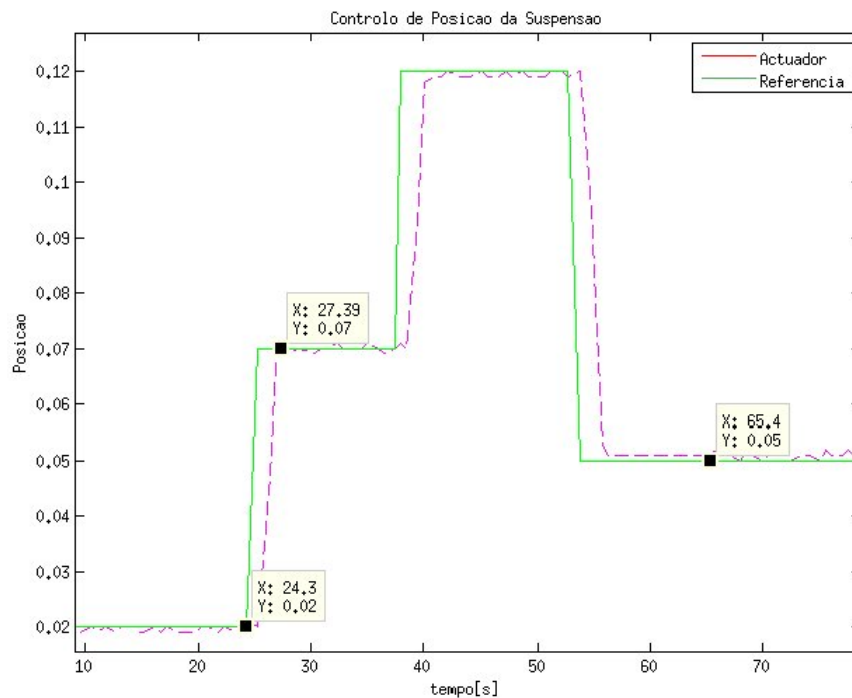


Figura 4-4: Controlo de posição do actuador da direcção.

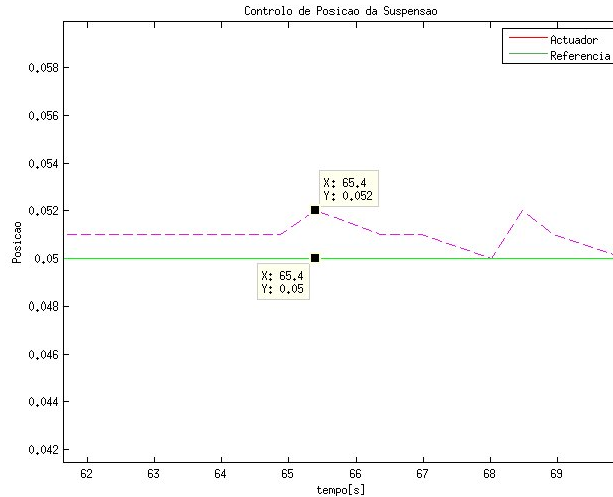


Figura 4-5: Controle de posição do actuador da direcção com ampliação.

4.2 Testes de condução

4.2.1 Slalom

O teste de Slalom foi efectuado com o intuito de avaliar a manobrabilidade da plataforma. O teste consistiu em, inicialmente, colocar três obstáculos alinhados com 4m de intervalo. Se a plataforma conseguisse contornar os obstáculos o intervalo era reduzido em 0.5m até a plataforma não conseguir contornar os obstáculos sem recorrer a reposicionamento. Verificou-se que a plataforma conseguiu contornar os obstáculos com um intervalo mínimo de 3m. Este teste foi unicamente executado com condução *Ackerman* quatro rodas.

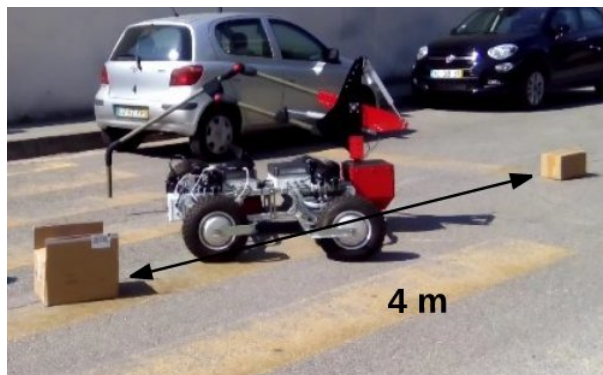


Figura 4-6: Teste de slalom.

4.2.2 Circunferência

O teste da circunferência tem como objectivo apurar qual o raio mínimo ao centro de curvatura que a plataforma consegue efectuar sem haver deslizamento lateral das rodas. Para efectuar este teste, começou-se a fazer circunferências com o ângulo máximo e foi-se diminuindo até não haver deslizamento. O raio mínimo conseguido foi 1.7 m do centro de curvatura ao centro de massa do robô, no entanto o robô não fez uma circunferência perfeita, o raio variou entre 1.7 m a 2 m.

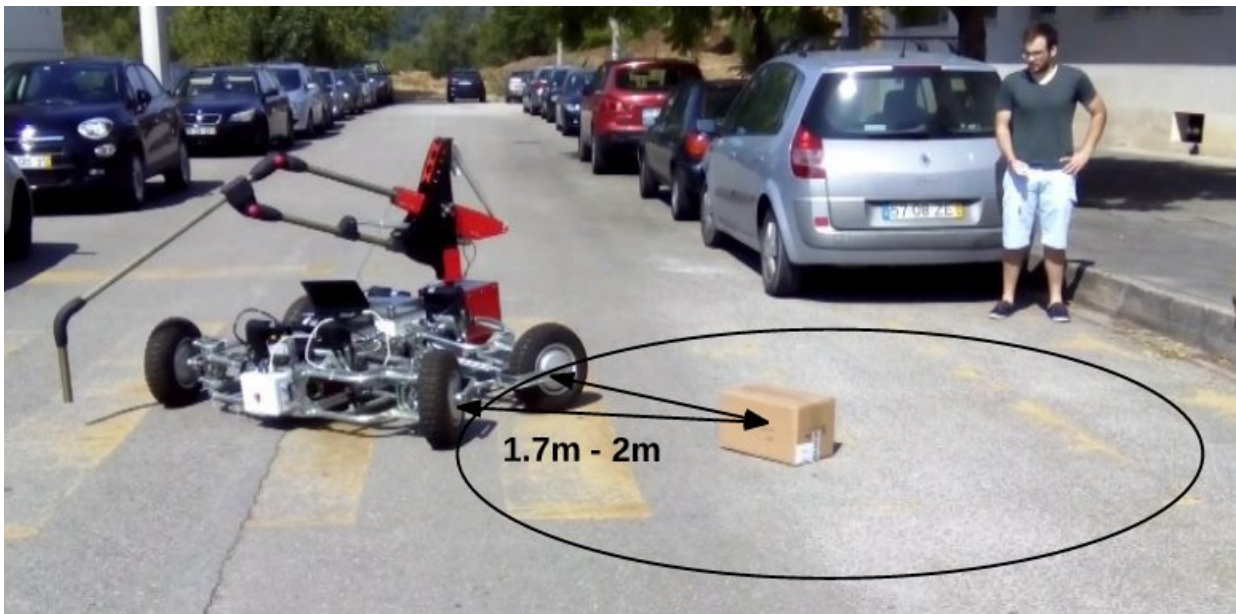


Figura 4-7: Teste da circunferência

Verificou-se que o desempenho, em termos de manobrabilidade, foi se degradando devido aos parafusos das rodas se soltarem com as vibrações e derrapagens laterais. Durante os testes verificou-se que a plataforma consegue ter um raio de rotação menor para o lado direito do que para o lado esquerdo. Esta restrição pode ter várias origens, no entanto, a mais óbvia será devido à calibração das rodas e dos próprios actuadores da direcção se degradar ao longo da utilização.

Capítulo 5

Conclusões

Em relação à plataforma tenho alguns aspectos a apontar que limitaram o desempenho e as funcionalidades que inicialmente estavam previstas implementar. Os actuadores lineares da suspensão não têm potência suficiente para ajustar a altura da estrutura, devido ao elevado peso da mesma. Deveria-se, portanto, equacionar a substituição dos mesmos ou, então, alterar a configuração para conseguir aproveitar, de forma mais eficiente, a potência que disponibilizam. As rodas usadas nesta plataforma foram projectadas para aplicações onde é necessário velocidades elevadas, o que não é o caso. Estas rodas, devido às suas características eléctricas e mecânicas, não conseguem aplicar um binário satisfatório para velocidades reduzidas, tornando-as impróprias para plataformas móveis. Apesar destas desvantagens, o desempenho da plataforma, como um todo, é satisfatório, apresentando bons resultados em termos de manobrabilidade.

Os actuadores lineares são constituídos por um motor DC convencional e um potenciómetro para efeitos de *feedback*. O acoplamento entre estes dois elementos é feito através de rodas dentadas de plástico. O actuador ao aplicar forças mais elevadas, força a roda dentada do potenciómetro, provocando a rotação do potenciómetro, enquanto que o embolo permanece na mesma posição, levando a leituras de posição erradas. Tive que calibrar várias vezes alguns actuadores lineares. É portanto, aconselhável usar actuadores lineares com mecanismos mais robustos.

A calibração da direcção (ponto central e mapeamento dos ângulos) também é um

aspecto importante para conseguir um bom desempenho na condução. O método que usei para fazer o mapeamento teve resultados positivos, no entanto, seria aconselhado usar outros métodos para ter uma métrica mais credível.

Este projecto, permitiu-me encarar problemas reais de várias áreas. O planeamento de tarefas e a utilização de utensílios para aumentar a produtividade foram alguns aspectos que tive de melhorar durante este projecto. A primeira dificuldade que tive de enfrentar foi o modo como se aborda um projecto multidisciplinar como foi este. A selecção de conectores e material adequado nem sempre foi evidente.

Na electrónica, encontrei diversas dificuldades em termos de processamento de sinais e desenvolvimento de PCBs, tendo sido, necessário fazer várias versões até chegar à versão implementada. Para além, das tarefas directamente relacionadas com o projecto, também, houveram outras tarefas que eram fruto de enganos e erros que levavam a avarias de equipamento e às respectivas substituições por componentes novos.

5.1 Trabalho Futuro

A automatização da plataforma foi um passo importante no desenvolvimento da plataforma para todo-o-terreno, no entanto, ainda existe muito que pode ser feito. Começando por melhorar aspectos como os actuadores, substituindo-os por outros mais adequados. A automatização do braço existente na estrutura era um avanço fulcral para que a plataforma possa desempenhar tarefas concretas. Tendo em conta, que a plataforma possui quatro rodas direccionáveis, outras geometrias de condução poderão ser implementadas, assim como, avaliação do consumo energético poderá ser outro objecto para futuro.

Bibliografia

- [1] Dimitrios Apostolopoulos. *Analytic Configuration of Wheeled Robotic Locomotion*. PhD thesis, Carnegie Mellon University, The Robotics Institute, 2001.
- [2] Karsten Berns and Ewald von Puttkamer. *Autonomous Land Vehicles*. Springer, 2009.
- [3] William Bolton. *Mechatronics: a multidisciplinary approach*. Pearson Education, 4th edition, 2008.
- [4] Ward Brown. *Brushless DC Motor Control Made Easy*. Microchip Technology Inc, 2002. AN857.
- [5] Gonalo Cabrita. In wheel BLDC motor. Report, LSE-ISR, 2010.
- [6] James Jay Carafano and Andrew Gudgel. The pentagon’s robots: Arming the future. Technical report, Center for Foreign Policy Studies, Kathryn and Shelby Cullom Davis Institute for International Studies, 2007.
- [7] Gert De Cubber, D Serrano, K Berns, K Chintamani, R Sabino, S Ourevitch, D Doroftei, C Armbrust, T Flamma, and Y Baudoin. Search and rescue robots developed by the european ICARUS project. In *7th Int. Workshop on Robotics for Risky Environments*, 2013.
- [8] Gregory Dudek and Michael Jenkin. *Computational principles of mobile robotics*. Cambridge university press, 2010.
- [9] Michael Fagan. ORYX 2.0: a planetary exploration mobility platform. Bachelor thesis, Worcester Polytechnic Institute, 2011.

- [10] Bastian Gaspers Frank E. Schneider. Current state of the art of unmanned systems with potential to be used for radiation measurements and sampling. Technical report, Institute for the Protection and Security of the Citizen, 2015.
- [11] Karl Iagnemma and Steven Dubowsky. *Mobile robots in rough terrain: Estimation, motion planning, and control with application to planetary rovers*, volume 12. Springer Science & Business Media, 2004.
- [12] Texas Instruments. *SN75176A Differential Bus Transceiver*. Texas Instruments, 1995.
- [13] Michel Lévis Jacob Apkarian. *Student Workbook: SRV02 Base Unit Experiment For Matlab /Simulink Users*. Quanser Inc, 2010.
- [14] Geert-Jan M Kruijff, Viatcheslav Tretyakov, Tamas Linder, Fiora Pirri, Mario Gianni, Panagiotis Papadakis, Matia Pizzoli, Aloka Sinha, Emanuele Pianese, Salvatore Corrao, et al. Rescue robots at earthquake-hit mirandola, italy: a field report. In *Safety, Security, and Rescue Robotics (SSRR), 2012 IEEE International Symposium on*. IEEE, 2012.
- [15] Thomas Kugelstadt. *The RS-485 Design Guide*. Texas instruments Appl., 2008. Literature Number: SLLA272B.
- [16] K Lohith, SR Shankapal, and MH Monish Gowda. Development of four wheel steering system for a car. *SAS Tech Journal*, 12:90–97, 2013.
- [17] Olimex Ltd. *STM32-E407 Development Board User Manual*, 2013.
- [18] Aaron Martinez and Enrique Fernández. *Learning ROS for robotics programming*. Packt Publishing Ltd, 2013.
- [19] Dave Polka. *Motor and Drivers. A practical Technology Guide*. ISA - The Instrumentation, Systems, and Automation Society, 2003.
- [20] Giulio Reina and Mario Foglia. On the mobility of all-terrain rovers. *Industrial Robot: An International Journal*, 40(2):121–131, 2013.

- [21] Roland Siegwart and Illah Reza Nourbakhsh. *Introduction to autonomous mobile robots*. Intelligent robotics and autonomous agents. MIT Press, Cambridge (Mass.), 2004.
- [22] Darko Skupnjak. Design of a mobile platform of the robot MNE-ROBECO. In *19th International Research/Expert Conference*, 2015.
- [23] Edward W. Tunstel, Richard V. Welch, and Brian H. Wilcox. Embedded control of a miniature science rover for planetary exploration. In *In 7th International Symposium on Robotics with Applications, WAC'98*, 1998.

Apêndice A

Desenvolvimento da Caixa de Controlo

A caixa de controlo tem como objectivo tornar o *hardware* mais modular e robusto a estímulos tanto mecânicos como eléctricos. Para tal foram seleccionados um conjunto de conectores e desenvolvido PCBs por forma a fazer o interface entre os microcontroladores e os actuadores.

A.1 Caixa

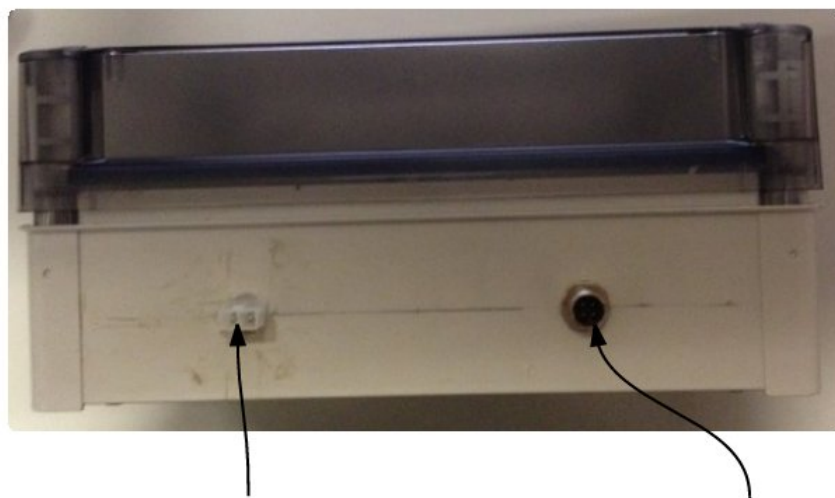
A caixa é feita de poliéster pelo fabricante *Schneider* com as seguintes dimensões: 291 mm de comprimento por 241 mm de largura e 128 mm de altura. A referência da caixa é a seguinte :NSYTBS292412HT.

A.2 Conectores

Os conectores escolhidos foram adquiridos no distribuidor *farnell*.



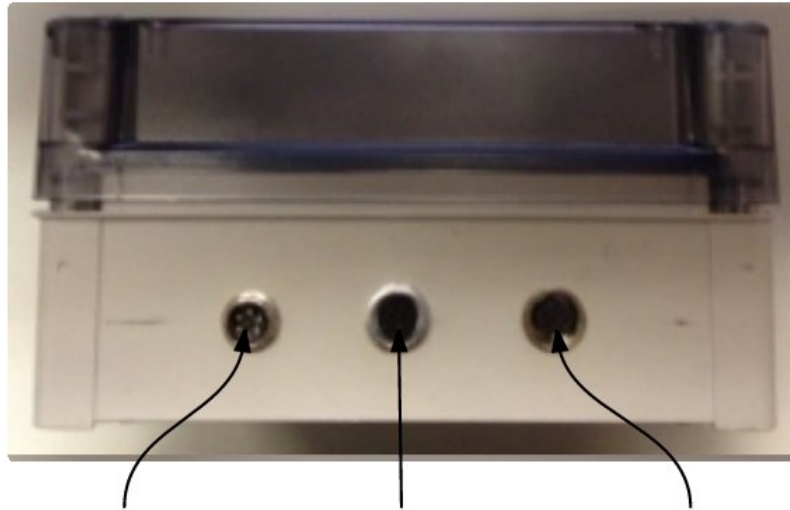
Figura A-1: Caixa de controlo.



**Conector de
Alimentação**

**Conector de
Comunicação**

Figura A-2: Caixa de controlo com a perspectiva frontal.



Conector Direcção Conector BLDC Conector Suspensão

Figura A-3: Caixa de controlo perspectiva lateral.

A.3 PCB de Interface

A PCB foi desenvolvida com o intuito de robustecer as ligações entre o microcontrolador e os actuadores. A PCB também foi concebido com o objectivo de acondicionar sinais de entrada e de saída, converter a comunicação de TTL para RS-485, através de electrónica adicional.

A.3.1 Lista de Material

Na tabela que se segue são enunciados os componentes usados na PCB de interface.

A.3.2 Circuito

Tabela A.1: Lista de material da PCB.

Tipo	Caract.	Vend.	Ref.
Conector	FCI,HEADER,1.25MM,SMT, 3WAY	Farnell	2100758
Conector	FCI,HEADER,1.25MM,SMT, 3WAY,RESPTICAL	Farnell	2100711
Conector	MOLEX,CRIMP, HOUSING, 1.25MM, 6WAY	Farnell	1012261
Conector	MOLEX,HEADER, RIGHT ANGLE, SMT, 6WAY	Farnell	1125375
Conector	FCI HOUSING, RECEPTACLE, 1.25MM, 10WAY	Farnell	2100718
Conector	FCI HEADER, 1.25MM, SMT, 10WAY	Farnell	2100735
Contacto	WURTH CONTACT, 1.25MM, CRIMP	Farnell	1841429
Resistência	TE CONNECTIVITY RESISTOR,10K,0.33W,0805	Farnell	2331808
Resistência	TE CONNECTIVITY RESISTOR,2K2,0.33W,0805	Farnell	2331800
Resistência	TE CONNECTIVITY RESISTOR,120R,0.33W,0805	Farnell	2331784
Resistência	TE CONNECTIVITY RESISTOR,3K3,0.33W,0805	Farnell	2331802
Condensador	MULTICOMP CAP, MLCC, X7R,100NF,50V,1206	Farnell	1759361
Condensador	MULTICOMP CAP, MLCC, X7R,1UF,16V,1206	Farnell	1759436
chip	Texas Instruments RS485 Transceiver	Mouser	SN75176AP

Apêndice B

Firmware

B.1 Funcionamento

Saber quem está ligado na rede

O primeiro passo é saber quem está ligado à rede, para isso é enviado o comando (Who is there?). Inicialmente era enviada uma mensagem em *Broadcast*, no entanto verificou-se que poderá existir momentos em que os dispositivos queiram escrever durante o mesmo período e uma das respostas perder-se-ia. Para evitar este problema começou-se a enviar uma mensagem por elemento na rede, sabendo à priori que só podem existir 3 elementos na rede (PC e as duas caixas de controlo). A resposta dos dispositivos vêm identificadas com o seu ID e com o ID do comando (who is there?).

Exemplo: O PC quer saber quem está na rede e começa por enviar uma mensagem ao dispositivo 2. Os campos importantes nesta função são ID do destinatário e emissor. O ID da tarefa tem que ser 0, pois envolve o dispositivo. O campo *Value* tem que ter o valor 1 para que o dispositivo saiba que tem que responder à mensagem.

Mensagem enviada pelo PC:

0	1	2	3	4-5	6	7- 14	15
&	2	1	0	4	0	1	!

A resposta do dispositivo 2 :

0	1	2	3	4-5	6	7- 14	15
&	1	2	0	4	1	0	!

A resposta informa o PC que o dispositivo 2 está disponível. O campo *Estado* vem com o valor 1.

Função Start/Stop

O comando *Start/Stop*, tanto pode ligar inicializar o *firmware*, isto é, inicializar os periféricos e as variáveis de controlo. Usando para isso o ID da tarefa igual 0. Pode também ligar e desligar cada tarefa individualmente, usando o ID da tarefa pretendida. O valor (0 ou 1) atribuído ao campo Estado é que definirá se a tarefa seleccionada é desactivada ou activada respectivamente. Em modo manual esta função liga e desliga cada tarefa individualmente, no entanto em modo teleoperado esta função liga todos as tarefa simultaneamente.

Exemplo: O PC quer ligar o microcontrolador 2 (caixa de controlo frontal)

Mensagem enviada pelo PC:

0	1	2	3	4-5	6	7- 14	15
&	2	1	0	0	1	1	!

A resposta do dispositivo 2 :

0	1	2	3	4-5	6	7- 14	15
&	1	2	0	0	1	0	!

A resposta do micrcontrolador significa que o *firmware* foi inicializado, pois o campo Estado = 1.

Start All Threads

Esta função tem com objectivo activar ou desactivar todas as tarefas simultaneamente, tendo, para tal, que atribuir o valor 1 ou 0 ao campo *Estado*, respectivamente. Se for necessário a confirmação é necessário atribuir o valor 1 ao campo *Value*.

Exemplo: Pretende-se activar todas as tarefas da caixa de controlo frontal e pretende-se, também, que que o microcontrolador retribua a confirmação que todas as tarefas estão ligadas.

Mensagem enviada pelo PC:

0	1	2	3	4-5	6	7- 14	15
&	2	1	0	6	1	1	!

A resposta do microcontrolador 2 :

0	1	2	3	4-5	6	7- 14	15
&	1	2	0	6	1	0111111	!

A mensagem devolvida pelo microcontrolador devolve o campo Value = 0111111, isto significa que as tarefas M6,M5,M4,M3,M2 e M1 estão activadas, mesma ordem.

Target Value

Esta função é responsável pelo envio dos valores de referência para os controlos de posição e de velocidade. Atribuindo um valor de 1 a 6 ao campo ID da tarefa selecciona-se a tarefa pretendida. O valor de referência é atribuído ao campo *Value* e tem 8 caracteres disponíveis. Por forma a evitar valores com virgula valor de referência é multiplicado por 1000, enviado e no *firmware* é novamente dividido por 1000. Isto é valido para o modo manual, porém em modo teleoperação, esta função envia valores de referência para quatro tarefas simultaneamente. Para isso foram constituídos grupos de tarefas com a seguinte distribuição: $(M1_f, M4_f, M1_r, M4_r)$, $(M2_f, M5_f, M2_r, M5_r)$ e $(M3_f, M6_f, M3_r, M6_r)$. Onde *f* representa a caixa de controlo frontal e *r* representa a caixa de controlo traseiro. Nesta configuração a valor de referência é convertido para uma escala entre 0 e 255 e depois é convertida para hexadecimal tendo uma escala de 00 a FF. Como cada grupo representa um tipo de actuador (rodas, suspensão ou direcção), estes são identificados atribuindo ao ID da tarefa 1 para o grupo das rodas, 2 para o grupo das suspensões e 3 para o grupo das direcções.

em numeração hexadecimal 2 caracteres por tarefa.

Exemplo: Pretende-se enviar -1 como valor de referência para o motor 1 em modo manual.

Mensagem enviada pelo PC:

0	1	2	3	4-5	6	7- 14	15
&	2	1	1	2	1	-1000	!

Neste caso não é necessário resposta.

Exemplo: Pretende-se enviar o valor de referência 255, ou seja FF para as quatro rodas em modo teleoperação.

Mensagem enviada pelo PC:

0	1	2	3	4-5	6	7- 14	15
&	2	1	1	2	1	FFFFFFFF	!

Função Control Mode

Esta função tem como objectivo configurar o *firmware* em modo manual ou teleoperado.

Exemplo: Pretende-se configurar o microcontrolador frontal em modo manual e pretende-se que o microcontrolador responda com a confirmação.

Mensagem enviada pelo PC:

0	1	2	3	4-5	6	7- 14	15
&	2	1	0	7	0	1	!

A resposta do microcontrolador 2 :

O modo manual é configurado atribuindo o campo estado igual 0, enquanto se fosse atribuído o valor 1, o microcontrolador seria configurado em teleoperação.

0	1	2	3	4-5	6	7- 14	15
&	1	2	0	7	1	0	!

Apêndice C

Software

O *software* é o nível mais alto da hierarquia da plataforma. aqui são executadas tarefas de alto nível, como condução, mapeamento de unidades físicas para as unidades dos actuadores.

Inicialmente, o software foi desenvolvido em Matlab, onde foi concebido um módulo manual e um módulo automático. À posteriori foi considerado que a implementação num *framework* amplamente usado na robótica como é ROS, seria mais vantajoso para trabalhos futuros.

C.1 Matlab

O *software* desenvolvido em Matlab consiste em três menus principais. O menu principal é apresentada na figura C-1, neste menu são detectados os dispositivos ligados ao PC. Para estabelecer *link* de comunicação com os USBs ligados ao PC é necessário carregar no botão "Abrir". Após do *link* estar estabelecido o botão muda para "Fechar".

Só após do *link* estar estabelecido os botões "Manual" e "Comando" ficam disponíveis. Ao carregar no botão "Manual" o menu manual aparece.

O menu manual tem como objectivo controlar os actuadores individualmente. Na figura C-2 pode-se observar vários botões, campos editáveis, uma área para mostrar gráficos e uma consola onde são mostrados os comandos de entrada.

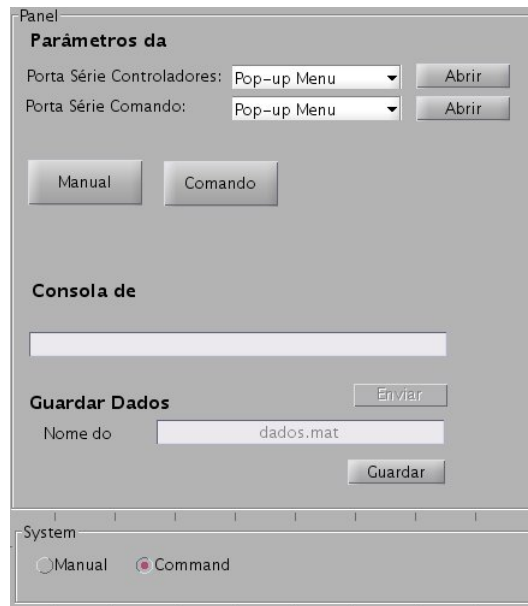


Figura C-1: Menu principal do software desenvolvido para Matlab.

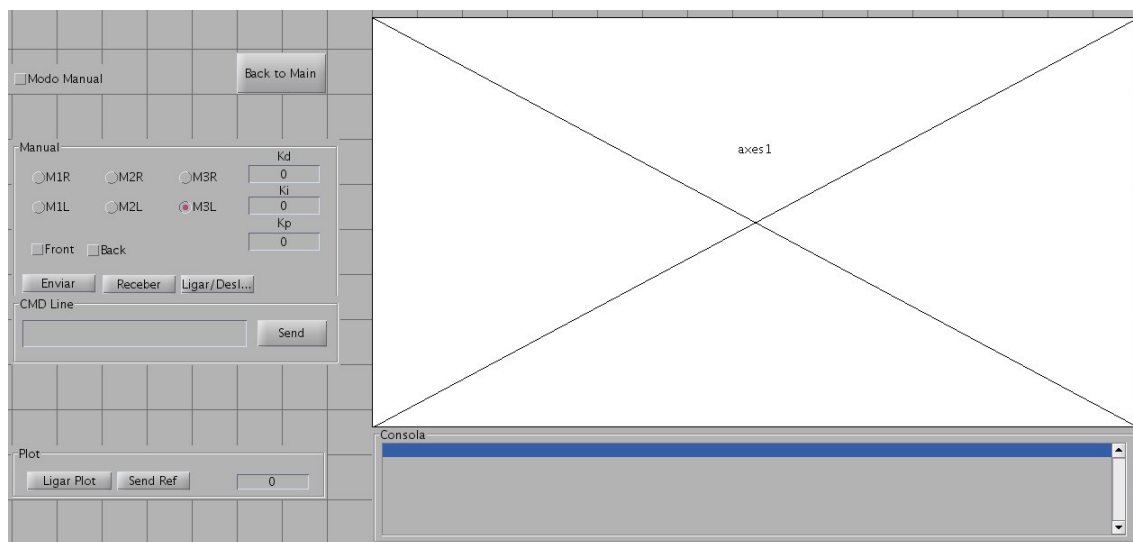


Figura C-2: Menu Manual do software Matlab.

A caixa *Modo Manual* só é assinalada quando os microcontroladores responderem ao software com a confirmação que se encontram em modo manual.

as circunferências são campos com a finalidade de seleccionar o actuador pretendido. As caixas *Front* e *Back* seleccionam a caixa frontal ou/e traseira respectivamente. Com o botão *Ligar/Desligar* ordena o software o envio de uma mensagem para de ligação ou desactivação ao actuador seleccionado. Os botões *Enviar* e *Receber* têm como função enviar e pedir ganhos do actuador seleccionado, estes são introduzido para o caso de envio e são apresentados para o caso do pedido nos três campos com os nome K_p, K_i e K_d . Os botões *Send Ref* envia a referência introduzida no campo ao lado para o microcontrolador seleccionado. O botão *Ligar Plot* liga a visualização dos dados recebidos no gráfico.

Por fim o botão *Back to Main* envia mensagens com ordem de paragem para para o micrcontrolador e regressa para o menu *Manual*.

Na figura C-3 pode-se visualizar o funcionamento o *software* em Matlab.

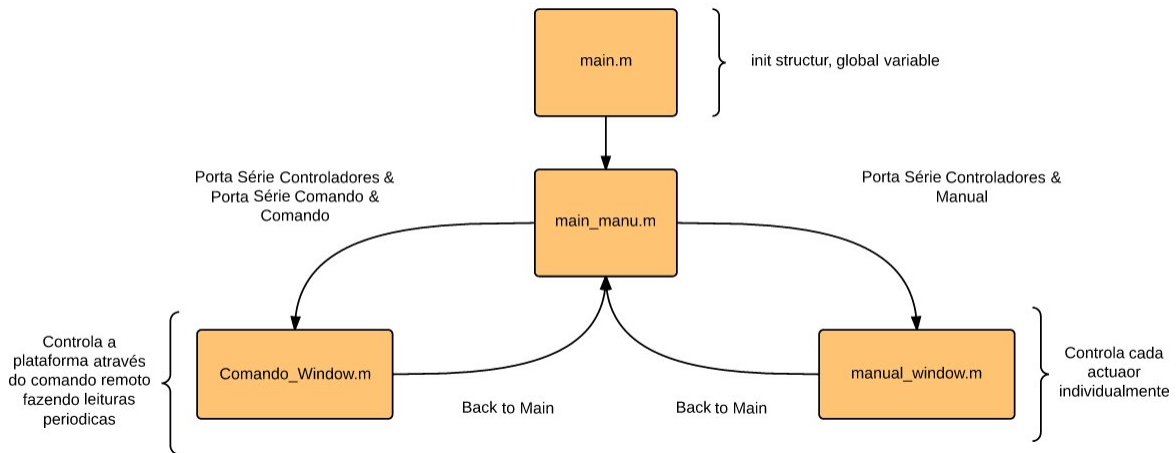


Figura C-3: Diagrama de blocos do funcionamento do software desenvolvido em Matlab.

C.2 Mapeamento

O mapeamento dos ângulos para as unidades dos actuadores é um processo que requer especial atenção. O processo consiste em criar uma referência na roda e relacio-

nar essa, mesma referência, com uma referência angular fixa, desta forma consegue-se registrar a variação da roda (actuador) com a variação angular. Após de obter vários ponto é necessário apurar a natureza da curva para obter uma função que tenha como entrada o ângulo da roda e como variável de saída a posição do actuador.

O método usado para fazer este mapeamento, consistiu em fixar uma bússola em cima da roda e apontar a variação do actuador e da bússola. Desta forma, foi conseguido obter uma função linear.

Tendo o gráfico características lineares, foi usado uma equação linear:

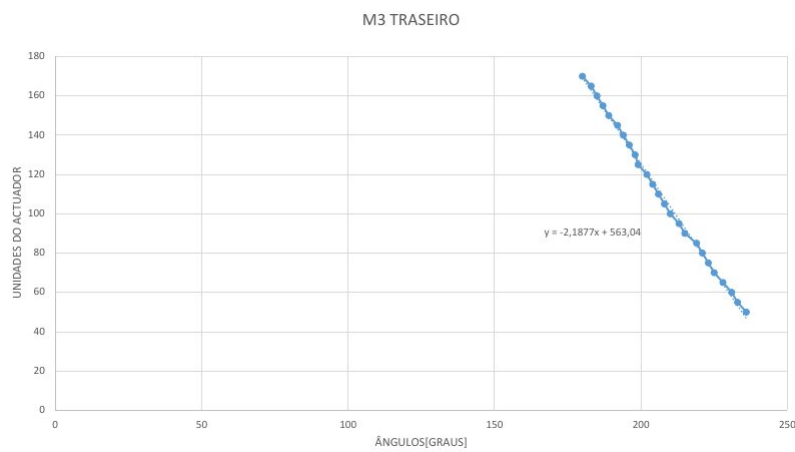


Figura C-4: Gráfico dos pontos do mapeamento do actuador M3 traseiro.

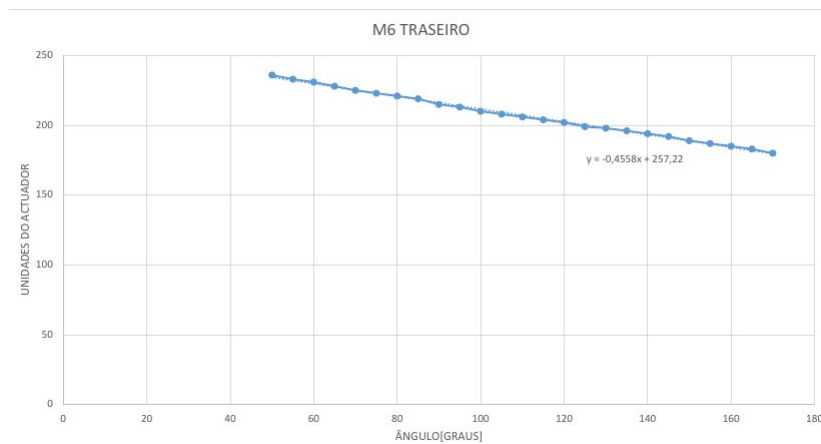


Figura C-5: Gráfico dos pontos do mapeamento do actuador M6 traseiro.

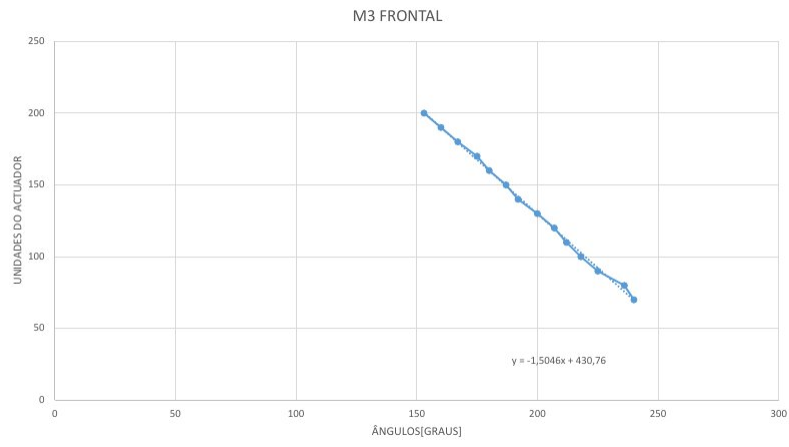


Figura C-6: Gráfico dos pontos do mapeamento do actuador M3 frontal.

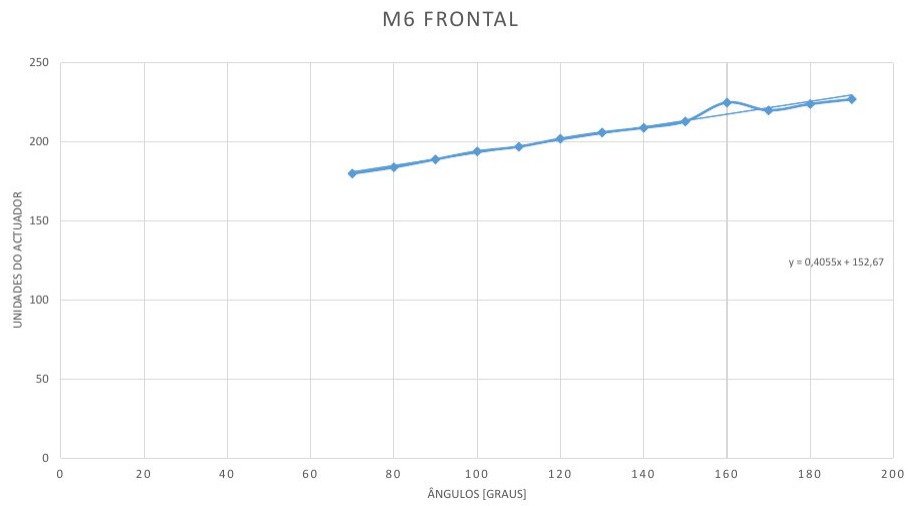


Figura C-7: Gráfico dos pontos do mapeamento do actuador M6 frontal.