



## **Desenvolvimento de uma Plataforma Robótica Omnidirecional**

**JOSÉ ANDRÉ PEREIRA DE OLIVEIRA**

novembro de 2017

# DESENVOLVIMENTO DE UMA PLATAFORMA ROBÓTICA OMNIDIRECIONAL

José André Pereira de Oliveira



Mestrado em Engenharia Eletrotécnica e de Computadores  
Área de Especialização de Automação e Sistemas  
Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto  
2017



Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha da Unidade Curricular de Tese / Dissertação (TEDI), do 2º ano, do Mestrado em Engenharia Eletrotécnica e de Computadores

Candidato: José André Pereira de Oliveira, 1110784@isep.ipp.pt

Orientação científica: Lino Manuel Batista Figueiredo, lbf@isep.ipp.pt

Empresa: Follow Inspiration, S.A.

Supervisão: António J. R. Neves, antonioneves@followinspiration.pt

José Eduardo S. C. Xavier, josexavier@followinspiration.pt



Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

20 de Novembro de 2017



*À Minha Família*



## *Agradecimentos*

Como não poderia deixar de ser em primeiro lugar à minha família. A todos de uma forma geral, mas em particular aos meus pais, irmão e avós que, ao longo de toda esta caminhada, sempre tiveram aquela palavra de ânimo e aquele conselho que, no final, fez toda diferença. Obrigado pela confiança, pela paciência e pelo apoio incondicional que sempre depositaram em mim durante toda a minha formação.

À Ângela, a minha namorada, por todo o suporte que me deu, por sempre ter acreditado em mim e por me ter sempre incentivado a querer e a ser cada vez mais e melhor, lembrando-nos do que somos e de onde viemos. Obrigado por toda a dedicação mesmo nos momentos mais difíceis.

Ao Engenheiro Lino Figueiredo, por toda a orientação, experiência, conhecimento e conselhos transmitidos, não só durante o desenvolvimento deste projeto, mas ao longo de grande parte da minha formação.

A toda a equipa da *Follow Inspiration*, que durante estes seis meses me integrou de uma forma extraordinária, com os quais aprendi muito, passei muito bons momentos e de onde levo bons amigos. Em especial aos meus orientadores, José Xavier e António Neves, por todo o apoio prestado e pela confiança depositada em mim desde o início.

Por último, mas não menos importante, aos meus Amigos. Aos que sempre souberam ser verdadeiros e apoiar, cada um à sua maneira, durante todos estes anos, com os quais vivi as melhores experiências e que certamente irão ficar para a vida.

Esta página foi intencionalmente deixada em branco.

## Resumo

Neste documento descreve-se o projeto desenvolvido na unidade curricular de Tese e Dissertação (TEDI) durante o 2<sup>o</sup> ano do Mestrado de Engenharia Eletrotécnica e de Computadores (MEEC) no ramo de Automação e Sistemas, no Departamento de Engenharia Eletrotécnica (DEE) do Instituto Superior de Engenharia do Porto (ISEP), resultado de um estágio em parceria com a empresa *Follow Inspiration*.

Nos dias de hoje, a presença de sistemas robóticos no nosso quotidiano começa a ser uma prática usual, os quais são capazes de auxiliar o ser humano nas mais diversificadas tarefas. Neste seguimento, a *Follow Inspiration* desenvolve, entre outros, soluções robóticas móveis autónomas (wiiGO) que interagem com o utilizador, prestando-lhe auxílio nas suas tarefas.

Desta forma, o principal objetivo desta dissertação está relacionado com o desenvolvimento de uma plataforma omnidirecional, com possível aplicação ao wiiGO e/ou a outros sistemas. Do desenrolar do projeto foram desenvolvidas duas plataformas omnidirecionais, com configurações diferentes (quatro rodas *mecanum* e quatro rodas universais), nas quais foi elaborado o devido controlo dos motores assim como foi desenvolvido *software* que permite a estimação do seu deslocamento e posição ao longo do tempo, entre outros.

Nesta dissertação estão documentadas todas escolhas ao nível de *hardware*, assim como todo o *software* desenvolvido que, em conjunto, conferem mobilidade omnidirecional às plataformas.

## Palavras-chave

Robótica Móvel, Mobilidade Omnidirecional, Controlo de Motores, ROS, Odometria.

Esta página foi intencionalmente deixada em branco.

## Abstract

This dissertation reports the project developed in the Thesis/Dissertation course during the 2nd year of the Master of Electrical and Computer Engineering in the field of Automation and Systems, Department of Electrical Engineering of the Instituto Superior de Engenharia do Porto, result of a internship in partnership with the company *Follow Inspiration*.

Nowadays, the presence of robotic systems in our daily life begins to be an usual practice, which are able to assist the human being in the most diversified tasks. In this sequence, *Follow Inspiration* develops, among others, autonomous mobile robotic solutions (wiiGO), that interact with the user, assisting him in his tasks.

So, the main goal of this dissertation is related with the development of an omnidirectional platform, for possible application in wiiGO and/or other systems. During the project two omnidirectional platforms with different configurations were developed (four *mecanum* wheels and four universal wheels) in which were made the proper control of motors as well as was developed software that allow estimate the displacement and position over time, among others.

In this dissertation all hardware-level choices are documented and the developed software that together confer omnidirectional mobility to the platforms.

## Keywords

Mobile Robotics, Omnidirectional Mobility, Motors Control, ROS, Odometry.

Esta página foi intencionalmente deixada em branco.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Enquadramento e Motivação . . . . .	3
1.2	Objetivos . . . . .	5
1.3	Planificação e Calendarização . . . . .	6
1.4	Estrutura do Relatório . . . . .	8
<b>2</b>	<b>Estado da Arte</b>	<b>9</b>
2.1	Mobilidade Omnidirecional . . . . .	9
2.1.1	Tipos de Rodas . . . . .	10
2.2	Configurações de Plataformas Omnidirecionais . . . . .	15
2.2.1	Plataforma com Três Rodas Universais . . . . .	15
2.2.2	Plataforma com Quatro Rodas Universais . . . . .	16
2.2.3	Plataforma com Quatro Rodas <i>mecanum</i> . . . . .	18
2.3	Modelo Cinemático . . . . .	20
2.3.1	Cinemática do Robô com Três Rodas . . . . .	21
2.3.2	Cinemática do Robô com Quatro Rodas . . . . .	23
2.3.3	Cinemática do Robô com Quatro Rodas <i>Mecanum</i> . . . . .	25
2.4	Aplicações dos Robôs Omnidirecionais . . . . .	27
2.4.1	Indústria . . . . .	28
2.4.2	Domicílio e Cuidados de Saúde . . . . .	30
2.5	<i>Robot Operating System</i> (ROS) . . . . .	32
2.5.1	Modelo de Comunicação <i>Publisher/Subscriber</i> . . . . .	33
2.5.2	<i>Filesystem Level</i> . . . . .	35

2.5.3	<i>Computation Graph Level</i> . . . . .	35
2.5.4	<i>Community Level</i> . . . . .	37
2.6	Modelo de Odometria . . . . .	37
2.6.1	Erros na Odometria . . . . .	38
<b>3</b>	<b>Projeto e Implementação de <i>Hardware</i></b>	<b>41</b>
3.1	Arquitetura Geral do Sistema . . . . .	42
3.2	Motores . . . . .	43
3.2.1	Motores CC <i>Brushed</i> . . . . .	44
3.2.2	Motores CC <i>Brushless</i> . . . . .	46
3.3	Caixas Redutoras ( <i>Gearboxes</i> ) . . . . .	47
3.3.1	Escolha dos Motores . . . . .	50
3.4	<i>Encoders</i> . . . . .	53
3.4.1	<i>Encoders</i> Absolutos . . . . .	54
3.4.2	<i>Encoders</i> Incrementais . . . . .	54
3.4.3	<i>Encoders</i> Magnéticos . . . . .	55
3.4.4	<i>Encoders</i> Óticos . . . . .	56
3.4.5	<i>Encoders</i> Utilizados . . . . .	58
3.5	Sistema de Controlo de Motores . . . . .	61
3.5.1	<i>Pulse Width Modulation</i> (PWM) . . . . .	62
3.5.2	Circuito de Ponte H . . . . .	64
3.5.3	<i>RoboteQ</i> SDC2130 . . . . .	67
3.5.4	<i>Sabertooth</i> 2x25 . . . . .	70
3.5.5	<i>Kangaroo x2 Motion Controller</i> . . . . .	73
3.5.6	Comparativo dos Controladores . . . . .	75
3.6	Escolha das Rodas . . . . .	77
3.7	Desenvolvimento da Estrutura . . . . .	79
<b>4</b>	<b>Controlo e Desenvolvimento de <i>Software</i></b>	<b>89</b>
4.1	Controlo . . . . .	92

4.1.1	Controlo PID . . . . .	92
4.1.2	Configuração do Controlador . . . . .	95
4.1.3	Descrição do <i>Loop</i> de Controlo . . . . .	98
4.1.4	Sintonia/Retificação do PID . . . . .	99
4.1.5	<i>Script</i> de Envio de <i>Feedback</i> . . . . .	101
4.2	Modelo de Odometria - Quatro Rodas <i>Mecanum</i> . . . . .	102
4.3	Modelo de Odometria - Quatro Rodas Universais . . . . .	105
4.3.1	Odometria em Sistemas Diferenciais . . . . .	106
4.4	Desenvolvimento do <i>Software</i> em ROS . . . . .	110
4.4.1	Nó <code>wiigo_omni_mecanum</code> . . . . .	110
4.4.2	Nó <code>wiigo_omni_universal</code> . . . . .	113
4.4.3	Nó <code>wiigo_omni_joy</code> . . . . .	115
4.4.4	Nós <code>roboteq_driver</code> . . . . .	116
4.4.5	<i>Launch Files</i> . . . . .	117
<b>5</b>	<b>Testes e Resultados</b>	<b>119</b>
5.1	Testes com a Base de Rodas <i>Mecanum</i> . . . . .	120
5.1.1	Testes de Deslocamento de 1 <i>m</i> . . . . .	122
5.1.2	Testes de Deslocamento de 4 <i>m</i> . . . . .	124
5.1.3	Teste de Deslocamento de $\pi$ <i>rad</i> . . . . .	127
5.2	Testes com a Base de Rodas Universais . . . . .	131
5.2.1	Teste de Deslocamento de 1 <i>m</i> . . . . .	133
5.2.2	Teste de Deslocamento de 4 <i>m</i> . . . . .	135
5.2.3	Teste de Deslocamento de $\pi$ <i>rad</i> . . . . .	138
<b>6</b>	<b>Conclusões e Trabalho Futuro</b>	<b>143</b>
	<b>Referências Documentais</b>	<b>152</b>
	<b>Anexo A. <i>Datasheets</i> do <i>Hardware</i> Utilizado</b>	<b>153</b>

Anexo B. Resultados do Estudo das Rodas Omnidirecionais Universais	171
Anexo C. Desenhos Técnicos	175
Anexo D. Diagramas do <i>Software</i> desenvolvido em ROS	187
Anexo E. Resultados dos Testes de Odometria	191

# Lista de Figuras

Figura 1.1 Carrinho de Compras Autônomo: wiiGO [5]. . . . .	4
Figura 2.1 Rodas de desenho convencional: (a) Rodízio, (b) Roda direcional [6]. . . . .	10
Figura 2.2 Rodas de desenho especial: (a) Universal, (b) <i>Mecanum</i> [6]. . . . .	11
Figura 2.3 Rodas Universais: (a) Simples, (b) Dupla, (c) Alternada [12]. . . . .	12
Figura 2.4 Esquema do movimento da roda <i>mecanum</i> [12]. . . . .	13
Figura 2.5 Configurações Omnidirecionais: (a) três rodas universais, (b) quatro rodas universais, (c) quatro rodas <i>mecanum</i> . . . . .	15
Figura 2.6 Movimentos do Robô Omnidirecional com Três Rodas [14]. . . . .	16
Figura 2.7 Movimentos do Robô Omnidirecional com Quatro Rodas [14]. . . . .	17
Figura 2.8 Rodas <i>mecanum</i> : “ <i>left-handed</i> ” e “ <i>right-handed</i> ” [16]. . . . .	18
Figura 2.9 Movimentos do Robô Omnidirecional com Quatro Rodas <i>mecanum</i> [17]. . . . .	19
Figura 2.10 Configuração do Robô com Três Rodas [15]. . . . .	21
Figura 2.11 Configuração do Robô com Quatro Rodas [15]. . . . .	23
Figura 2.12 Configuração do Robô com Quatro Rodas <i>Mecanum</i> . . . . .	26
Figura 2.13 Plataforma Omnidirecional KUKA omniMove [24]. . . . .	29
Figura 2.14 Robô Toyota HSR [30]. . . . .	31

Figura 2.15 Comunicação entre dois ROS <i>Nodes</i> utilizando o método <i>Publisher/Subscriber</i> [32]. . . . .	34
Figura 3.1 Arquitetura Geral do Sistema. . . . .	43
Figura 3.2 Motor CC <i>Brushed</i> [36]. . . . .	45
Figura 3.3 Motor CC <i>Brushless</i> [38]. . . . .	47
Figura 3.4 Engrenagem Direta [42]. . . . .	49
Figura 3.5 Engrenagem Planetária [41]. . . . .	49
Figura 3.6 <i>Gearmotor</i> IG52-04. . . . .	51
Figura 3.7 <i>Gearmotor</i> PK42BB-EC035.120 . . . . .	52
Figura 3.8 Estrutura de um <i>Encoder</i> Absoluto [44]. . . . .	54
Figura 3.9 Estrutura de um <i>Encoder</i> Incremental [44]. . . . .	55
Figura 3.10 Funcionamento de um <i>Encoder</i> Magnético [45]. . . . .	56
Figura 3.11 <i>Encoder</i> Ótico de Quadratura [10]. . . . .	57
Figura 3.12 <i>Encoder</i> AEDM-5810 [47]. . . . .	59
Figura 3.13 <i>Pinout</i> do <i>Encoder</i> AEDM-5810 [47]. . . . .	59
Figura 3.14 <i>Output</i> do <i>Encoder</i> AEDM-5810 [47]. . . . .	60
Figura 3.15 Sinais PWM. . . . .	63
Figura 3.16 Esquema do Circuito de Ponte-H [49]. . . . .	65
Figura 3.17 Esquema de Condução da Ponte-H [49]. . . . .	66
Figura 3.18 <i>RoboteQ</i> SDC2130 [50]. . . . .	68
Figura 3.19 Interface do <i>Software</i> <i>Roborun+</i> . . . . .	69
Figura 3.20 <i>Sabertooth</i> 2x25 V2. . . . .	70
Figura 3.21 Ligações do Controlador <i>Sabertooth</i> 2x25 [53]. . . . .	71
Figura 3.22 <i>Kangaroo x2 Motion Controller</i> . . . . .	73
Figura 3.23 Esquema do módulo <i>Kangaroo x2</i> [33]. . . . .	74
Figura 3.24 Modos de Operação do <i>Kangaroo x2</i> consoante interruptores DIP [54]. . . . .	74
Figura 3.25 Rodas Omnidirecionais <i>Mecanum</i> Utilizadas. . . . .	77
Figura 3.26 Rodas Omnidirecionais Universais Utilizadas. . . . .	78

Figura 3.27	Configurações das Bases Desenvolvidas: a) Retangular com 4 Rodas <i>Mecanum</i> ; b) Circular com 4 Rodas Universais.	79
Figura 3.28	Estrutura da Plataforma com Rodas <i>Mecanum</i> .	80
Figura 3.29	Projeto da Base Omnidirecional (Diferentes Vistas).	81
Figura 3.30	Projeto da Base Omnidirecional (Plataforma Inferior).	82
Figura 3.31	Interface do Autodesk Fusion 360.	82
Figura 3.32	Corte CNC da Base Omnidirecional.	83
Figura 3.33	Projeto dos Suportes de Motores (à esquerda) e Respetivas Peças (à direita)	84
Figura 3.34	Impressão 3D dos Suportes dos Motores.	84
Figura 3.35	Aplicação dos Suportes de Motores na Base.	85
Figura 3.36	Projeto do Sistema de Acoplamento Roda-Eixo.	85
Figura 3.37	Componentes do Sistema de Acoplamento Roda-Eixo.	86
Figura 3.38	Plataforma Omnidirecional com Rodas Universais Desenvolvida.	87
Figura 4.1	Diagrama de Funcionamento do Sistema.	90
Figura 4.2	Esquema de <i>Software</i> da Plataforma Omnidirecional.	91
Figura 4.3	Esquema de Controlo PID [56].	93
Figura 4.4	Resposta de um Sistema em Malha-Fechada [57].	94
Figura 4.5	Configuração dos Parâmetros <i>Max Speed</i> e <i>Pulses/Rev</i> no <i>Roborun+</i> .	96
Figura 4.6	Esquema do Sistema de Acoplamento Roda-Motor - Base com Rodas <i>Mecanum</i> .	97
Figura 4.7	Esquema do Sistema de Acoplamento Roda-Motor - Base com Rodas Universais.	97
Figura 4.8	Esquema do Algoritmo de Controlo PID do RoboteQ [58].	99
Figura 4.9	Interface para Sintonia do PID - <i>Roborun+</i> .	100
Figura 4.10	Fluxograma do <i>Script</i> para Envio de <i>Feedback</i> .	101

Figura 4.11 Deslocamentos da Plataforma <i>Mecanum</i> com Quatro Rodas. . . . .	102
Figura 4.12 Deslocamentos da Plataforma com Quatro Rodas Universais. . . . .	105
Figura 4.13 Deslocamentos de um Robô Diferencial [32] . . . . .	106
Figura 4.14 Fluxograma do Nó <i>wiigo_omni_mecanum</i> . . . . .	111
Figura 4.15 Numeração das Rodas da Plataforma com Rodas <i>Mecanum</i> . . . . .	112
Figura 4.16 Referencial Cartesiano da Posição da Plataforma. . . . .	112
Figura 4.17 Numeração das Rodas da Plataforma com Rodas Universais. . . . .	114
Figura 4.18 Fluxograma do Nó <i>wiigo_omni_joy</i> . . . . .	115
Figura 5.1 Sentidos positivos do Referencial da Plataforma <i>Mecanum</i> . . . . .	121
Figura 5.2 Rodas <i>Mecanum</i> com Rolos de Borracha Aplicadas. . . . .	130
Figura 5.3 Sentidos positivos do Referencial da Plataforma com Rodas Universais. . . . .	131

# Lista de Tabelas

Tabela 1.1	Calendarização do Estágio. . . . .	7
Tabela 2.1	Tabela comparativa dos diferentes tipos de rodas [12].	14
Tabela 3.1	Tabela comparativa dos controladores <i>RoboteQ</i> SDC2130 e <i>Sabertooth 2x25 V2</i> . . . . .	75
Tabela 4.1	Relação entre os Ganhos do PID e as Especificações do Sistema. . . . .	95
Tabela 4.2	Ganho dos Controlo PID. . . . .	100
Tabela 4.3	Movimentos das Plataformas Segundo os Comandos do <i>Joystick</i> . . . . .	116
Tabela 5.1	Movimentos e Sentidos do Robô Consoante Movimento das Rodas. . . . .	121
Tabela 5.2	Resultados do Teste de Deslocamento de 1 <i>m</i> - Piso Alcatifado. . . . .	122
Tabela 5.3	Resultados do Teste de Deslocamento de 1 <i>m</i> - Piso Liso. . . . .	123
Tabela 5.4	Resultados do Teste de Deslocamento de 4 <i>m</i> - Piso Alcatifado. . . . .	125
Tabela 5.5	Resultados do Teste de Deslocamento de 4 <i>m</i> - Piso Liso. . . . .	126
Tabela 5.6	Resultados do Teste de Deslocamento de $\pi$ <i>rad</i> - Piso Alcatifado. . . . .	127

Tabela 5.7	Resultados do Teste de Deslocamento de $\pi \text{ rad}$ - Piso	
	Liso. . . . .	128
Tabela 5.8	Movimentos e Sentidos do Robô Consoante Movimento	
	das Rodas. . . . .	132
Tabela 5.9	Resultados do Teste de Deslocamento de $1 \text{ m}$ - Piso	
	Alcatifado. . . . .	133
Tabela 5.10	Resultados do Teste de Deslocamento de $1 \text{ m}$ - Piso	
	Liso. . . . .	134
Tabela 5.11	Resultados do Teste de Deslocamento de $4 \text{ m}$ - Piso	
	Alcatifado. . . . .	135
Tabela 5.12	Resultados do Teste de Deslocamento de $4 \text{ m}$ - Piso	
	Liso. . . . .	137
Tabela 5.13	Resultados do Teste de Deslocamento de $\pi \text{ rad}$ - Piso	
	Alcatifado. . . . .	138
Tabela 5.14	Resultados do Teste de Deslocamento de $\pi \text{ rad}$ - Piso	
	Liso. . . . .	139

# Lista de Acrónimos

**AGV** *Automated Guided Vehicles*

**AMR** *Automated Mobile Robots*

**CA** Corrente Alternada

**CAD** *Computer-Aided Design*

**CAM** *Computer-Aided-Manufacturing*

**CAN** *Controller Area Network*

**CC** Corrente Contínua

**CNC** *Computer Numerical Control*

**CPR** *Cycles per Revolution*

**CPU** *Central Processing Unit*

**DEE** Departamento de Engenharia Eletrotécnica

**DOF** *Degrees of Freedom*

**HSR** *Human Support Robot*

**ISEP** Instituto Superior de Engenharia do Porto

**LED** *Light-Emitting Diode*

**MDF** *Medium-Density Fiberboard*

**MEEC** Mestrado em Engenharia Eletrotécnica e de Computadores

**MOSFET** *Metal Oxide Semiconductor Field Effect Transistor*

**PID** Proporcional, Integral e Derivativo

**PWM** *Pulse Width Modulation*

**ROS** *Robot Operating System*

**RPM** Rotações por Minuto

**SAIL** *Stanford Artificial Intelligence Laboratory*

**SLAM** *Simultaneous Localization and Mapping*

**SSH** *Secure Shell*

**TEDI** Tese/Dissertação

**XML** *eXtensible Markup Language*

# Capítulo 1

## Introdução

Já desde tempos remotos que o Homem sonha com a possibilidade de existência de máquinas, mais ou menos móveis, capazes de o auxiliar nas mais variadas tarefas, ou até mesmo com a possibilidade de as realizar com maior velocidade, precisão e segurança [1].

São vários os exemplos, desde o início dos tempos, nos quais o Homem ambiciona criar artefactos à sua imagem. O relógio de água - Clepsidra, introduzido pelos Babilônios (1400 aC), foi um dos primeiros sistemas mecânicos automatizados existentes, com o objetivo de medir o tempo. Nos séculos seguintes, a criatividade do ser humano deu origem a uma série de dispositivos, desde o autómato do teatro de Heron de Alexandria (100 dC) até aos numerosos desenhos engenhosos de Leonardo da Vinci (1500 dC). Estas e muitas outras criações, marcaram o desenvolvimento dos autómatos e foram evidenciando o esclarecimento do conceito de robô. Contudo, a realização física deste conceito, só surgiu durante o decorrer do século XX, à medida que iam surgindo também avanços tecnológicos.

Foi em 1920 que, pela primeira vez, foi introduzido o termo robô, por Karel Capek, dramaturgo checo, e cujo significado etimológico provém da palavra

*robota* - trabalho forçado ou servo [2]. Alguns anos mais tarde, em 1940, Isaac Asimov concebeu a ética da interação entre os robôs e o Homem, através das três famosas leis fundamentais da robótica.

No princípio dos anos 70, foram introduzidos na indústria os robôs manipuladores, como um acréscimo ao processo produtivo, pela sua precisão, rapidez e facilidade de programação. Esta tecnologia proliferou-se rapidamente representando uma nova era na automação industrial, pelo que a flexibilidade na produção levou à criação do conceito de célula de fabrico robotizada.

Em 1980 surgiu a “ciência que estuda a conexão inteligente entre a perceção e a ação”, como uma das possíveis definições para a robótica. De acordo com esta definição, a ação de um sistema robótico pressupõe a existência de um sistema de locomoção (rodas, lagartas, pernas, hélices e outros) e/ou um sistema de manipulação, com a finalidade de operar determinado(s) objeto(s) presente(s) no seu ambiente, através de atuadores que animam os seus componentes mecânicos. A perceção é extraída por sensores que fornecem informações acerca do estado do robô e do ambiente onde está inserido. A conexão inteligente está assente na arquitetura de programação, planeamento e controlo, que se baseia na perceção e adjudica a ação [3].

Já na década de 90, aparecem os primeiros robôs móveis como dispositivos de transporte automático. São constituídos por plataformas mecânicas providas de um sistema de locomoção capaz de navegar num determinado ambiente, de forma mais ou menos autónoma. A aplicação deste tipo de robôs é vasta, desde robôs domésticos e de serviço, até aplicações industriais, ou mesmo robôs militares ou de vigilância. Estes, são apenas alguns exemplos de áreas da sociedade onde se podem encontrar robôs móveis [4].

Nos dias de hoje, a presença de sistemas robóticos no quotidiano do ser humano começa a demonstrar-se uma prática recorrente e, acima de tudo,

bastante útil e vantajosa. Tanto a nível lúdico quer profissional existe, na atualidade, um sem-fim de sistemas robóticos, tanto móveis como manipuladores, e até um misto dos dois, adequados para as mais variadas tarefas.

No seguimento desta realidade, o projeto, descrito na presente dissertação, pressupõe o desenvolvimento de uma plataforma móvel, dotada de locomoção omnidirecional, capaz de se movimentar sem restrições no espaço bidimensional, como possível solução para integração no wiiGO, robô móvel e autónomo desenvolvido pela Follow Inspiration.

## 1.1 Enquadramento e Motivação

O trabalho realizado, descrito ao longo do presente relatório, insere-se no âmbito da unidade curricular de Tese/Dissertação do 2º ano do Mestrado em Engenharia Eletrotécnica e de Computadores, ramo de Automação e Sistemas, do Instituto Superior de Engenharia do Porto. Este projeto surge do culminar da atividade num estágio curricular, realizado pelo autor, com a duração de seis meses, na empresa *Follow Inspiration* em parceria com o ISEP.

A *Follow Inspiration* é uma empresa tecnológica, fundada em 2012, focada no desenvolvimento de soluções inovadoras de *software* e *hardware*, baseadas essencialmente na experiência do utilizador. Tem como principais áreas de conhecimento a robótica e o reconhecimento de imagem aliados ao mapeamento e navegação. Na atualidade, o seu principal foco de atividade concentra-se no desenvolvimento de um robô designado wiiGO (representado na figura 1.1), cujo princípio assenta num carrinho de compras autónomo, capaz de seguir pessoas, com ou sem mobilidade reduzida, auxiliando-as no transporte das suas compras em superfícies comerciais. Este robô, é sustentado por um sistema de locomoção diferencial, combinado com um conjunto de sensores

ultrassónicos, laser e câmaras que, em conjunto, formam um mecanismo de perceção e possibilitam a total autonomia no seu deslocamento.



Figura 1.1: Carrinho de Compras Autónomo: wiiGO [5].

Mesmo sendo o wiiGO o seu principal foco, a *Follow Inspiration* está também empenhada no desenvolvimento de soluções com outros propósitos, baseadas no conceito do wiiGO, mas dedicadas a outros ramos do mercado, que não o retalho. Um exemplo disso é o projeto em desenvolvimento, denominando wiiGO Industry, que consiste na criação de um robô autónomo, aplicado ao setor industrial, que possui as mesmas características do wiiGO original, mas cuja locomoção é sustentada por um sistema omnidirecional. De encontro às necessidades impostas por este projeto, surge o propósito da presente dissertação, no qual se pretende o desenvolvimento de uma plataforma omnidirecional, aplicada a este sistema pensado para a indústria, mas com possível adaptação ao wiiGO.

Por parte do autor, em termos motivacionais, a escolha deste projeto foi sustentada em primeiro lugar, pelo desejo de realizar um estágio curricular na área e também pelo fascínio e grande interesse por projetos de inovação.

A possibilidade de aquisição de conhecimentos, numa ótica de desenvolvimento de produto, relacionada com um caso real de uma empresa, na qual a inovação é um fator preponderante, tornou-se um dos principais objetos motivacionais e inspiradores, na opção pela realização deste trabalho.

## 1.2 Objetivos

O objetivo principal do projeto é o desenvolvimento de uma plataforma móvel omnidirecional. Em conjunto com o objetivo principal é requerido o desenvolvimento de um sistema que estime o deslocamento e posição angular da plataforma ao longo do seu movimento. Para o alcance destes objetivos tornou-se necessária a divisão do trabalho em vários tópicos que resultaram nas seguintes contribuições:

- Estudo acerca da mobilidade robótica omnidirecional e dos diferentes tipos de plataformas;
- Identificação do *hardware* necessário para o desenvolvimento da base omnidirecional;
- Estudo, desenho e concretização da estrutura da plataforma e seus componentes;
- Montagem e controlo da base omnidirecional;
- Estudo, implementação e validação do modelo de odometria aplicado a plataformas omnidirecionais.

### 1.3 Planificação e Calendarização

Para estruturação e planificação do projeto, tanto pelo autor como pela empresa, durante o período de estágio, foram efetuadas reuniões semanais com toda a equipa de desenvolvimento relacionada com o projeto em questão. Nestas reuniões, era discutido o trabalho efetuado no período anterior à mesma e eram planeadas e discutidas as tarefas para a semana subsequente. Todas as decisões relacionadas com o projeto, foram tomadas em conjunto com os restantes membros da equipa de desenvolvimento, sendo que algumas decisões mais específicas, relativas ao desenvolvimento, foram tomadas pelo autor e supervisores.

Em termos de organização de trabalho foi utilizada a plataforma *SCRAIM* (já utilizada pela empresa) para gestão de tarefas e *deadlines*. Para além desta plataforma, ao longo das várias semanas de trabalho foram efetuados *logbooks* com um resumo descritivo das tarefas efetuadas.

A calendarização do projeto desenvolvido durante o período do estágio foi efetuada pelo autor de acordo com a Tabela 1.1. Nesta tabela pode-se observar o tempo dedicado a cada tarefa. Sendo que o desenvolvimento foi efetuado em ambiente de empresa, ocorreram algumas mudanças relativas ao planeamento, no entanto todas as etapas foram cumpridas.

Tabela 1.1: Calendarização do Estágio.

Tarefa	Fevereiro	Março	Abril	Maio	Junho	Julho	Agosto
	-   -   1   2	3   4   5   6	7   8   9   10	11   12   13   14	15   16   17   18	19   20   21   22	23   24   -   -
Estudo da Mobilidade Omnidirecional							
Estudo da Cinemática e Odometria							
Estudo do ROS							
Montagem e Configuração da base							
Desenvolvimento de Software							
Estudo dos Controladores							
Desenvolvimento CAD							
Testes com a Base							
Elaboração do Relatório							

## 1.4 Estrutura do Relatório

A presente dissertação está organizada em seis capítulos.

No primeiro capítulo é introduzido o tema da dissertação, é efetuado o seu enquadramento, são apresentados os objetivos principais e é mencionada a planificação e calendarização.

O capítulo 2 é referente ao Estado da Arte, no qual são introduzidos conceitos relativos à mobilidade omnidirecional quanto ao tipo de rodas, configurações e modelos cinemáticos. Também são demonstradas aplicações de plataformas omnidirecionais em diferentes áreas assim como é mencionado o *Robot Operating System* e a odometria.

No capítulo 3 é mencionado o projeto e implementação de *Hardware*, onde se refere a arquitetura do sistema assim como se explicam todas as decisões relativas ao *hardware* escolhido e desenvolvido.

No quarto capítulo são apresentados os princípios relativos ao controlo dos motores, são explicados os modelos de odometria aplicados e é demonstrado todo o desenvolvimento de *software*.

O capítulo 5 é referente à análise dos resultados obtidos assim como aos testes de validação efetuados, para as duas plataformas.

No sexto capítulo são apresentadas as conclusões finais da dissertação bem como são sugeridas algumas propostas de melhorias futuras.

Por fim, a última parte desta dissertação é reservado para o anexos onde constam os documentos consultados para elaboração do projeto, bem como outros documentos e esquemas elaborados que, não estão representados ao longo do documento, mas que são parte integrante do que é descrito.

## Capítulo 2

# Estado da Arte

*Neste capítulo serão introduzidos alguns conceitos relativos à mobilidade omnidirecional, assim como serão explicados alguns dos diferentes tipos de configurações de plataformas com este tipo mobilidade. Também será efetuada uma análise da aplicação deste tipo de plataformas em diferentes setores da sociedade e por fim serão abordados alguns temas e conceitos fundamentais, aplicados nesta área, de modo a tornar mais compreensível a sua abordagem no decorrer do projeto.*

### 2.1 Mobilidade Omnidirecional

Diz-se omnidirecional, toda a plataforma capaz de se mover em qualquer direção do plano bidimensional, a partir de qualquer configuração, de forma instantânea, isto é, uma plataforma capaz de controlar de forma independente todos os graus de liberdade no espaço bidimensional [6].

Um robô omnidirecional é capaz de realizar facilmente tarefas em ambientes congestionados com obstáculos estáticos e dinâmicos [7]. Este tipo de plataformas, permite o deslocamento entre dois pontos sem necessidade de

girar sobre si primeiro, assim como é capaz de efetuar um movimento de translação combinado com um movimento de rotação. Neste seguimento a utilização de um mecanismo de controlo de direção omnidirecional consegue garantir uma boa mobilidade, em grande parte dos casos.

### 2.1.1 Tipos de Rodas

A grande maioria das rodas omnidirecionais existentes rege-se pelo princípio de que a roda é responsável pela tração na direção perpendicular ao eixo do motor, no entanto permite o deslizamento na direção do seu eixo [8]. Para serem omnidirecionais é necessário que as rodas possuam pouco atrito na direção do eixo do motor, de forma a facilitar o movimento nesta direção. É possível a combinação de movimentos de translação e rotação durante o deslocamento do robô, possibilitando-lhe que alcance um destino com uma disposição angular desejada.

Existem dois tipos de rodas omnidirecionais: as de desenho convencional e as de desenho especial. As rodas de desenho convencional podem-se dividir entre as rodas direcionais e os rodízios, enquanto que, em relação às rodas de desenho especial existem as rodas universais e as rodas *mecanum*. Na Figura 2.1 estão representados os dois tipos de rodas de desenho convencional.

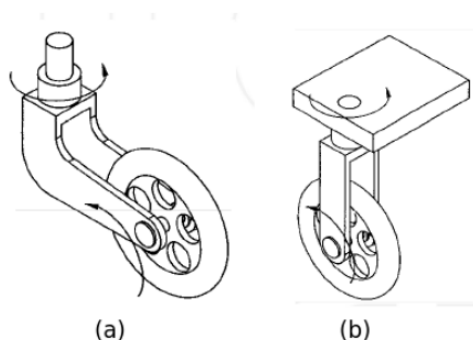


Figura 2.1: Rodas de desenho convencional: (a) Rodízio, (b) Roda direcional [6].

Este tipo de rodas, ditas convencionais, não possuem características holonômicas<sup>1</sup> necessárias para serem consideradas verdadeiramente omnidirecionais. Os veículos com este tipo de rodas para serem omnidirecionais, necessitam de ter na sua configuração pelo menos duas rodas, cada uma das quais com um atuador de direção. Assim, este tipo de sistema não é totalmente omnidirecional, porque necessita que as rodas parem e se reorientem para a direção desejada sempre que efetuar uma trajetória com uma curvatura não contínua [11]. Contudo, estas rodas, pela capacidade que possuem de rotação em torno de si próprias, apresentam características omnidirecionais. Em comparação com as rodas de desenho especial, apresentam uma maior capacidade de carga, assim como maior tolerância a superfícies irregulares.

As rodas de desenho especial têm a particularidade de ter tração num sentido e permitirem movimento passivo noutro. Os tipos de rodas de desenho especial estão ilustrados na Figura 2.2.

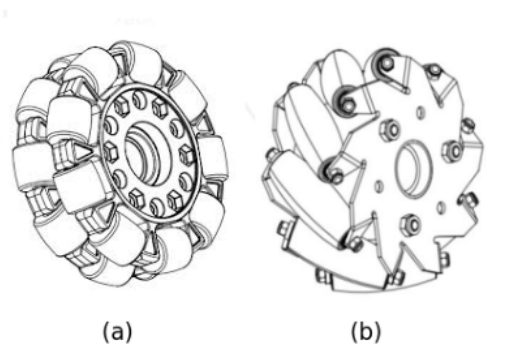


Figura 2.2: Rodas de desenho especial: (a) Universal, (b) Mecanum [6].

---

<sup>1</sup>Holonomia – característica de um sistema cujos vínculos são expressos apenas por um número finito de equações. Na robótica móvel refere-se às restrições cinemáticas do chassis do robô, ou seja, um robô holonômico é um robô sem restrições de movimento no espaço bidimensional  $XY$ [9][10].

Este tipo de rodas caracterizam-se pela presença de rolos, que possuem um movimento não controlado, em torno do seu diâmetro exterior, conferindo-lhe três graus de liberdade (DOF - *Degrees of Freedom*).

Nas rodas universais, estes rolos encontram-se colocados perpendicularmente ao eixo da roda, permitindo que a roda efetue o movimento de rotação normal, mas também o movimento livre na mesma direção que o eixo da roda. Para a obtenção de movimento omnidirecional numa plataforma robótica, com este tipo de rodas, é necessária a presença de pelo menos três rodas. Este tipo específico de rodas é aplicado com diferentes configurações para os rolos externos, como se pode verificar através da análise da Figura 2.3.

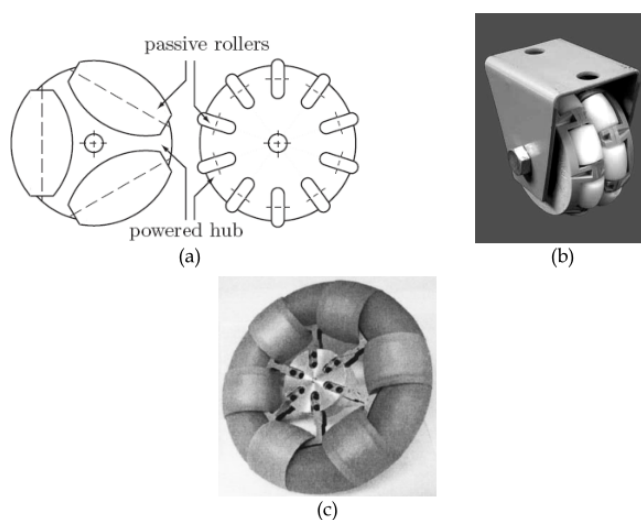


Figura 2.3: Rodas Universais: (a) Simples, (b) Dupla, (c) Alternada [12].

As rodas *mecanum* são semelhantes às rodas universais, com a diferença de que o posicionamento dos rolos exteriores apresenta um ângulo diferente em relação ao eixo da roda, sendo o ângulo convencional de  $45^\circ$ . Esta característica, inerente a este tipo de rodas, permite-lhe que uma parte da força de tração seja aplicada na direção do movimento normal da roda e outra parte seja perpendicular ao eixo de rotação dos rolos. As plataformas robóticas

com este tipo de rodas, obtêm mobilidade omnidirecional através da combinação de pelo menos quatro rodas, cujo posicionamento é efetuado de forma estratégica. A combinação das velocidades de cada uma das quatro rodas, resulta no movimento da plataforma em qualquer direção, quer em termos de translação quer rotação. Na Figura 2.4 é possível visualizar um esquema da roda *mecanum*, representativo do seu movimento rotacional, assim como do movimento passivo dos rolos.

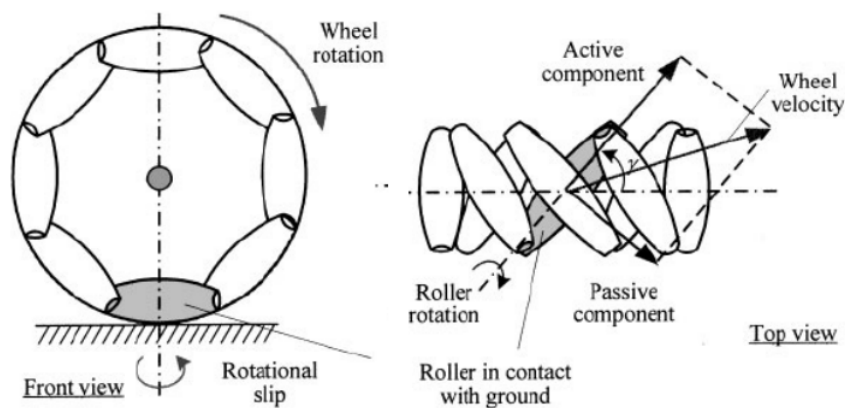


Figura 2.4: Esquema do movimento da roda *mecanum* [12].

De encontro aos tipos de rodas apresentadas anteriormente, na Tabela 2.1 está representada um comparativo, em termos de características positivas e negativas, entre os vários tipos de rodas, segundo a obra “*Bioinspiration and Robotics: Walking and Climbing Robots*”.

Tabela 2.1: Tabela comparativa dos diferentes tipos de rodas [12].

Categoria	Roda	Caraterísticas positivas	Caraterísticas negativas
Desenho Especial	<b>Universal</b>	Compacta e leve; Conceção mecânica simples; Maior disponibilidade comercial.	Contacto descontínuo com o piso ou variação do raio da roda; Sensibilidade às irregularidades do piso.
	<b>Mecanum</b>	Compacta; Alta capacidade de carga.	Contacto descontínuo com o piso Alta sensibilidade às irregularidades do piso; Conceção complexa.
Desenho Convencional	<b>Direcional</b>	Contacto constante com o piso; Alta capacidade de carga; Robusta às condições do piso.	Volumosa e pesada; Alta força de fricção durante viragem; Conceção mecânica complexa
	<b>Rodízio</b>	Contacto constante com o piso; Alta capacidade de carga; Baixa força de fricção durante viragem; Robusta às condições do piso.	Volumosa; Transmissão de sinal e alimentação através de juntas rotativas; Conceção mecânica complexa.

Da análise deste quadro comparativo, pode-se verificar que as rodas de desenho convencional têm um *design* mais volumoso e a sua conceção mecânica é, em geral, mais complexa. Contudo, este tipo de desenho de rodas, apresenta nos diferentes tipos uma maior capacidade de carga, assim como maior robustez em relação às irregularidades do piso. As rodas de desenho especial têm a desvantagem de não realizar contacto contínuo com a superfície durante todo o movimento, assim como de serem mais sensíveis às irregularidades. No entanto, estas apresentam um *design* mais compacto e permitem a obtenção de movimentos de verdadeira omnidirecionalidade, como mencionado anteriormente, graças à sua excelente manobrabilidade.

## 2.2 Configurações de Plataformas Omnidirecionais

Tanto a nível de investigação como a nível do mercado, têm surgido diferentes tipos de robôs móveis omnidirecionais. Estes robôs para serem operados no espaço cartesiano bidimensional devem apresentar três graus de liberdade, dois dos quais são para o movimento translacional e um para o movimento rotacional [13]. No que diz respeito a plataformas robóticas com rodas e locomoção omnidirecionais, existem três configurações principais: com três rodas universais, com quatro rodas universais e com quatro rodas *mecanum*. A Figura 2.5 apresenta os três tipos de configurações de plataformas omnidirecionais mencionados.

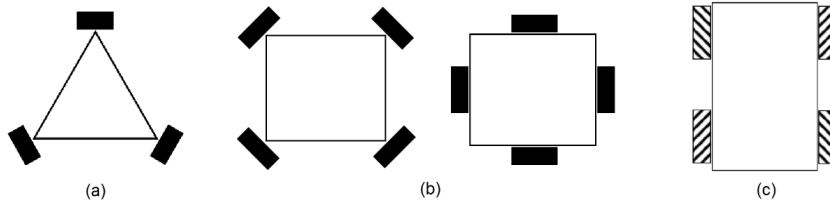


Figura 2.5: Configurações Omnidirecionais: (a) três rodas universais, (b) quatro rodas universais, (c) quatro rodas *mecanum*.

### 2.2.1 Plataforma com Três Rodas Universais

Como representado na Figura 2.5(a), a configuração omnidirecional com três rodas universais apresenta uma estrutura triangular, na qual as três rodas se encontram espaçadas  $120^\circ$ , de forma igual. Cada uma das rodas move-se de forma individual através da interligação de três motores, cujos movimentos permitem que a plataforma se desloque em qualquer direção do plano  $XY$ , consoante a combinação de velocidade e sentido de cada roda. Na Figura 2.6 pode-se visualizar alguns dos movimentos que este tipo de robôs pode apresentar consoante a velocidade e direção das suas três rodas.

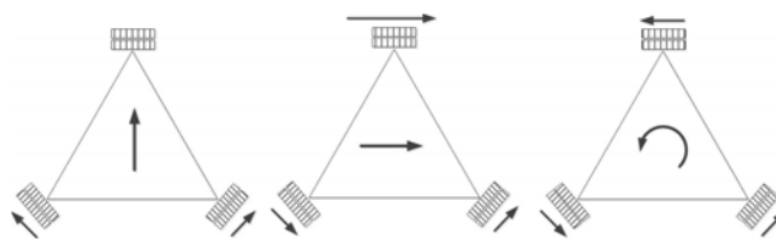


Figura 2.6: Movimentos do Robô Omnidirecional com Três Rodas [14].

Este tipo de configuração é bastante utilizado por vários grupos de investigação, principalmente pela sua boa manobrabilidade e controlo simples. Normalmente as três rodas omnidirecionais universais utilizadas neste tipo de plataforma, proporcionam uma menor tração, uma vez que a força de reação é distribuída por apenas três pontos [14]. Este *design* é mecanicamente mais simples, comparativamente com os outros apresentados, e permite que todas as rodas mantenham o contacto com o solo durante o movimento, mesmo que se encontre em superfícies irregulares. Contudo, esta distribuição triangular das rodas pode apresentar um problema de estabilidade, principalmente quando se desloca ao longo de uma superfície inclinada em que o robô tem o centro de gravidade alto ou em plataformas de tamanho mais reduzido.

### 2.2.2 Plataforma com Quatro Rodas Universais

Na Figura 2.5(b) estão representadas duas vertentes da configuração omnidirecional com quatro rodas universais. Este tipo de plataformas é composto normalmente, por quatro rodas universais localizadas em cada um dos lados, ou em cada um dos vértices, espaçadas  $90^\circ$  de forma igual entre elas. Analogamente à configuração anterior, cada uma das rodas é conectada a um motor e o seu controlo é efetuado de forma independente em relação às outras. Consoante a combinação de sentido e velocidade das rodas atuadas,

a plataforma movimentar-se-á em qualquer direção do espaço bidimensional. Na Figura 2.7 pode-se observar alguns exemplos de combinações possíveis de sentidos nas rodas e respectivos movimentos.

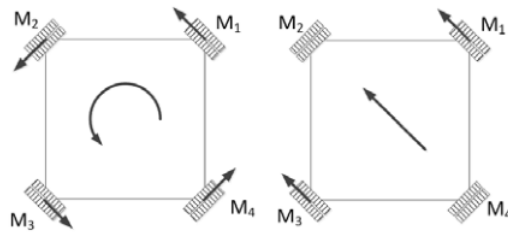


Figura 2.7: Movimentos do Robô Omnidirecional com Quatro Rodas [14].

Este tipo de configuração tem quatro pontos de contacto com a superfície, apresentando uma geometria retangular entre esses pontos, o que lhe permite ter maior tração que o tipo de plataformas apresentado anteriormente, assumindo que as quatro rodas estão sempre em contacto com o chão. Para cumprir este último requisito, por vezes é necessária a aplicação de um sistema de suspensão de forma a distribuir as forças de igual forma pelas quatro rodas [15]. Em comparação com os sistemas de três rodas omnidirecionais, estes são mecanicamente mais complicados, no entanto, permitem maior aceleração e, graças a possuírem mais pontos de apoio, conseguem suportar maiores pesos (caso as rodas a comprar sejam iguais). Em termos de custo de implementação, como estas plataformas apresentam mais quantidade de *hardware*, o seu custo é geralmente superior às plataformas mencionadas na secção 2.2.1. No estudo efetuado por Oliveira (2007) [8], são comparados estes dois tipos de sistemas, de três e quatro rodas omnidirecionais, tendo em conta diferentes parâmetros.

### 2.2.3 Plataforma com Quatro Rodas *mecanum*

A Figura 2.5(c) apresenta a configuração da plataforma robótica com quatro rodas *mecanum*. Para este tipo de configuração são necessárias quatro rodas *mecanum*, das quais duas devem ser “*left-handed*” (com os rolos a  $+45^\circ$  em relação ao eixo da roda) e as outras duas “*right-handed*” (com os rolos a  $-45^\circ$  em relação ao eixo da roda). A Figura 2.8 demonstra estes dois tipos de rodas mencionados.

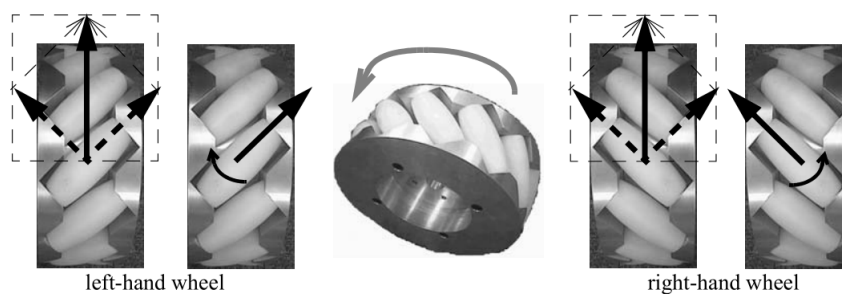


Figura 2.8: Rodas *mecanum*: “*left-handed*” e “*right-handed*” [16].

Como acontece nos exemplos anteriores, o controlo de cada uma das rodas é feito de forma individual, estando cada uma das quatro rodas conectadas a um respetivo motor.

Analisando cada roda de forma individual, a força gerada pela sua rotação atua na superfície por intermédio dos rolos passivos. Nestes rolos a força pode ser dividida em dois vetores, um perpendicular ao eixo do rolo e outro paralelo. A combinação das forças resultantes de cada roda determina o movimento global do veículo. Na Figura 2.9 podem-se visualizar os principais movimentos deste tipo de base omnidirecional, tendo em conta o sentido de rotação de cada uma das rodas.

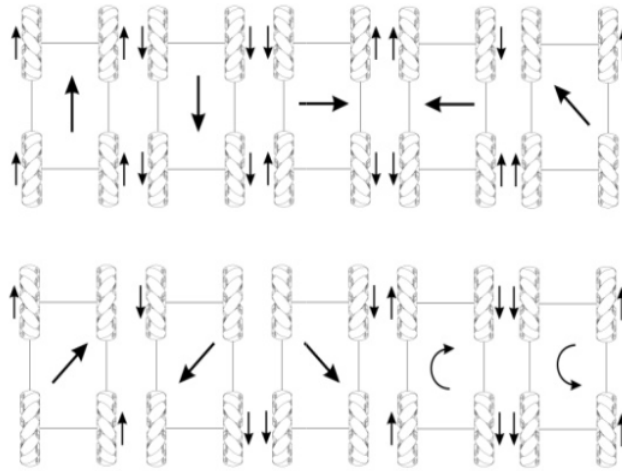


Figura 2.9: Movimentos do Robô Omnidirecional com Quatro Rodas *mecanum* [17].

Estas direções são obtidas tendo em conta o movimento de rotação das rodas a igual velocidade. No entanto, através da variação das velocidades individuais de cada roda, combinada com o sentido de rotação das mesmas é possível conseguir movimentos ao longo de qualquer vetor do plano  $2D$ .

Pelo facto de apenas um ponto de um único rolo se encontrar em contacto com a superfície, durante qualquer instante do seu movimento, este tipo de configuração pode apresentar problemas de derrapagem. Este problema pode ser resolvido ou minimizado, escolhendo o material dos rolos consoante o tipo de superfície em que se vai movimentar. Também esta configuração com quatro rodas, quando se desloca em superfícies com irregularidades, é suscetível a que nem sempre todas as rodas se encontrem em contacto com o chão. Para tal (analogamente ao mencionado na configuração da secção anterior), mediante as condições do terreno em que desloca e a finalidade para o qual são aplicados, os robôs com quatro rodas *mecanum*, por vezes são equipados com sistemas de suspensão.

## 2.3 Modelo Cinemático

A cinemática é o estudo do movimento de um corpo sem preocupação com os efeitos do mesmo, sendo que na robótica móvel esta é aplicada ao movimento dos robôs sem considerar as forças e as massas. A equação da cinemática direta permite a previsão do movimento do robô sabendo a velocidade angular de cada roda. A cinemática inversa possibilita o cálculo da velocidade necessária para cada roda de modo a gerar a velocidade e rotação necessárias para o robô [18]. Através da posição do robô  $(x, y, \theta)$  ao longo do tempo, é possível calcular as velocidades  $(v_x, v_y, \omega)$ , pela aplicação da derivada, como se pode verificar nas Equações (2.1), (2.2) e (2.3).

$$v_x(t) = \frac{dx(t)}{dt} \quad (2.1)$$

$$v_y(t) = \frac{dy(t)}{dt} \quad (2.2)$$

$$\omega(t) = \frac{d\theta(t)}{dt} \quad (2.3)$$

Nas quais  $x$  e  $y$  [m] correspondem à posição do robô,  $\theta$  [rad] à posição angular do robô e  $v_x$ , e  $v_y$  [m/s] à velocidade linear do robô segundo  $x$  e  $y$ . Das velocidades lineares é possível calcular as velocidades  $v$  e  $v_n$ , pela equação da cinemática (2.4).

$$\begin{bmatrix} v(t) \\ v_n(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & \sin(\theta(t)) & 0 \\ -\sin(\theta(t)) & \cos(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_x(t) \\ v_y(t) \\ \omega(t) \end{bmatrix} \quad (2.4)$$

Pela aplicação da cinemática inversa consegue-se determinar as velocidades  $v_x$  e  $v_y$  como representado na Equação (2.5)

$$\begin{bmatrix} v_x(t) \\ v_y(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & -\sin(\theta(t)) & 0 \\ \sin(\theta(t)) & \cos(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v(t) \\ v_n(t) \\ \omega(t) \end{bmatrix} \quad (2.5)$$

### 2.3.1 Cinemática do Robô com Três Rodas

Na Figura 2.10 está representado um esquema com a configuração do robô com três rodas, assim como as forças e velocidades respeitantes a este sistema.

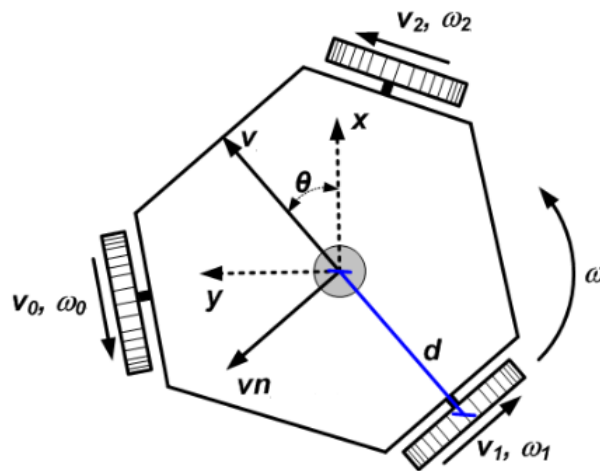


Figura 2.10: Configuração do Robô com Três Rodas [15].

As variáveis presentes nesta Figura têm a seguinte correspondência:

- $d [m]$  - Distância entre o ponto central do robô e as rodas;

- $v_0, v_1, v_2$  [m/s] - Velocidade linear das rodas;
- $\omega_0, \omega_1, \omega_2$  [rad/s] - Velocidade angular das rodas;
- $v, v_n$  [m/s] - Velocidade linear do robô;
- $\omega$  [rad/s] - Velocidade angular do robô;

A equação (2.6) descreve a relação entre as velocidades das rodas com as velocidades da plataforma robótica.

$$\begin{bmatrix} v_0(t) \\ v_1(t) \\ v_2(t) \end{bmatrix} = \begin{bmatrix} -\sin(\frac{\pi}{3}) & \cos(\frac{\pi}{3}) & d \\ 0 & -1 & d \\ \sin(\frac{\pi}{3}) & \cos(\frac{\pi}{3}) & d \end{bmatrix} \cdot \begin{bmatrix} v(t) \\ v_n(t) \\ \omega(t) \end{bmatrix} \quad (2.6)$$

Invertendo-se a matriz, obtêm-se as equações das velocidades do robô em função das velocidades das rodas, como se pode verificar na Equação (2.7).

$$\begin{bmatrix} v(t) \\ v_n(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} -\frac{\sqrt{3}}{3} & 0 & \frac{\sqrt{3}}{3} \\ \frac{1}{3} & -\frac{2}{3} & \frac{1}{3} \\ \frac{1}{3 \cdot d} & \frac{1}{3 \cdot d} & \frac{1}{3 \cdot d} \end{bmatrix} \cdot \begin{bmatrix} v_0(t) \\ v_1(t) \\ v_2(t) \end{bmatrix} \quad (2.7)$$

Assim as velocidades linear, normal e angular do robô em função das velocidades das rodas são representadas pelas seguintes equações:

$$v(t) = \frac{\sqrt{3}}{3} \cdot (v_2(t) - v_0(t)) \quad (2.8)$$

$$v_n(t) = \frac{1}{3} \cdot (v_2(t) + v_0(t) - \frac{2}{3} \cdot v_1(t)) \quad (2.9)$$

$$\omega(t) = \frac{1}{3 \cdot d} \cdot (v_0(t) + v_1(t) + v_2(t)) \quad (2.10)$$

### 2.3.2 Cinemática do Robô com Quatro Rodas

Na Figura 2.11 apresenta-se a configuração do robô com quatro rodas, assim como as velocidades a ele associadas. Todas as variáveis presentes na figura são análogas ao exemplo anterior, com a adição de mais uma variável de velocidade linear ( $v_3$ ) e velocidade angular ( $\omega_3$ ), por esta plataforma apresentar mais uma roda.

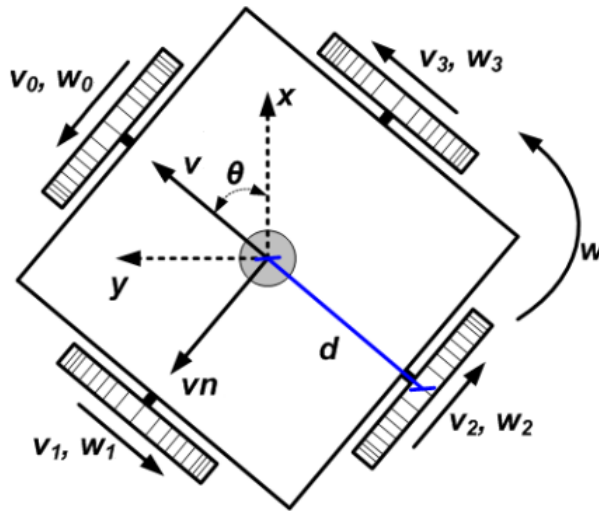


Figura 2.11: Configuração do Robô com Quatro Rodas [15].

A relação entre a velocidade de cada uma das quatro rodas com as velocidades do robô é representada pela equação de matrizes (2.11).

$$\begin{bmatrix} v_0(t) \\ v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & d \\ -1 & 0 & d \\ 0 & -1 & d \\ 1 & 0 & d \end{bmatrix} \cdot \begin{bmatrix} v(t) \\ v_n(t) \\ \omega(t) \end{bmatrix} \quad (2.11)$$

Desta equação obtêm-se as velocidades individuais para cada uma das rodas, como demonstram as equações (2.12), (2.13), (2.14) e (2.15).

$$v_0 = v_n(t) + d \cdot \omega(t) \quad (2.12)$$

$$v_1 = -v(t) + d \cdot \omega(t) \quad (2.13)$$

$$v_2 = -v_n(t) + d \cdot \omega(t) \quad (2.14)$$

$$v_3 = v(t) + d \cdot \omega(t) \quad (2.15)$$

Como efetuado no exemplo anterior, através da aplicação da cinemática direta consegue-se obter as equações das velocidades do robô em função das velocidades das rodas. Neste caso específico, como a matriz não é quadrada, não se consegue inverter, significando que o sistema é redundante. A redundância do sistema permite que sejam efetuados pedidos de velocidades que são impossíveis de executar, sendo isto evidenciado pela existência de duas equações para o cálculo de  $\omega(t)$ . Deste modo, através da relação das equações anteriores obtêm-se as seguintes equações das velocidades linear, normal e angulares.

$$v(t) = \frac{v_3 - v_1}{2} \quad (2.16)$$

$$v_n(t) = \frac{v_0 - v_2}{2} \quad (2.17)$$

$$\omega_1(t) = \frac{v_1 + v_3}{2 \cdot d} \quad (2.18)$$

$$\omega_2(t) = \frac{v_0 + v_2}{2 \cdot d} \quad (2.19)$$

Como mencionado, a redundância do sistema faz com que existam duas soluções para a velocidade angular. Este caso pode ser vantajoso em termos

de melhoria da estimativa do valor desta velocidade, aplicando a média entre os dois resultados, como apresentado na equação (2.20).

$$\omega(t) = \frac{v_0 + v_1 + v_2 + v_3}{4 \cdot d} \quad (2.20)$$

### 2.3.3 Cinemática do Robô com Quatro Rodas *Mecanum*

No que diz respeito ao modelo cinemático do robô com quatro rodas *mecanum*, encontra-se representado na Figura 2.12 um esquema com as seguintes variáveis:

- $v_1, v_2, v_3, v_4$  [ $m/s$ ] - Velocidade linear das rodas, em que  $v_i = R \times \omega_i$ ;
- $R$  [ $m$ ] - Raio das rodas;
- $v_{1r}, v_{2r}, v_{3r}, v_{4r}$  [ $m/s$ ] - Velocidade perpendicular ao eixo do rolo;
- $\omega_1, \omega_2, \omega_3, \omega_4$  [ $rad/s$ ] - Velocidade angular das rodas;
- $l$  e  $L$  [ $m$ ] - Distância entre o centro do robô e o centro da roda, em  $Y$  e  $X$ ;
- $v, v_n$  [ $m/s$ ] - Velocidade linear do robô;
- $\omega$  [ $rad/s$ ] - Velocidade angular do robô;
- $\alpha$  [ $^\circ$ ] - Ângulo entre o eixo do rolo e o eixo da roda.

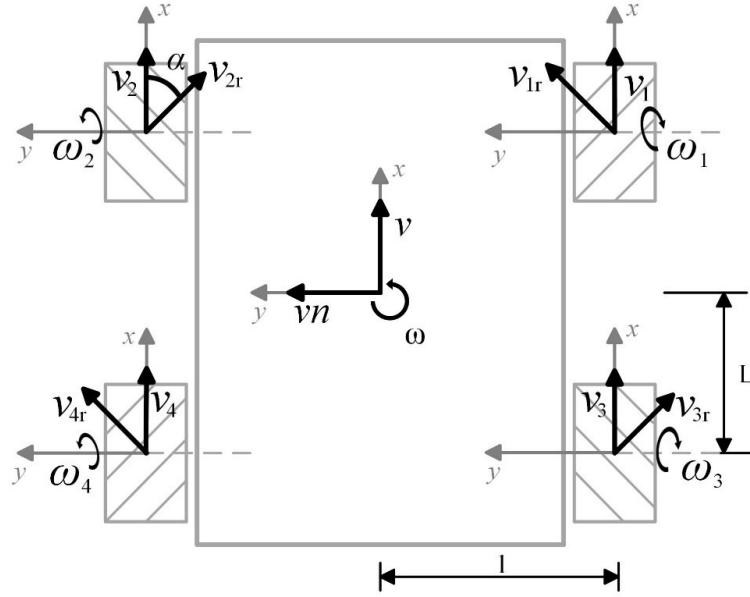


Figura 2.12: Configuração do Robô com Quatro Rodas Mecanum.

Como mencionado na Secção 2.2.3 , quando se atuam as rodas mecanum, os rolos convertem uma porção da força de rotação da roda numa força perpendicular ao eixo dos mesmos. Sendo que na maioria das rodas mecanum  $\alpha = 45^\circ$ , consegue-se definir a velocidade de cada roda em função da velocidade do robô. As equações 2.21 e 2.22 definem as componentes em  $X$  e  $Y$  da velocidade da roda um (1).

$$v_{1X} = v(t) + l \cdot \omega(t) \quad (2.21)$$

$$v_{1Y} = v_n(t) + L \cdot \omega(t) \quad (2.22)$$

Tendo em consideração a direção de cada roda e exprimindo as suas componentes  $X$  e  $Y$  em função das velocidades lineares e angular tem-se:

$$v_1(t) = v(t) + v_n(t) + (l + L) \cdot \omega(t) \quad (2.23)$$

$$v_2(t) = v(t) - v_n(t) - (l + L) \cdot \omega(t) \quad (2.24)$$

$$v_3(t) = v(t) - v_n(t) + (l + L) \cdot \omega(t) \quad (2.25)$$

$$v_4(t) = v(t) + v_n(t) - (l + L) \cdot \omega(t) \quad (2.26)$$

Podendo estas ser apresentadas na forma matricial, representando assim a cinemática inversa deste tipo de plataformas.

$$\begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \\ v_4(t) \end{bmatrix} = \begin{bmatrix} 1 & 1 & (l + L) \\ 1 & -1 & -(l + L) \\ 1 & -1 & (l + L) \\ 1 & 1 & -(l + L) \end{bmatrix} \cdot \begin{bmatrix} v(t) \\ v_n(t) \\ \omega(t) \end{bmatrix} \quad (2.27)$$

Como no caso do tipo de plataforma explicada na Secção 2.3.2, a cinemática inversa desta configuração é representada por uma matriz que não é quadrada, representando assim um sistema redundante (com mais equações do que incógnitas). Na equação seguinte encontra-se representada a equação da cinemática direta do robô [19].

$$\begin{bmatrix} v(t) \\ v_n(t) \\ \omega(t) \end{bmatrix} = \frac{1}{4} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -\frac{1}{(l+L)} & \frac{1}{(l+L)} & -\frac{1}{(l+L)} & \frac{1}{(l+L)} \end{bmatrix} \cdot \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \\ v_4(t) \end{bmatrix} \quad (2.28)$$

## 2.4 Aplicações dos Robôs Omnidirecionais

Atualmente são múltiplas as aplicações de robôs omnidirecionais em diferentes áreas e tarefas. Pelas suas características holonómicas, a aplicação deste tipo de plataformas tem chamado a atenção tanto na investigação em robótica, como na sociedade em geral. A sua aplicação é transversal, desde a indústria aos serviços e até nos domicílios. Devido às suas características de manobrabilidade, agilidade e holonomia, revelam-se bastante úteis na

qualidade de robôs autônomos em diversas tarefas como: prestadores de serviços, assistência domiciliar, cuidados de enfermagem, limpeza, manipulação móvel, entre outros [20].

### 2.4.1 Indústria

Na área da indústria já não é novidade a presença de veículos autônomos. Este tipo de plataformas são normalmente utilizadas com o intuito de mover peças de trabalho, como equipamentos de carga ou até equipadas com ferramentas (como por exemplo braços robóticos) e seguem um caminho pre-determinado ou criado virtualmente. Inicialmente estes veículos, designados AGVs (*Automated Guided Vehicles*), movimentavam-se nos ambientes industriais através do seguimento de linhas, marcas na superfície, ou até através de ímanes para orientação do seu movimento.

Na atualidade começam a entrar em meios industriais veículos autônomos livres, ou seja, cuja navegação é possível através da criação de trajetórias virtuais, atualmente denominados por AMR (*Automated Mobile Robots*). Este tipo de navegação é normalmente baseada em scanners laser, que permitem a criação de um mapa para auto localização e deteção de obstáculos [21]. A principal distinção entre um AGV e um AMR é a “inteligência”.

Os AMRs não são restritos a rotas fixas e através do mapa criado conseguem navegar dinamicamente, planeando os seus próprios caminhos, de forma a efetuar o trajeto mais rápido e eficiente. Outra diferença entre estes dois tipos de veículos autônomos é que, ao contrário dos AGVs, os AMRs, através da sua capacidade sensorial, reagem ao meio ambiente em que se encontram, conseguindo efetuar o seu trabalho com segurança, não importando o quão ocupado se encontra o seu meio circundante [22].

Os principais fatores de navegação em ambiente industrial são a precisão,

segurança e a eficiência. Estes fatores complementam-se de certa forma, pois a navegação de alta precisão é um pré-requisito para a segurança e significa que a plataforma seguirá de perto os movimentos planeados [23]. Para maior adaptação ao ambiente fabril e maior flexibilidade no deslocamento começa a ser comum a utilização de plataformas autónomas equipadas com sistemas de locomoção omnidirecionais.

Um exemplo é o *omniMove*, desenvolvido pela *KUKA*, representado na Figura 2.13, que consiste numa plataforma robótica modular omnidirecional, que permite o transporte de cargas até 90 toneladas (na versão máxima) com um comprimento até 30 metros.



Figura 2.13: Plataforma Omnidirecional KUKA omniMove [24].

O *KUKA omniMove* tem a possibilidade de controlo manual ou de se deslocar de forma autónoma através de orientação ótica ou indutiva. O seu método de locomoção omnidirecional, baseado nas típicas rodas *mecanum*, foi desenvolvido pela própria *KUKA*, cujas rodas podem ir desde 250 mm até 570 mm de diâmetro e aguentar cargas desde 150 kg até 3000 kg, cada roda. Cada roda é atuada por um motor elétrico individualmente e a sua configuração composta por nove rolos livres a 45° em relação ao eixo da roda, permite movimentos de translação e rotação em espaços mais apertados. Apesar do

seu tamanho e capacidade de carga enormes, esta plataforma navega em segurança com uma precisão de movimento de 1mm com a máxima carga [25]. Para além do *omniMove*, a *KUKA* tem outras plataformas e robôs móveis, equipados com o mesmo sistema de locomoção omnidirecional, mas com diferentes funcionalidades. Exemplos disso são o *Triple Lift* - uma plataforma omnidirecional com um elevador que sobe até 8 metros suportando cargas até 1000 quilos; e o *KMR iiwa* - um robô autónomo, omnidirecional, equipado com um braço robótico assistente de trabalho individual.

### 2.4.2 Domicílio e Cuidados de Saúde

Tendo em conta o aumento considerável do número de idosos, e indivíduos com necessidades específicas, que necessitam de cuidados especiais que se tem verificado nos dias de hoje, a utilização de robôs de serviço inteligentes que possam auxiliar as pessoas nas suas atividades diárias começa a ser uma prática a adotar. Sendo, por vezes, difícil a deslocação de robôs no espaço interior dos domicílios, devido à grande quantidade de obstáculos, de tamanhos e formas variadas, a opção por métodos de locomoção omnidirecionais tem-se demonstrado uma solução eficaz para contorno deste problema.

Já em 1999, Wada e Asada propuseram um sistema de locomoção omnidirecional reconfigurável, aplicado a cadeiras de rodas [26]. Em 2011, Zou, apresentou um robô omnidirecional de assistência domiciliar com três rodas, equipado com um sistema de posicionamento interno, um braço robótico e uma câmara. Este sistema robótico possui um sistema de deteção de obstáculos, composto por sensores de infravermelhos colocados em todo o redor do robô [27]. Com o intuito de transportar e movimentar pessoas idosas no interior do seu domicílio Tan *et al.* desenvolveram, em 2012, uma plataforma robótica almofadada, composta por um sistema de três rodas omnidirecionais universais [28]. Em 2015, Fukuda *et al.* projetaram um robô móvel, que tem

funcionalidades similares a uma bengala, e auxilia a caminhada de pessoas idosas ou com deficiências motoras. Este robô é composto por uma vara, um conjunto de sensores e uma base omnidirecional, que permitem reconhecer a "intenção de andar" do utilizador, ajudando-o nessa tarefa [29].

Na atualidade já existem empresas de renome a desenvolver e a colocar no mercado soluções robóticas de serviço domiciliar e de prestação de cuidados de saúde, dotadas de mobilidade omnidirecional. Um exemplo disto é o HSR (*Human Support Robot*), desenvolvido pela Toyota com o objetivo de auxiliar pessoas idosas, ou com limitações especiais, nas suas atividades diárias, com vista a poder ser um instrumento útil a todas as pessoas, num futuro próximo.

A Toyota começou o desenvolvimento do HSR em 2012 e até então foram concebidas várias versões. A versão atual, representada na Figura 2.14, é capacitada de uma plataforma omnidirecional de 430 *mm* de diâmetro, movendo-se à velocidade máxima de 0,8 *m/s*, que torna o seu movimento muito menos restringido.



Figura 2.14: Robô Toyota HSR [30].

Possui também um braço articulado de 60 *cm* e um corpo telescópico que vai desde os 100 *cm* até 135 *cm* de altura, que possibilitam a sua atuação num espaço de trabalho considerável, mesmo apresentando uma arquitetura compacta [30].

O HSR é dotado de três modos de funcionamento: *pick-up*, *fetch* e *Manual Control*. No modo *pick-up*, o robô utiliza uma pinça simples para agarrar vários tipos de objetos, como canetas ou comandos de TV, e para objetos mais difíceis de agarrar, como papéis ou cartões, utiliza um sistema de vácuo, implementado na extremidade do braço. O modo *fetch* permite que, através de comandos de voz ou da interface tátil presente na cabeça do robô, o utilizador possa pedir ao robô para lhe ir buscar objetos, previamente sinalizados. No modo *Manual Control* as tarefas que normalmente estão para além das capacidade autónomas do HSR, podem ser realizadas manualmente através da interface do utilizador. O controlo manual também pode ser efetuado remotamente, o que permite a outras pessoas, como membros da família, a possibilidade de controlar o robô e comunicar com o proprietário através de Skype ou outros serviços, por intermédio do *display* presente na cabeça do HSR.

## 2.5 *Robot Operating System (ROS)*

O *Robot Operating System* é uma *framework* bastante utilizada no desenvolvimento de sistemas robóticos. A sua filosofia assenta no desenvolvimento de *software* que possa ser transversal a vários robôs, apenas com pequenas mudanças no código.

O ROS foi originalmente desenvolvido em 2007 pela *Stanford Artificial Intelligence Laboratory* (SAIL) e a partir de 2008 foi continuado pela *Willow Garage*, instituto de investigação robótica com mais de 20 instituições par-

ceiras [31].

Após o seu desenvolvimento, um grande número de centros de investigação começaram a utilizar esta *framework* para o desenvolvimento dos seus projetos, adicionando *hardware* e partilhando os seus exemplos de código. Também as empresas começaram a adaptar os seus produtos para ser utilizado ROS. Os sensores e atuadores utilizados na robótica também foram, e têm vindo a ser, adaptados para serem compatíveis com ROS. Devido à sua versatilidade, tem existido um aumento crescente do número de dispositivos suportados por esta *framework*.

ROS não é um sistema operativo, mas fornece serviços tais como abstração de *hardware*, controlo de dispositivos de baixo nível, passagem de mensagem entre processos e gestão de pacotes. Um sistema ROS é composto por um número de nós independentes, que comunicam uns com os outros utilizando um modelo de mensagens do tipo *publisher/subscriber*.

### 2.5.1 Modelo de Comunicação *Publisher/Subscriber*

No modelo de comunicação *publisher/subscriber* do ROS, os *nodes* comunicam entre si, enviando mensagens através de tópicos. Um *publisher* é um *node* que escreve mensagens num tópico e um *subscriber* é um *node* que lê essas mensagens do tópico. Um determinado tópico pode ter múltiplos *publishers* e *subscribers*, assim como um único *node* pode publicar e/ou subscrever vários tópicos.

Um esquema deste modelo de comunicação é ilustrado na Figura 2.15, representativo da comunicação entre dois *nodes* denominados *camera* e *visualizer*, executados na mesma máquina. Para começar a escrever mensagens num tópico, o *publisher* (*camera* na Figura 2.15), necessita de se registar no nó principal (*Master node*), que possui um registo de todos os *publishers* e *subs-*

*scribers* (Figura 2.15 (a)). O registo no nó mestre é necessário para permitir que outros *nodes*, que queiram subscrever aquele tópico, consigam localizar o *publisher*. Uma vez registado, o *publisher* começa a escrever mensagens no tópico (Figura 2.15 (b)). Para um *node* receber as mensagens de um tópico tem que o subscrever. Assim, como se verifica na Figura 2.15 (b), o *node* comunica com o *Master* solicitando o endereço do *node* que publica esse tópico. Depois de obter o seu endereço o *subscriber* contacta diretamente o *publisher*, através de TCPROS, um protocolo que utiliza *sockets* TCP/IP. Uma vez estabelecida a ligação entre *nodes* as mensagens são passadas diretamente do *publisher* para *subscriber* por intermédio do tópico (Figura 2.15 (c)).

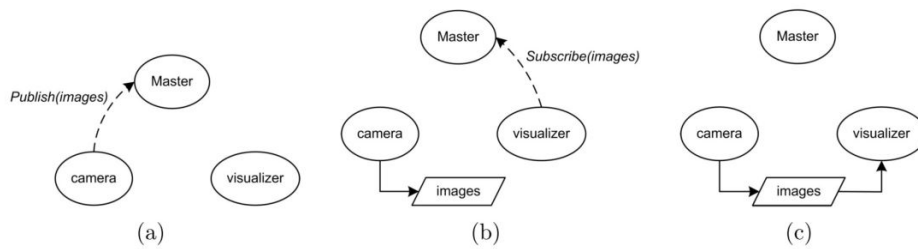


Figura 2.15: Comunicação entre dois ROS *Nodes* utilizando o método *Publisher/Subscriber* [32].

O ROS também fornece um conjunto de ferramentas e bibliotecas para a escrita e execução de código através de múltiplos computadores [33].

A arquitetura do ROS foi projetada e dividida em três níveis principais de conceitos:

- *Filesystem Level*;
- *Computation Graph Level*;
- *Community Level*.

### 2.5.2 *Filesystem Level*

O primeiro nível é o *Filesystem Level*, no qual um grupo de conceitos é utilizado para esclarecer como o ROS é formado internamente, a sua estrutura de pastas e os arquivos mínimos necessários para correr um programa em ROS [31]. À semelhança de um sistema operativo, um programa ROS está dividido em pastas e essas pastas possuem ficheiros que descrevem as suas funcionalidades. Os componentes principais do sistema de arquivos são:

- **Pacote:** Um pacote possui a estrutura e conteúdo mínimos para criar um programa em ROS. É a unidade de organização de *software* do código em ROS. Cada pacote pode conter bibliotecas, *scripts*, executáveis, entre outros;
- **Manifestos:** Um manifesto fornece informação acerca de um pacote, dependências, *flags* de computação e outros;
- **Stacks:** São conjuntos de pacotes com funcionalidades relacionadas;
- **Tipos de mensagem:** Uma mensagem é a informação que um processo envia a outro processo. O ROS tem um conjunto de diferentes tipos de mensagens padrão;
- **Tipos de serviço:** Descrições de serviço definem a estrutura de dados de um pedido e resposta dos serviços em ROS.

### 2.5.3 *Computation Graph Level*

O segundo nível é onde acontecem as comunicações entre processos e sistemas. Deste nível fazem parte os conceitos e sistemas que o ROS apresenta para lidar com todos os processos, para configuração e para comunicar com um ou mais computadores. O ROS cria uma rede na qual todos os processos

estão conectados. Qualquer nó do sistema pode aceder a esta rede, interagir com outros nós, ver a informação que eles estão a enviar e transmitir dados para a rede. Os componentes básicos deste nível são:

- **Nós (*Nodes*):** Os nós são processos onde a computação é feita. Para um processo comunicar com outro é necessário criar um nó que contenha esse processo e se conecte à rede ROS;
- **Mestre (*Master*):** O *master* atribui nomes e serviços de registo a todos os nós do sistema. O papel principal do *master* é permitir que os ROS *nodes* individuais se localizem uns aos outros. Uma vez que os nós se localizam, estabelecem ligação uns com os outros de forma *peer-to-peer*<sup>2</sup>;
- **Servidor de Parâmetros:** O servidor de parâmetros possibilita o armazenamento de dados utilizando chaves num local centralizado. Este parâmetro permite a configuração de nós durante a sua execução ou a alteração do seu funcionamento;
- **Mensagem:** Os nós comunicam entre eles através de mensagens. Uma mensagem contém dados que enviam informação para outros nós. Existem vários tipos de mensagens e o utilizador pode desenvolver o seu próprio tipo;
- **Tópicos:** Um tópico é um nome que se utiliza para identificar o conteúdo de uma mensagem enviada por um nó. O envio e receção de mensagens entre nós entende-se como publicação e subscrição de tópicos. Um único tópico pode ser publicado e subscrito por diferentes nós, assim como um único nó pode publicar e subscrever vários tópicos;

---

<sup>2</sup>Tipo de comunicação que permite a partilha de ficheiros entre duas partes de forma direta, ou seja, sem a existência de um alojamento intermédio do ficheiro em questão.

- **Serviços:** Os serviços dão a possibilidade de interação com os nós. Quando é publicado um tópico, os dados podem ser enviados de diferentes formas, mas quando é necessário um pedido ou uma resposta de um nó é necessária a utilização de um serviço;
- **Bags:** *Bags* são ficheiros próprios do ROS para guardar toda a informação das mensagens, tópicos, serviços e outros. Os *bags* são bastante úteis para armazenar dados de sensores ou efetuar *debug*, principalmente em sistemas robóticos complexos.

#### 2.5.4 *Community Level*

Do *Community Level* fazem parte todas as ferramentas e conceitos necessários para a partilha de conhecimento, algoritmos e códigos por parte dos ROS *developers*. Este nível é importante porque, o crescimento ROS é tanto maior quanto maior for a contribuição da sua comunidade.

## 2.6 Modelo de Odometria

A odometria é um método de localização que não depende de referências externas para a estimativa da posição. Por este método, a informação da localização obtém-se através da deteção do movimento dos motores utilizando para tal sensores (tais como *encoders*) que permitem o cálculo do deslocamento.

Na odometria, a posição atual de uma plataforma móvel é determinada medindo o deslocamento efetuado desde a última posição anterior conhecida. A medição do deslocamento fica a cargo dos *encoders* que detetam a rotação dos eixos dos motores que conduzem as rodas do robô. As leituras dos *encoders* são utilizadas para calcular as distâncias percorridas por cada uma

das rodas desde a última posição conhecida. A partir do conhecimento das distâncias percorridas por cada uma das rodas, os deslocamentos, quer linear quer de rotação, podem ser obtidos através da utilização dos cálculos de odometria [34].

As vantagens da odometria estão diretamente relacionadas com a facilidade de implementação, a boa precisão a curto prazo, o baixo custo de implementação e permite taxas de amostragem muito altas [32]. Por estas razões a odometria é o método mais utilizado para estimação de localização de robôs móveis

### 2.6.1 Erros na Odometria

A odometria é um método de localização que proporciona uma boa estimativa da posição de um robô, mas apenas para curtas distâncias devido à acumulação de erros. A cada iteração do algoritmo de odometria, mais possibilidade de erro é introduzido nos valores de  $x$ ,  $y$  e  $\theta$  da posição. As principais causas de erro na aplicação de modelos de odometria são:

- **Deslizamento da roda**

O deslizamento é, geralmente, a maior causa de erros na odometria. Isto porque é um fator que, mesmo em curtas distâncias, pode influenciar significativamente o cálculo da odometria. Esta questão tem especial relevância em plataformas com rodas omnidirecionais, pelo facto de uma parte do deslocamento destas rodas ser efetuado de forma não controlada (movimento dos rolos);

- **Terreno irregular**

A odometria é utilizada em robôs em que seja assumido que à partida se vão movimentar em superfícies planas. Existem movimentos que

a odometria não consegue controlar em superfícies irregulares (como solavancos), pois esta depende da medida da rotação das rodas. Assim, quanto mais irregular for o terreno, maior é a probabilidade de existirem erros de odometria;

- **Parâmetros geométricos**

Erros de medição entre distâncias de eixos ou de diâmetros das rodas também são uma causa para a existência de erros na odometria. Um erro recorrente é assumir-se que na teoria as rodas de um sistema têm todas os mesmo tamanho e no entanto, mesmo sendo pequenas diferenças, isso não acontece. Sendo as dimensões das rodas um parâmetro a considerar no cálculo das equações de odometria, estas discrepâncias vão também contribuir para um possível erro.

Esta página foi intencionalmente deixada em branco.

## Capítulo 3

# Projeto e Implementação de *Hardware*

*Este capítulo é dedicado ao hardware utilizado e desenvolvido tendo por base o estudo de temáticas abordadas no capítulo anterior. Será apresentada a arquitetura do sistema na qual assenta todo o projeto, assim como serão explicados todos os seus componentes integrantes e respetivos conceitos. Também serão elucidadas e justificadas opções de hardware em detrimento, de outras e por fim, é mencionado todo o processo de desenvolvimento da estrutura da base omnidirecional.*

Com vista a satisfação das necessidades requeridas pelo projeto, foi idealizada uma solução de conceção da plataforma robótica omnidirecional, baseada no estudo efetuado e apresentado no capítulo 2.

Desta forma, numa fase inicial do projeto, de forma agilizar o desenvolvimento e avaliar a sua viabilidade, foi adquirida por parte da empresa, uma plataforma omnidirecional, em *kit*, com quatro rodas *mecanum*, composta pela estrutura, rodas, motores e *encoders*.

Após a validação da viabilidade desta primeira plataforma e concretização da mesma, propôs-se o desenvolvimento de uma outra base omnidirecional, com uma configuração diferente. Esta plataforma seria desenvolvida na íntegra, desde o desenho da estrutura da base, passando pela escolha do todo o seu *hardware*, até à sua concretização física e controlo dos seus componentes.

Em termos gerais, uma base omnidirecional é composta por um conjunto de três ou mais rodas, com características específicas, controladas de forma independente umas das outras, que, quando atuadas consoante diferentes combinações, permitem que a plataforma se desloque em qualquer direção do plano bidimensional e sobre si própria. Para o desenvolvimento deste tipo de plataforma é necessária a compreensão do funcionamento e características de alguns componentes de *hardware* essenciais, sem os quais não seria possível a execução deste projeto. Estes componentes, em termos gerais são análogos aos dois tipos de configurações omnidirecionais e, no seu conjunto, são parte integrante da arquitetura do sistema que o projeto tem por base.

### **3.1 Arquitetura Geral do Sistema**

A arquitetura geral do sistema é um dos requisitos mais básicos com vista à implementação e respetiva compreensão do mesmo. Esta descreve o sistema de uma forma simplificada assim como nos permite ter uma noção geral da estrutura do sistema desenvolvido.

Tendo como finalidade o cumprimento do objetivo mencionado na Secção 1.2, foi idealizado o desenvolvimento de uma base móvel, dotada de mobilidade omnidirecional, composta por quatro rodas também omnidirecionais. As rodas seriam atuadas através de motores de forma individual, cujo controlo seria processado numa Unidade Central de Processamento (CPU - *Central Processing Unit*) por intermédio de controladores de motores. De forma a

poder estimar a posição da base no plano  $XY$ , seriam utilizados *encoders*, para efetuar leituras do movimento dos quatro motores e enviar esse *feedback* para o CPU. Na Figura 3.1 estão representados os blocos principais de *hardware*, preponderantes para o desenvolvimento de uma base omnidirecional.

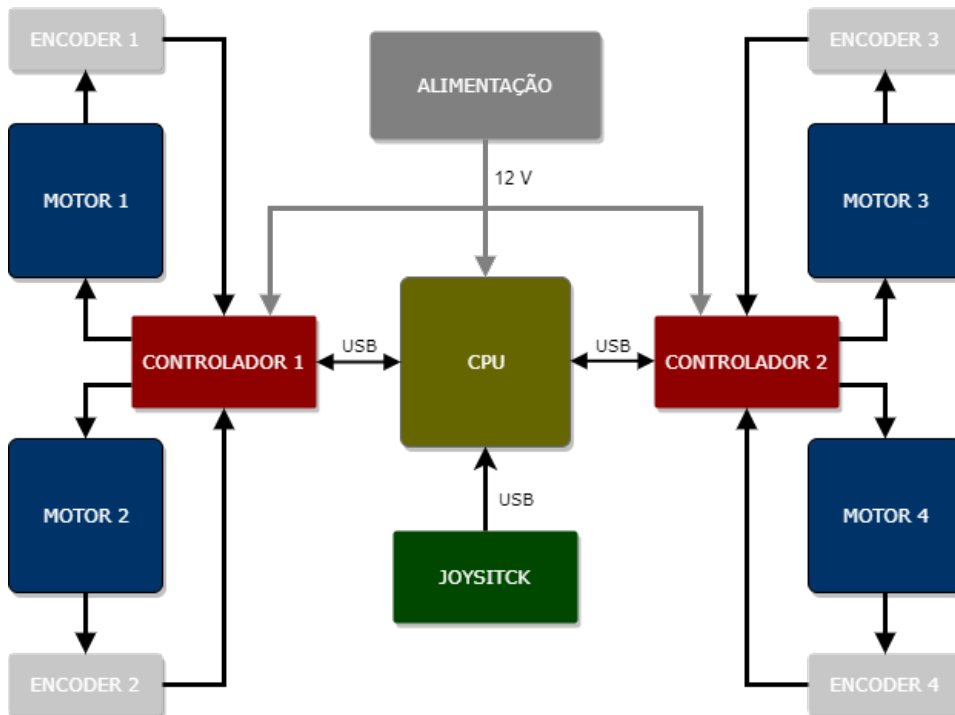


Figura 3.1: Arquitetura Geral do Sistema.

## 3.2 Motores

Um motor é um dispositivo de conversão de energia, que tem como funcionalidade converter energia elétrica em energia mecânica sob a forma de movimento rotativo.

Este sistema eletromecânico é caracterizado por um conjunto específico de parâmetros, que quando conhecidos, permitem nos prever, até certo ponto, a sua performance.

Em robótica móvel, os motores de Corrente Alternada (CA) raramente são utilizados, pelo facto da alimentação dos robôs, na sua maioria, ser efetuada através de Corrente Contínua (CC) proveniente de baterias. Para além disso, uma vez que os seus componentes eletrónicos também são alimentados por CC, é conveniente utilizar o mesmo tipo de alimentação para os atuadores. Assim, para desempenhar a função de atuadores a auxiliar o processo de locomoção da base, apenas foram considerados os motores CC.

Os motores CC funcionam através da criação de campos magnéticos, cuja atração e repulsão induzem o movimento giratório do rotor. Tendo em conta a sua composição e o seu funcionamento, quando aplicados a projetos de robótica, os tipos de motores mais utilizados são os motores CC *Brushed* e os motores CC *Brushless*.

### 3.2.1 Motores CC *Brushed*

Os motores *brushed* utilizam comutadores elétricos internos (escovas - *brushes*) que invertem a polaridade das bobinas ativas enquanto interagem com ímanes com polaridade fixa. A velocidade é controlada através da variação de tensão de alimentação, o que representa uma grande vantagem em comparação com os motores *brushless* que requerem um controlador de velocidade trifásico.

As três partes principais de um motor *brushed* são: o rotor, o estator e o comutador/*brushes* [35]. A parte do estator do motor contém os ímanes permanentes. Estes ímanes são dispostos em pares polares, de configuração "norte-sul", alternadamente. Desta forma, neste tipo de motores, existe sempre um número de ímanes com o polo sul virado para dentro igual aos de polo norte. O rotor contém uma armadura com bobinas de cobre, que quando alimentado, metade desta armadura terá um polo norte magnético

induzido eletricamente que atrairá o polo sul do íman permanente, enquanto que o polo sul da armadura atrai o polo norte do íman permanente. Isto fará com que a armadura e o eixo conectado rodem em direção ao equilíbrio. À medida que a armadura se aproxima da sua posição de equilíbrio magnético, a polaridade magnética das bobines é invertida pelo comutador elétrico e a inércia do rotor leva-lo-á até uma nova posição de equilíbrio. Na Figura 3.2 pode-se observar a estrutura esquematizada do funcionamento de um motor *brushed*.

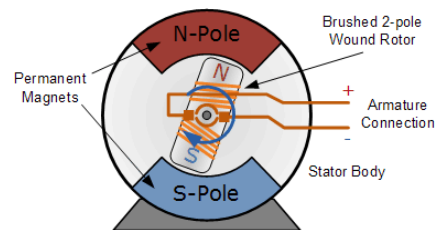


Figura 3.2: Motor CC *Brushed* [36].

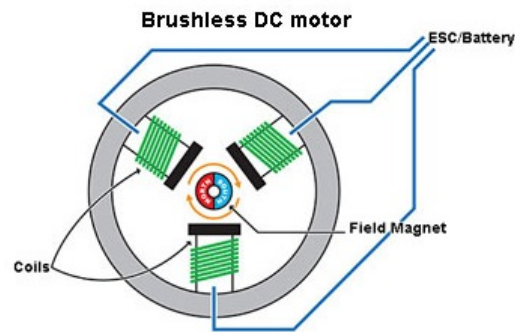
A utilização deste tipo de motores CC também apresenta algumas desvantagens, relacionadas principalmente com as escovas dentro do motor. Estas, ao longo do tempo, acabam por apresentar desgaste, podendo, nos casos em que o motor se encontra permanentemente ativo, necessitar de substituição. Outra desvantagem está relacionada com a corrente que atravessa as bobines que aumenta a produção de calor e alguma quantidade de ruído eletromagnético. Este último fator, pode causar problemas em aplicações robóticas que utilizem comunicações rádio ou que tenham circuitos ou sensores eletrônicos com os quais o ruído possa interferir. Apesar destas desvantagens mencionadas, os motores *brushed* são um dos tipo de motores mais utilizados em aplicações robóticas.

### 3.2.2 Motores CC *Brushless*

O outro tipo principal de motores que têm sido utilizados em aplicações robóticas são os motores CC *Brushless*. Estes motores não necessitam de comutador e a parte eletrónica do motor está contida no estator. Ao contrário dos motores *brushed* os motores *brushless* não podem ser controlados apenas através da variação da tensão de alimentação. Devido à inexistência de escovas, estes necessitam de um circuito de controlo para a mudança de polaridade.

Tal como acontece com os *brushed*, este tipo de motores também possui um rotor e um estator. Contudo, neste caso, o rotor contém os ímanes permanentes e o estator as bobines eletromagnéticas. Para uma configuração trifásica, o número de estatores deve ser sempre múltiplo de três [37]. De forma a evitar o equilíbrio magnético, que resultaria na impossibilidade do motor iniciar o seu movimento, o número de polos dos ímanes permanentes não pode ser múltiplo do número de estatores.

Como não possuem comutador, os motores *brushless* necessitam de eletrónica dedicada para o controlo de velocidade. Um motor *brushless* move-se através da aplicação de tensão em duas das suas três fases, fazendo com que os estatores associados a cada uma das fases criem um campo elétrico. Desta forma, os ímanes permanentes fazem com que o rotor se mova à medida que os seus polos vão sendo atraídos e repelidos pela ativação de bobines eletromagnéticas [35]. Esta mudança de fase ao nível eletrónico ocorre sucessivamente fazendo com que o rotor "persiga" o campo magnético criado pelos estatores ativos sequencialmente. Na Figura 3.3 está representado um esquema deste tipo de motores.

Figura 3.3: Motor CC *Brushless* [38].

Destas características resultam diferentes vantagens e desvantagens na sua utilização. Uma das vantagens está relacionada com a eficiência, pois não existe queda de tensão nas escovas e durante a comutação. O facto de não possuir escovas, leva a que a fricção interna seja mais reduzida podendo-se obter velocidades mais elevadas. Outro benefício deste tipo de motores está relacionado com o facto de apresentar uma menor atenuação de ruídos eletromagnéticos.

Contudo, os motores CC *brushless* também apresentam as suas desvantagens. Uma das principais está associada ao controlo, pois para além de exigir controladores eletrónicos mais complexos, o seu custo pode ser tão ou mais elevado quanto os próprios motores.

### 3.3 Caixas Redutoras (*Gearboxes*)

Para utilização de motores CC em projetos de robótica torna-se necessária a redução da sua velocidade e, ao mesmo tempo, o aumento do seu torque<sup>3</sup>. Isto é feito através do acoplamento aos motores de um sistema mecânico que possa realizar essas funções - a caixa redutora.

<sup>3</sup>**Torque ou Binário** - Tendência de uma força para rodar um objeto em torno de um eixo. [39]

O propósito das caixas redutoras é atuar como um componente de transmissão de energia. Como tal, os dois fatores mais importantes no seu eixo de saída são a velocidade (em RPM - Rotações por Minuto) e a porção de trabalho que realiza por quantidade de torque produzido. As *gearboxes* são, normalmente, utilizadas para absorver a potência do motor, reduzir a sua velocidade e ampliar o torque [40].

As caixas redutoras são compostas por conjuntos de engrenagens que fornecem uma vantagem mecânica que multiplica o torque do lado da entrada para o seu eixo de saída. Esta vantagem mecânica é uma relação, chamada de *gearbox ratio* que é o número utilizado para determinar a multiplicação do torque da entrada para a saída. No caso em que a relação da caixa redutora seja de 30:1, por exemplo, o torque de saída é 30 vezes mais elevado do que o de entrada. Esta quantificação do aumento do torque é um valor teórico, pois na prática é necessário considerar pequenas perdas que ocorrem devido ao atrito das partes mecânicas.

Os tipos de caixas redutoras são classificados consoante o tipo de engrenagens que possuem. As mais utilizadas em motores CC aplicados a projetos de robótica móvel são de dois tipos de configurações: engrenagem direta e engrenagem planetária.

O tipo de *gearbox* mais comum é o de engrenagem direta (Figura 3.4). Estas têm a configuração mais simples e mais barata de fabricar. As engrenagens simples são feitas com dentes retos montados num eixo paralelo, permitindo a redução da velocidade e aumento do torque, e vice-versa, ou a mudança de direção rotacional [41]. Geralmente são compostas por diferentes variedades de tamanho e relações entre engrenagens, de modo a atender a requisitos específicos de torque e velocidade.



Figura 3.4: Engrenagem Direta [42].

As *gearboxes* planetárias (na Figura 3.5), têm esta nomenclatura devido à sua semelhança com o sistema solar. Na sua composição incluem uma engrenagem solar, uma engrenagem anelar e engrenagens planetárias. A engrenagem solar é o componente principal fixo no centro da caixa redutora. A anelar, como o próprio nome indica, é um anel externo com os dentes da engrenagem voltados para o interior. As engrenagens planetárias rodam em torno da solar e engrenam com esta assim como com a anelar.

Este tipo de caixa redutora é mais complexo e o seu custo de fabrico é mais elevado. Contudo, proporcionam uma grande transmissão de torque, alta eficiência e baixo ruído. Outra das suas vantagens é a capacidade de suportar reduções muito elevadas.

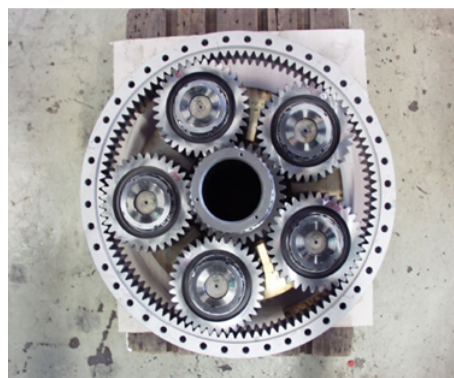


Figura 3.5: Engrenagem Planetária [41].

### 3.3.1 Escolha dos Motores

No desenvolvimento das bases omnidirecionais foram utilizados dois tipos diferentes de motores CC *brushed* com caixas redutoras acopladas. A escolha pelos motores CC *brushed*, em detrimento dos *brushless*, teve como principal fator o seu custo de aquisição. Para a escolha dos motores, outros fatores importantes relacionados com as especificações dos motores CC, foram considerados, os quais se apresentam de seguida.

#### Tensão Nominal (*Nominal Voltage*)

É a tensão que corresponde à maior eficiência do motor. A tensão de alimentação do sistema deve coincidir com a tensão nominal dos motores, de modo a poder tirar partido de toda a eficiência do motor. Se o motor for operado fora da sua tensão nominal, a eficiência diminui, requerendo muitas vezes corrente adicional, o que gera mais calor e diminui a vida útil do motor. Para além da tensão nominal, os motores CC também possuem um intervalo de tensão de operação, fora do qual os fabricantes não sugerem o seu funcionamento.

#### Potência Nominal (*Power Rating*)

A potência é a relação entre a corrente ( $I$ ) e a tensão ( $V$ ), estabelecida pela equação  $P = V \times I$ . Caso a potência do motor não esteja documentada nas suas especificações, esta pode ser calculada, ou pelo menos aproximada. Para calcular a potência nominal são utilizados os valores da corrente sem carga e a tensão nominal. A potência máxima do motor (que só deve ser utilizada por um curto período de tempo) pode ser aproximada multiplicando a corrente de corte (*Stall Current*) e a tensão nominal.

#### Velocidade Sem Carga (*No Load RPM*)

É a velocidade máxima (angular) a que o eixo do motor vai rodar assumindo

que não tem nada acoplado a ele. Se o motor tiver uma caixa redutora e a sua velocidade não estiver especificada, a velocidade sem carga é o valor de velocidade depois da redução. As RPM do motor são proporcionais à tensão de entrada. “Sem Carga” significa que o motor não encontra nenhuma resistência (nenhuma roda acoplada ou algo que contrarie o movimento). Normalmente a velocidade sem carga está associada à tensão nominal.

#### **Binário de Corte (*Stall Torque*)**

É o binário máximo que um motor consegue fornecer sem o eixo conseguir rodar mais. É importante mencionar que a maioria dos motores sofrerão danos irreparáveis se forem submetidos a condições de corte mais do que poucos segundos. Ao escolher um motor, deve-se ter em conta que este não esteja sujeito a mais do que  $1/3$  ou  $1/4$  do binário de corte, aproximadamente.

Como tal, para aplicação na base de rodas *mecanum* foram adquiridos, com o restante kit, o conjunto de motor CC e caixa redutora planetária IG52-04 (Figura 3.6), da *Shayang Ye Industrial Co., Ltd.*



Figura 3.6: *Gearmotor* IG52-04.

Este motor apresenta as seguintes especificações:

- Tensão de alimentação: 24V CC;
- *Reduction Ratio*: 1:26;
- Velocidade nominal: 136 RPM;
- Binário nominal: 23 kgf-cm;
- Corrente sem carga: < 700 mA;
- Potência: 48,6 W.

Para a base omnidirecional com rodas universais, pelo facto da empresa já possuir dois destes motores e com vista à sua reutilização, adquiriu-se o conjunto *Planetary Gearbox* e motor CC PK42BB-EC035.120 (Figura 3.7), da marca italiana *Transtecno*.

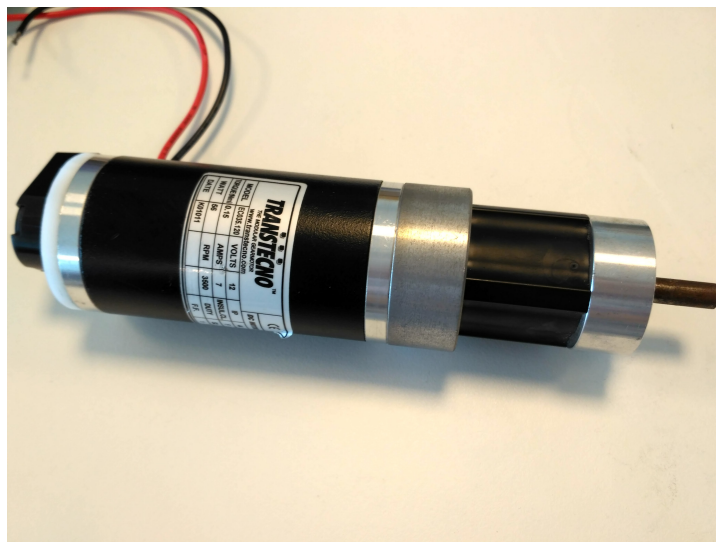


Figura 3.7: Gearmotor PK42BB-EC035.120

Segundo o seu *datasheet*, este motor apresenta as seguintes especificações:

- Tensão de alimentação: 12V CC;
- *Reduction Ratio*: 1:25;
- Velocidade nominal: 140 RPM;
- Binário nominal: 0,15 Nm;
- Corrente sem carga:  $< 700\text{ mA}$ ;
- Potência: 56 W.

No Anexo A.1 encontram-se os *datasheets* dos motores utilizados, com o descritivo completo das suas características.

### 3.4 Encoders

Em robótica móvel, os *encoders* são sensores propriocetivos<sup>4</sup> utilizados para determinar a posição, velocidade e direção de um eixo de um motor ou outro movimento mecânico. Estes equipamentos são utilizados para converter movimentos rotativos ou deslocamentos lineares em impulsos elétricos, fornecendo uma quantidade fixa de impulsos por volta de um eixo ou por unidade de deslocamento linear.

Tendo em conta o tipo de *output* que fornecem, os *encoders* podem ser classificados como absolutos ou incrementais.

---

<sup>4</sup>**Sensor Propriocetivo** - Sensor que fornece informação interna relativa ao sistema robótico (*i.e.* velocidade das rodas, nível da bateria, etc) [43].

### 3.4.1 Encoders Absolutos

Os *encoders* absolutos fornecem a posição angular de um eixo de rotação. O seu princípio de funcionamento tem por base um conjunto de dois discos opacos, um fixo e outro móvel (que acompanha a rotação do eixo), uma fonte de luz e um conjunto de foto-sensores para captar essa luz. O disco fixo possui uma ranhura radial e o disco móvel apresenta padrões de ranhuras dispostas de forma a que a cada padrão corresponda uma posição angular diferente. À medida que o disco móvel roda, em conjunto com o eixo, os foto-sensores vão captando diferentes padrões de luz. Como a cada padrão de luz corresponde uma posição angular, a determinação da posição do eixo à qual o *encoder* está associado é imediata. O sinal de saída deste tipo de *encoders* consiste em valores binários discretos codificados, cujo tamanho pode ir desde 4 a 16 *bits*. Na Figura 3.8 está representado um esquema da estrutura simplificada de um *encoder* absoluto.

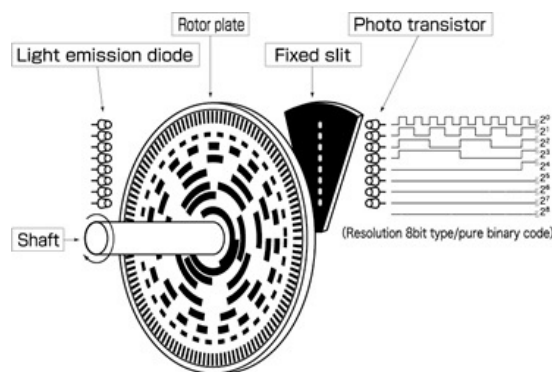


Figura 3.8: Estrutura de um *Encoder* Absoluto [44].

### 3.4.2 Encoders Incrementais

Os *encoders* incrementais apresentam um princípio de funcionamento parecido com os absolutos, mas fornecem um sinal de saída periódico, cujo

período é inversamente proporcional à velocidade de rotação do eixo. Às velocidades de rotação elevadas o *encoder* fornece um sinal com uma frequência elevada e reduzido período. Neste tipo de *encoders*, as ranhuras do disco móvel não estão associadas a nenhuma posição angular, mas são dispostas de modo a fornecerem à saída um sinal periódico. Através da medição do período deste sinal é possível identificar a velocidade de rotação do eixo associado. Na Figura 3.9 pode-se observar um exemplo da estrutura deste tipo de *encoders*.

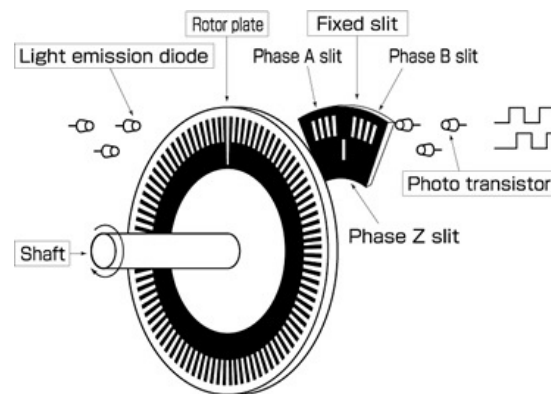


Figura 3.9: Estrutura de um *Encoder* Incremental [44].

Também em relação à tecnologia utilizada, os *encoders* incrementais podem-se dividir em *encoders* magnéticos (ou de *hall-effect*) e óticos.

### 3.4.3 *Encoders* Magnéticos

Os *encoders* magnéticos utilizam uma combinação de ímãs permanentes e sensores magnéticos para detetar movimento e posição [45]. A sua construção típica utiliza ímãs colocados em redor de um disco rotativo, acoplado a um eixo, e posicionado de forma a que um sensor detete mudanças no campo magnético, à medida que os polos alternados do ímã passem por este sensor. Na sua configuração mais simples, este tipo de *encoder*, teria

um único íman, com os polos em extremos opostos do disco rotativo e um único sensor magnético. O *output* deste dispositivo seria uma onda sinusoidal com uma frequência igual à velocidade de rotação do eixo. Através da utilização de um segundo sensor magnético, desfasado 90° em relação ao primeiro (como representado na Figura 3.10), torna-se possível detetar, para além da direção de rotação, também a posição absoluta do eixo, através dos sinais sinusoidais desfasados intercalados que se obtêm. A sua resolução é tão maior, quanto maior for também, a quantidade de polos magnéticos e sensores, em torno do disco rotacional. Num *encoder* magnético composto por 128 polos e 4 sensores, conseguem-se obter 1024 posições (resolução de 10 *bits*), numa volta completa do eixo.

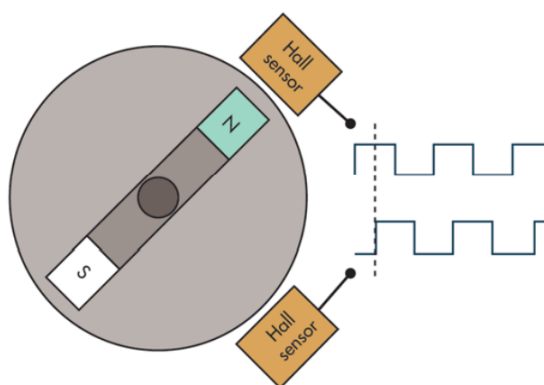


Figura 3.10: Funcionamento de um *Encoder* Magnético [45].

#### 3.4.4 Encoders Óticos

Os *encoders* incrementais óticos são dos dispositivos mais utilizados para medição da velocidade angular e posição num motor ou eixo de uma roda, em robótica móvel. Um *encoder* ótico é basicamente um mecanismo composto por um disco de plástico ou vidro, com um padrão de áreas transparentes e opacas intercaladas, que são detetadas à medida que o disco gira entre uma fonte de luz e um foto-sensor. Da mesma forma que os *encoders* magnéticos,

a sua resolução é tão maior, quantas mais divisões opacas e transparentes tiver o disco de rotação.

Na robótica, a onda sinusoidal resultante é transformada numa onda quadrada discreta, utilizando-se um *threshold* para escolha entre os estados escuros e claros. A resolução é medida em Ciclos Por Revolução (CPR do inglês *Cycles per Revolution*), valor este que é tão maior quanto mais preciso for o *encoder*. Um *encoder* típico em robótica móvel pode ter cerca de 2000 CPR, enquanto que um *encoder* ótico industrial pode ter uma resolução de 10000 CPR [10].

O tipo de *encoder* mais usado em robótica móvel é o *encoder* de quadratura. Neste tipo de *encoders* é utilizado um segundo conjunto de detecção de iluminação desfasado 90° em relação ao original. O conjunto de ondas quadradas A e B resultantes (representadas na Figura 3.11), permitem a obtenção de mais informação do que apenas uma única. A ordem pela qual cada onda quadrada apresenta o primeiro flanco ascendente do impulso, permite identificar a direção de rotação do eixo do motor. Para além disto, os quatro estados detetáveis diferentes melhoram a resolução por um fator de 4, ou seja, um *encoder* de 2000 CPR em quadratura efetua 8000 contagens .

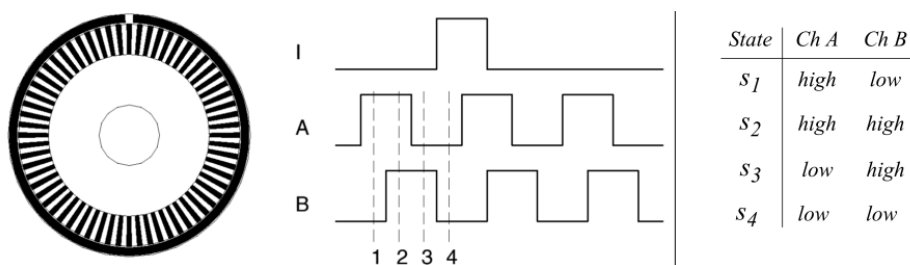


Figura 3.11: *Encoder* Ótico de Quadratura [10].

Alguns *encoders* deste tipo possuem no disco uma pista externa com um único *slot* que gera um impulso de referência denominado *index* ou *Z-pulse*. [46] Este impulso é gerado apenas uma vez a cada volta completa e pode ser utilizado para reiniciar um contador ou para determinação precisa de uma posição de referência.

### 3.4.5 Encoders Utilizados

No desenvolvimento das plataformas omnidirecionais, utilizaram-se dois tipos *encoders* incrementais, magnéticos e óticos.

Como sendo parte do kit adquirido, os *encoders* que foram aplicados nos motores da base com rodas *mecanum* foram *encoders* do tipo *Hall Effect* de dois canais. Devido à sua folha de especificações não apresentar informações relativas à sua resolução nem ao número de polos, foi necessário efetuar o seu cálculo de forma experimental.

Para tal, obteve-se o valor medido pelo *encoder* durante uma volta completa da roda (680,53 CPR). Pelo facto de entre a roda e o *gearmotor* existir um sistema de engrenagens guiados por uma corrente, a relação entre os dentes de cada uma das engrenagens interfere neste cálculo, ou seja, como o número de dentes da engrenagem à qual esta acoplada a roda (22 dentes) é superior ao número de dentes da engrenagem do motor (15 dentes), a cada volta completa da roda, o motor tem que dar uma volta completa mais  $\frac{15}{22}$  de volta. Assim, o valor medido pelo *encoder* corresponde, a uma e volta e  $\frac{15}{22}$  do motor e não apenas uma volta completa. O valor da resolução do *encoder* utilizado é obtido então a partir da equação seguinte.

$$Imp_r = 680,53 \times \frac{15}{22} = 463,99 \approx 464 \quad (\text{CPR}) \quad (3.1)$$

No projeto da base omnidirecional com rodas universais, foram utilizados quatro *encoders* incrementais óticos de três canais AEDM-5810, representado na figura 3.12, do fabricante *Avago Technologies*, acoplados aos quatros motores.



Figura 3.12: *Encoder* AEDM-5810 [47].

Este *encoder* é alimentado por intervalo de tensão de 4,5 V a 5,5 V e consome um corrente típica de 25 mA. Possui uma resolução de 5000 CPR e três pinos de saída, identificados como canal A, canal B e canal I (*index*) e dois como GND e VCC para alimentação, como se pode verificar na Figura 3.13.

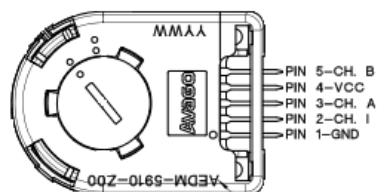


Figura 3.13: *Pinout* do *Encoder* AEDM-5810 [47].

Internamente o *encoder* possui um LED (*Light-Emitting Diode*) próximo a uma lente de policarbonato, que emite um feixe de luz na direção de fotodíodos. Entre estes existe um disco com um padrão de furos, que roda em conjunto com o eixo. O *output* deste *encoder* é composto por dois sinais de

onda quadrada, em quadratura, correspondentes aos canais A e B e um terceiro sinal, também de onda quadrada, que gera um impulso por cada volta completa do eixo (*index*). O conjunto de sinais de *output* está representado na Figura 3.14.

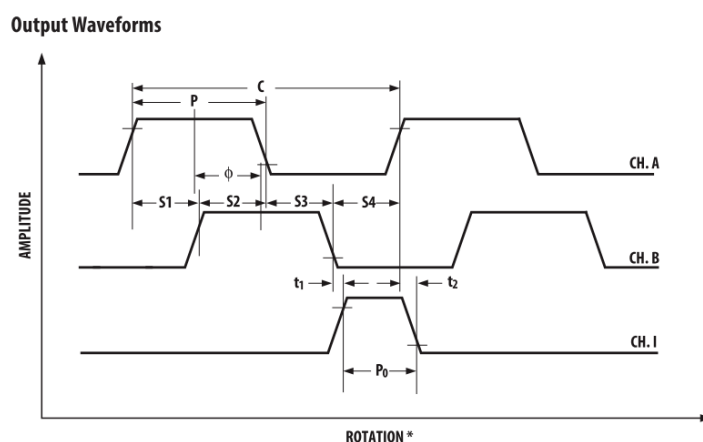


Figura 3.14: *Output* do Encoder AEDM-5810 [47].

O facto de possuir saída em quadratura, faz com que gere uma sequência de quatro impulsos diferentes a cada conjunto de períodos dos sinais dos canais A e B. A ordem pela qual é detetado o primeiro impulso (A ou B), permite identificar o sentido de rotação do motor.

Na descodificação dos dados dos *encoders*, são tidas em conta as diferentes posições de saída. Desta forma, tendo o *encoder* 5000 CPR e quatro impulsos por ciclo, vai enviar para o controlador 20000 impulsos por rotação completa, valor este que será incrementado ou decrementado, dependendo do sentido de rotação do motor.

### 3.5 Sistema de Controlo de Motores

Os sistemas de controlo de motores, de um modo geral, recebem sinais provenientes de microprocessadores, emissores de rádio controlo, ou computadores, e convertem-nos em tensão variável de alta corrente, para fornecer aos motores, atuando como interface entre os dois. A velocidade dos motores está relacionada de forma proporcional com a tensão aplicada. Para que os motores girem a metade da sua velocidade, por exemplo, necessitam de uma tensão de entrada com 50% da sua tensão operacional máxima. O sistema de controlo de motores satisfaz esta necessidade e faz a ponte entre o código de controlo de velocidade e as entradas do motor.

Os sistemas de controlo mais sofisticados, permitem a aplicação de controlo de *feedback* de baixo nível. Através da medição da diferença entre o comando que foi enviado para o motor e o movimento real que o motor efetuou, ajustam o sinal de comando, de forma a melhorar a concordância entre os dois. Se, por exemplo, for enviado um comando para o motor atingir a velocidade de 50%, mas na realidade apenas atingiu a velocidade de 49%, é aplicado um valor um pouco mais elevado no comando seguinte e a velocidade é medida novamente, de forma a verificar se o motor gira a 50% ou são necessários mais ajustes. Para efetuar esta tarefa, os controladores precisam de receber *feedback* dos motores, daí a necessidade de integração de sensores (como *encoders*) que meçam o deslocamento de rotação dos motores.

Para a escolha do sistema de controlo adequado existem algumas características importantes que se devem considerar:

- Circuito de condução adicional para converter sinais de controlo em sinais de potência (*p.e.* Ponte-H);
- Potência necessária para os motores;

- Disponibilidade e facilidade dos pinos de *output* da placa de controlo;
- Circuito de segurança para evitar picos de tensão, danos nos motores, e proteção térmica.

Atualmente existem no mercado diferentes tipos de soluções integradas que satisfazem grande parte das funcionalidades deste tipo de sistemas de controlo, recorrendo na sua maioria a controlo por PWM (*Pulse Width Modulation*) e tendo como base de circuito de condução pontes-H. De forma a compreender os métodos que estes sistemas utilizam para efetuar o controlo dos motores, torna-se relevante entender estes dois conceitos relacionados com os mesmos.

### 3.5.1 *Pulse Width Modulation* (PWM)

Uma opção para controlar a velocidade de operação dos motores é a utilização de *Pulse Width Modulation*. O PWM é um tipo de controlo comumente utilizado para diversos tipos de dispositivos, incluindo a maioria dos motores controlados por tensão. O controlo por PWM utiliza uma tensão de entrada constante (como a alimentação do sistema de controlo) e fá-la variar consoante o nível de tensão “considerado” pelo motor, permitindo assim que o motor funcione a diferentes velocidades. O termo “considerado” é utilizado, porque o PWM funciona por variações rápidas, “ligando” e “desligando” a tensão de alimentação para o motor. Isto faz com que o motor se comporte como se tivesse uma tensão constante igual à média entre, quanto tempo a tensão fornecida está *on* e *off*. Ao variar a percentagem de tempo que a tensão está realmente “ligada”, o sistema de controlo do motor consegue controlar a velocidade dos motores [48].

Se o controlador enviar, por exemplo, um sinal de entrada que está *on* tanto tempo quanto está *off*, a totalidade do sinal que é enviado será semelhante

ao representado pela onda quadrada da Figura 3.15 a). Como resultado, o motor receberá uma tensão de entrada com 50% da tensão constante “ligada”. Devido à natureza da onda quadrada, a parte do sinal que está *on* é muitas vezes referida como o envio de um sinal alto, da mesma forma que a parte *off* corresponde a um sinal baixo. A proporção entre o sinal baixo e alto pode alterar a tensão de entrada recebida pelo motor. Se o sinal alto for enviado 75% de cada período da onda quadrada (como representado na Figura 3.15 b)), o motor receberá uma tensão com 75% do valor de tensão de entrada do sistema de controlo. A percentagem de cada período durante a qual o sinal é alto, é reconhecida como *Duty Cycle*.

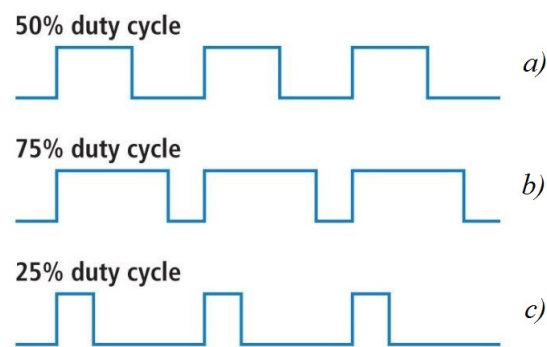


Figura 3.15: Sinais PWM.

O método de PWM funciona bem em controlo de motores pois, sendo estes de natureza mecânica, os seus tempos de resposta são muito mais lentos do que a maioria dos componentes eletrónicos. Isto significa que uma série de tensões de entrada, mudando rapidamente entre “ligado” e “desligado,” farão com que um motor as reconheça como uma única tensão constante aplicada, igual ao *Duty Cycle* do PWM multiplicado pela tensão máxima do sistema de controlo.

Os sistemas integrados que utilizam este método, geram os sinais PWM através de um microcontrolador, que posteriormente os enviam para o restante

circuito de controlo dos motores. Este circuito representa uma parte importante do controlo, pois recebe, tanto a tensão de alimentação como um segundo sinal de controlo, que é utilizado para definir o *Duty Cycle*. Um tipo de circuito de controlo bastante utilizado no controlo de motores CC é o circuito em ponte-H, mencionado na Secção 3.5.2.

### 3.5.2 Circuito de Ponte H

O controlo de motores CC *brushed*, é normalmente efetuado com recurso a uma ponte-H, que executa duas funções importantes em relação ao controlo de corrente.

A primeira função é de amplificação do sinal de comando, para garantir que está dentro da gama de operação do motor. O sinal de controlo é, geralmente, de baixa potência, enquanto que o motor requer uma tensão muito maior. De forma a traduzir o sinal de comando PWM de baixa potência, com um determinado *Duty Cycle*, numa tensão de entrada proporcional para os motores, é então utilizada a ponte-H [49].

A outra função das pontes-H é a de controlo da direção de rotação dos motores através da alteração do sentido da corrente que passa nos motores. Para melhor entendimento de como funciona este tipo circuito, é útil a análise do seu esquema, representado na Figura 3.16.

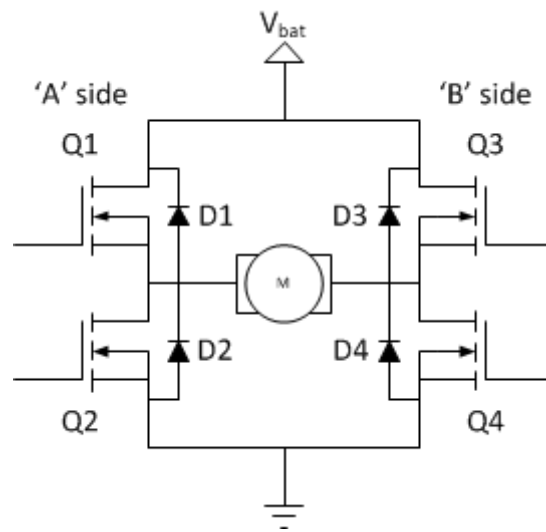


Figura 3.16: Esquema do Circuito de Ponte-H [49].

Uma ponte-H é composta por quatro *switches* (Q1-Q4), conectados aos terminais do motor dois a dois. Na maioria dos circuitos de controlo de motores, estes *switches* são transístores ou MOSFETs (*Metal Oxide Semiconductor Field Effect Transistor*), mas para melhor compreensão podem-se considerar como *switches*. Através do controlo de quais os *switches* estão abertos ou fechados, a polaridade da tensão aplicada no motor pode ser alterada.

Uma vez que a direção de rotação de um motor CC depende da polaridade tensão aplicada, utilizam-se pontes-H para efetuar esta alteração de polaridade, não precisando assim de inverter a tensão original de alimentação.

Quando os *switches* Q1 e Q4 estão fechados e Q2 e Q3 abertos, é aplicada uma tensão no motor da esquerda para a direita e o motor gira numa direção (Figura 3.17 a)). Quando Q1 e Q4 estão abertos e Q2 e Q3 fechados a polaridade é invertida e o motor gira na direção contrária (Figura 3.17 b)).

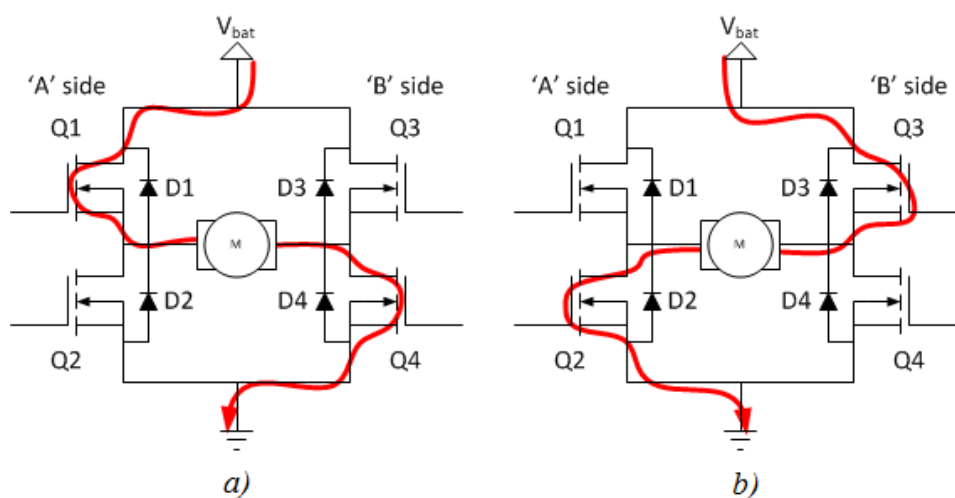


Figura 3.17: Esquema de Condução da Ponte-H [49].

Neste tipo de circuito, nunca se deve ativar Q1 e Q2 (ou Q3 e Q4) ao mesmo tempo, pois vai causar um curto circuito.

Através da junção destas duas funções da ponte-H, é claro perceber a viabilidade da sua aplicação para o controlo de velocidade e direção de motores CC. Se a corrente que flui, através dos *switches* fechados, tem o valor de tensão constante de alimentação e o sinal PWM de controlo é utilizado para abrir e fechar os *switches*, fazendo com que estes estejam fechados quando o sinal é alto (e vice versa), então o sinal recebido pelo motor será uma onda quadrada com a amplitude da tensão de alimentação e com o *Duty Cycle* do controlo PWM. Da mesma forma, se o sinal PWM for aplicado ao outro conjunto de *switches*, a velocidade do motor é controlada na direção oposta.

A decisão de utilização de dispositivos integrados para o controlo dos motores em vez do seu desenvolvimento, assentou no facto dos primeiros serem mais confiáveis, robustos e eficientes, do que qualquer circuito que fosse desenvolvido.

Mediante as especificações dos motores e requisitos exigidos pelo projeto, foram considerados os controladores *RoboteQ* SCD2130 e *Sabertooth* 2x25.

### 3.5.3 *RoboteQ* SDC2130

O controlador *RoboteQ* SDC2130 foi projetado para converter comandos recebidos por R/C, *Joystick* analógico, modem sem fios, PC (via RS232 ou USB), ou um microcontrolador, em corrente e tensão de saída para controlar um ou dois motores CC. Para aplicações em robôs móveis, os dois canais dos motores podem ser operados de forma independente ou em simultâneo, definindo a rotação do veículo, através da coordenação do movimento de cada motor.

Este controlador possui um microcontrolador de 32 *bits*, assim como entradas para *encoders* em quadratura, o que lhe permite a execução de algoritmos de controlo de movimento em *open* e *closed loop*, tanto a nível de velocidade como posição [50]. Para cada um dos canais, o *RoboteQ* possui os seguintes modos de controlo de movimento:

- *Open Loop Speed Mode* - Neste modo, o controlador fornece uma quantidade de energia proporcional à informação de comando. A velocidade atual do motor não é medida;
- *Closed Loop Speed Mode* - Neste modo, é utilizado um *encoder* ótico ou um tacómetro analógico para medir a velocidade atual do motor. Se a velocidade mudar devido a alterações na carga, o controlador compensa automaticamente a saída de alimentação. Este é o modo mais utilizado em controlo de precisão de motores;
- *Closed Loop Position Mode* - Neste modo, o motor é acoplado a um potenciómetro ou *encoder* utilizados para comparar a posição angular

do eixo com a posição desejada;

- *Torque Mode* - Neste modo, o controlador ajusta o *output* de modo a alcançar e manter um valor de corrente definido.

O *RoboteQ* SDC2130, representado na Figura 3.18, também possui um conjunto de entradas e saídas, analógicas e digitais, que podem ser configuradas para receber comandos ou outro tipo de *feedback*, para integrar interruptores de fim de curso, entre outros.

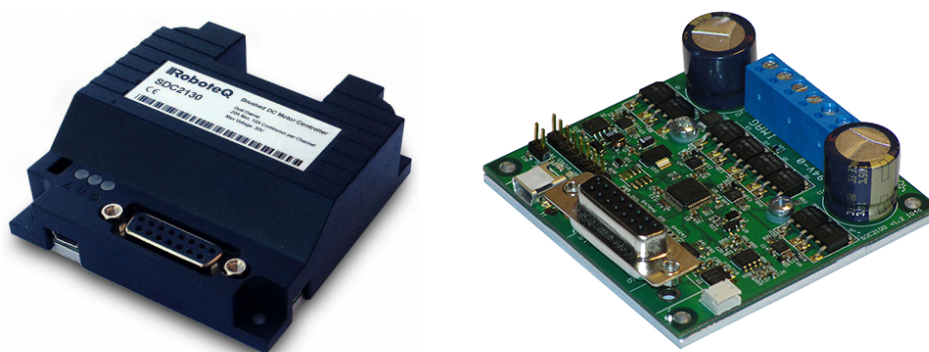


Figura 3.18: *RoboteQ* SDC2130 [50].

A sua alimentação vai desde os 7 V até um máximo de 30 V e fornece corrente de até 20 A por canal, a dois motores CC, sendo que controlando apenas um motor, pode ser configurado para fornecer até no máximo 40 A.

Possui interface CAN (*Controller Area Network*) com velocidades até 1 Mbit/s e suporta CANopen assim como outros dois protocolos CAN simplificados. Efetua controlo de avanço e retrocesso em cada canal e suporta controlo regenerativo. Este controlador tem a possibilidade de configuração dos ganhos PID (Proporcional, Integral e Derivativo) de forma separada para cada canal e possui sensores para proteção de sobretensão, sub-tensão e de temperatura.

Através do *software* dedicado *Roborun+* (disponibilizado de forma gratuita pela *RoboteQ*) é possível configurar o controlador, enviar comandos e receber *feedback* dos motores, *encoders* e do estado do próprio controlador, em tempo real e de uma forma intuitiva. Por intermédio do *Roborun+*, representado na Figura 3.19, existe a possibilidade de escrita de programas que são permanentemente guardados e executados a partir da memória *flash* do controlador.

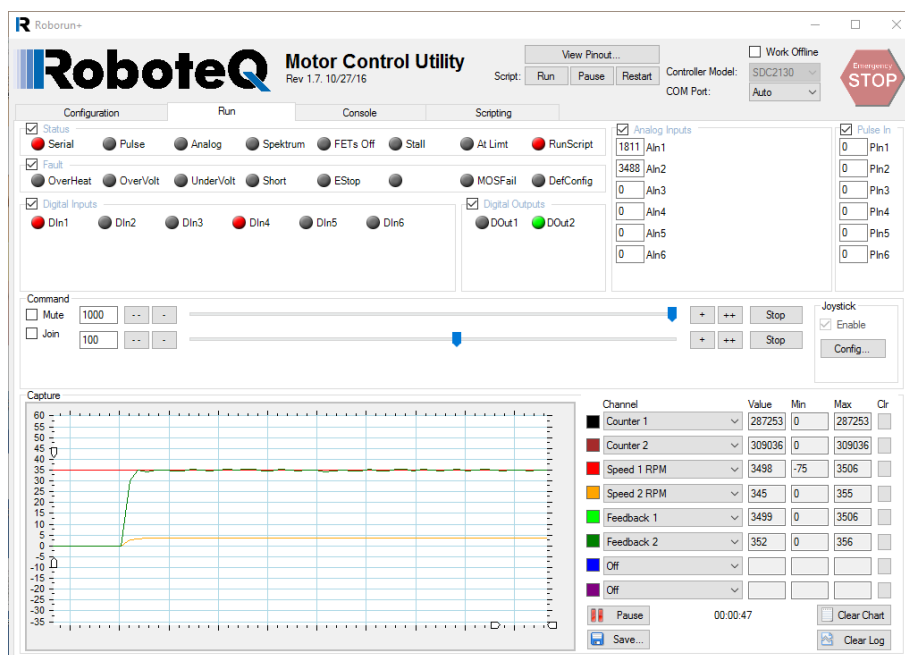


Figura 3.19: Interface do *Software Roborun+*.

Estes programas são escritos numa linguagem de programação integrada, baseada em *script* denominada *MicroBasic* [51]. Estes *scripts* têm uma sintaxe muito literal, muito parecido com o inglês escrito e possuem suporte para programação estruturada, permitindo a escrita desde programas mais simples até aos mais rebuscados e complexos. Para facilitar a escrita e o tamanho do código, esta linguagem também permite a utilização de atalhos para operadores ( $++$ ,  $!=$ ,  $+=$ , entre outros), herdados da linguagem C.

### 3.5.4 *Sabertooth 2x25*

O *Sabertooth 2x25* é um dos mais versáteis controladores de motores do mercado, adequado para robôs de elevada potência. Este controlador é capaz de fornecer 25 A a dois motores CC, com correntes de pico até 50 A por motor, durante um curto intervalo de tempo [52].

O *Sabertooth*, que se observa na Figura 3.20, apresenta modos mistos projetados especificamente para robôs diferenciais, mas também possui opções independentes para todos os modos de controlo. Permite o controlo de 2 motores, mesmo que sejam diferentes, desde que ambos estejam dentro dos limites operacionais do *Sabertooth*. É possível a sua operação através de quatro modos: por Rádio Controlo, entrada analógica e dois modos por entrada série.

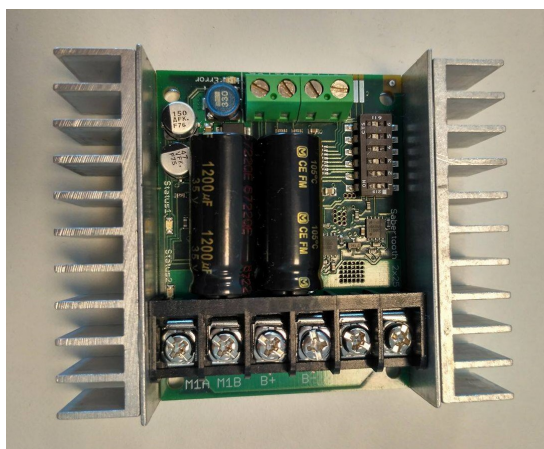


Figura 3.20: *Sabertooth 2x25 V2*.

O seu intervalo de alimentação vai dos 6 V aos 30 V e através de sensores de temperatura e picos de correntes, tem a capacidade de se proteger de falhas por sobreaquecimento, sobrecargas ou curtos-circuitos. Também possui controlo regenerativo síncrono, ou seja, o controlador retorna a potência para a bateria sempre que haja uma desaceleração ou inversão do sentido de rota-

ção do motor. Isto pode levar a enormes melhorias no tempo de execução de sistemas que parem ou invertam o sentido frequentemente, como um robô de posicionamento ou um veículo conduzido em terreno irregular. Este esquema de controlo também economiza bateria, devolvendo a energia indutiva armazenada nos enrolamentos do motor à bateria a cada ciclo de comutação, em vez de o transformar em calor nos enrolamentos. Isto torna o controlo de aceleração parcial muito eficiente. A frequência de comutação é de  $32\text{ kHz}$ , valor acima do valor audível pelo ser humano, não apresentando qualquer ruído mesmo a níveis baixos de potência. Na Figura 3.21 apresenta-se um esquema representativo das ligações do *Sabertooth 2x25*.

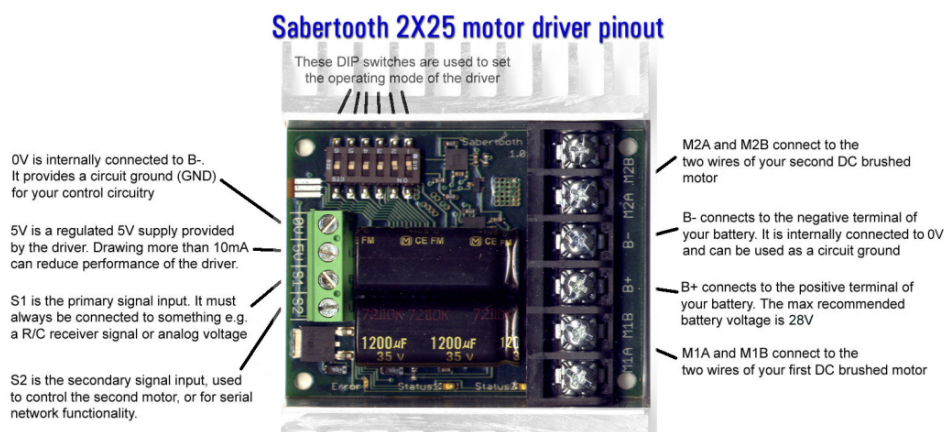


Figura 3.21: Ligações do Controlador Sabertooth 2x25 [53].

Os diferentes modos de operação são selecionados mediante os seis interruptores DIP que o controlador possui. As diferentes combinações das posições dos interruptores 1 e 2 correspondem aos quatro modos de operação, enquanto que os interruptores 4, 5 e 6 estabelecem diferentes opções em cada modo de controlo. A posição do interruptor 3 está relacionada com o tipo de alimentação do sistema. Os modos de operação disponíveis são:

- Modo 1 - Entrada Analógica: neste modo são utilizadas uma ou duas

entradas analógicas para definir a velocidade e direção do motor, sendo que o intervalo válido de valores de entradas é entre os 0 V e os 5 V;

- Modo 2 – Entrada R/C: neste modo utilizam-se dois canais *standard* R/C e, da mesma forma que no modo 1, é possível controlar a velocidade e direção do motor. Este modo possui uma configuração opcional de *timeout*, que quando está ativo desativa o controlador em casos de perda de sinal, como medida de segurança;
- Modo 3 – Série Simplificado: este modo utiliza dados do tipo série RS-232 do nível TTL para definir a velocidade e direção do motor. Este modo é, geralmente, utilizado como interface entre o *Sabertooth* e um computador ou um microcontrolador. Para se utilizar com a porta série de um computador é necessário um conversor de nível (como o MAX232) ou um conversor série USB to TTL. A taxa de transmissão (*Baud Rate*) é definida através das posições dos interruptores DIP e os comandos transmitidos e recebidos são *single-byte*. Também existe um modo *Slave Select* que permite o uso de vários *Sabertooth* a partir de uma única porta série.
- Modo 4 – Série Empacotado: este modo utiliza o mesmo tipo de dados do modo 3 no entanto a transmissão dos dados é efetuada em pacotes. Existe um formato de pacote curto composto por um *byte* de endereço, um *byte* de comando, um *byte* de dados e um *checksum* de 7 bits. Por defeito a taxa de transmissão é 9600 *bps* e os bytes de endereço são definidos através dos interruptores DIP. É possível ligar até 8 *Sabertooth* em série, tornando este modo o mais indicado para conectar vários controladores a um PC.

### 3.5.5 *Kangaroo x2 Motion Controller*

O *Kangaroo x2*, representado na Figura 3.22, é um módulo de extensão para controladores da *Dimension Engineering*, que lhes confere capacidade de controlo de movimento e receção de *feedback*. Tem a capacidade de leitura de um ou dois *encoders* de quadratura (ou potenciómetros), de modo a controlar a velocidade e/ou posição de um ou dois motores. Este módulo, possui a capacidade de auto ajuste do controlo PID e admite o controlo por radio controlo, sinais analógicos, ou entrada série. Também possui suporte para interruptores de fim de curso e paragens mecânicas.

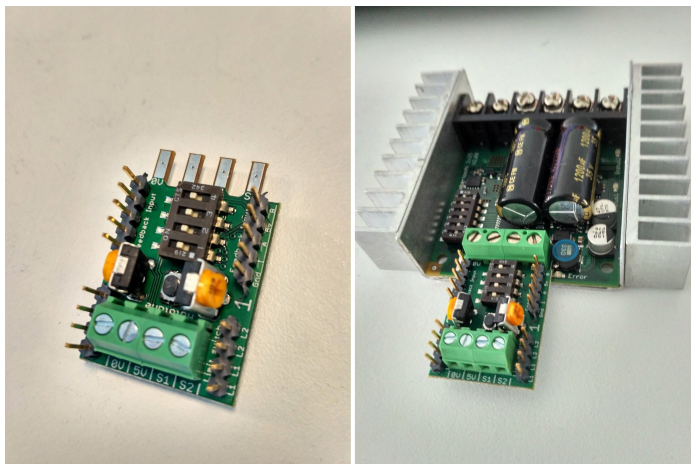


Figura 3.22: *Kangaroo x2 Motion Controller*.

O *Kangaroo x2* deteta automaticamente a direção, ganho e parâmetros do sistema do motor conectado e a velocidade máxima pode ser ajustada por intermédio de dois potenciómetros incorporados na placa. Este e outros parâmetros podem ser alterados através do software “DEScribe”, disponibilizado pela *Dimension Engineering*, compatível com uma larga gama de controladores da marca [54]. Na Figura 3.23 observa-se um esquema das ligações disponíveis deste módulo de controlo.

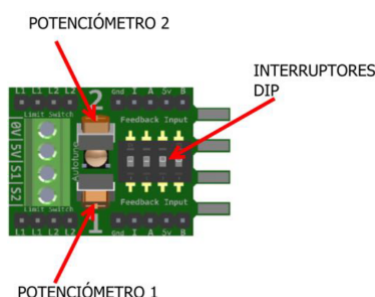


Figura 3.23: Esquema do módulo *Kangaroo x2* [33].

As ligações para os *encoders* encontram-se representadas nas laterais e são compostas pelos pinos Gnd, I, A, B e 5V. Os interruptores de fim de curso são conectados nos diferentes pinos L1 e L2. As conexões 0V, 5V, S1 e S2 têm as mesmas funcionalidades que os correspondentes no controlador *Sabertooth*. Ao centro possui um botão de auto ajuste do controlo PID (*Autotune*) e, analogamente ao controlador *Sabertooth*, também possui interruptores DIP cujas posições permitem a escolha do modo operação deste módulo. A Figura 3.24 apresenta os diferentes modos de operação do *Kangaroo x2* consoante as posições ON/OFF dos interruptores DIP.

OFF setting		ON setting
<b>1 off: Analog input.</b> Connect 0-5V analog signals to the S1 and S2 inputs.		<b>1 on: Digital input</b> Connect TTL serial, TX to S1 and RX to S2, or R/C servo signals to S1 and S2
<b>2 off: Analog feedback</b> Connect a 0-5V signal to Feedback Input A		<b>2 on: Quadrature feedback</b> Connect an encoder to Feedback Inputs A and B
<b>3 off: Velocity control</b> Motor speed and direction are controlled by the input signal		<b>3 on: Position control</b> Motor position is controlled by the input signal
<b>4 off: Mixed mode</b> The outputs are mixed together for differential drive mobile robots		<b>4 on: Independent mode</b> The outputs are independent. S1 controls motor 1 and S2 controls motor 2.

Figura 3.24: Modos de Operação do *Kangaroo x2* consoante interruptores DIP [54].

### 3.5.6 Comparativo dos Controladores

Para a escolha dos controladores dos motores foram tidas em conta as especificações dos motores, previamente selecionados e, mediante os requisitos do projeto, os fatores preponderantes foram: o número de canais por controlador (pelo menos dois), a corrente por canal, possuir interface para *encoder*, interface série e o preço.

Na Tabela 3.1 encontra-se um comparativo das características dos dois controladores de motores estudados, mencionados nas Secções 3.5.3 e 3.5.4.

Tabela 3.1: Tabela comparativa dos controladores *RoboteQ* SDC2130 e *Sabertooth* 2x25 V2.

	<b>Sabertooth 2x25 V2</b>	<b>RoboteQ SDC2130</b>
Tensão de entrada	6-30V	7-30 V
Tensão máxima de entrada	33,6V	30V
Nº. de Canais para motores	2	2
Regeneração síncrona	Sim	Sim
Corrente Max. por canal	50 A	20 A
Corrente contínua por canal	25 A	10 A
Proteção de temperatura	Sim	Sim
Proteção de sobretensão	Sim	Sim
Alimentação de saída	5V	5V
Entrada R/C	Sim	Sim
Interface Série	Sim	Sim
Interface Analógica	Sim	Sim
Interface USB	Não	Sim
Interface para <i>Encoder</i>	Sim (através de módulo adicional)	Sim
Suporte CAN	Não	Sim
Dimensões	65x80x21 mm	70x70x20 mm
Peso	90 g	100 g
Preço	\$124,99	\$175,00

Tendo em conta as características dos dois controladores estudados e, de acordo com as necessidades do projeto, ambos os controladores cumpriam os requisitos necessários, sendo que o custo de aquisição do *Sabertooth* é inferior ao do *RoboteQ*.

Desta forma, devido ao seu preço ser mais reduzido e pelo facto do controlador atualmente utilizado no wiiGO ser o *RoboteQ* SCD2130, foram testados dois *Sabertooth* 2x25 com módulos *Kangaroo x2*, por forma a avaliar a via-

bilidade da sua aplicação.

Após a realização de vários testes com os motores e *encoders*, foram detetados alguns problemas de comunicação série, assim como de configuração de parâmetros dos motores e do *feedback* recebido dos *encoders*. Mesmo depois de alguns contactos com o fabricante, alguns destes problemas encontrados não foram solucionados, levando a que se tomasse a decisão de não utilizar estes controladores, de forma a não atrasar o desenvolvimento, e optar pela escolha dos *RoboteQ* SDC230 (que a empresa já possuía).

### 3.6 Escolha das Rodas

Como mencionado no início do presente capítulo, foram desenvolvidas duas plataformas omnidirecionais com duas configurações diferentes, as quais diferiam principalmente, para além do seu formato e características físicas, nas rodas omnidirecionais utilizadas.

Para uma das configurações, foram utilizadas (como parte do kit adquirido) quatro rodas omnidirecionais *mecanum* (Figura 3.25) de 8" (203 mm) de diâmetro e 78 mm de largura, compostas por doze rolos de poliuretano e estrutura de aço inoxidável. Cada uma das rodas tem uma capacidade de carga de 37,5 kg e têm um peso de 2,4 kg. Todas as restantes características funcionais e cinemáticas deste tipo de rodas foram mencionados no capítulo 2.



Figura 3.25: Rodas Omnidirecionais *Mecanum* Utilizadas.

Numa fase posterior, como será explicado no Capítulo 5, estas rodas foram substituídas por umas de configuração igual mas com os rolos de material diferente.

Para a outra configuração, sendo que a plataforma foi idealizada de raiz, foi

convencionado que se utilizariam rodas omnidirecionais universais. Desta forma, foi elaborado um estudo relacionado com este tipo de rodas omnidirecionais, no qual os fatores principais a considerar, para além de serem do tipo universal (com os rolos a  $90^\circ$ ), foram a capacidade de carga ( $\geq 100$  *kg*), o preço e o material dos rolos, de forma a minimizar ao máximo o deslizamento. No anexo B encontram-se representados os resultados obtidos do estudo das rodas omnidirecionais universais.

Após a análise dos resultados obtidos do estudo das rodas universais foram escolhidas as rodas omnidirecionais Transwheel FXA357 da *Kornylak* (Figura 3.26). Estas rodas apresentam 4" (101,6 *mm*) de diâmetro compostas por 16 rolos de prolipopileno revestido a borracha sintética, com a estrutura em plástico de elevada resistência. Têm uma capacidade de carga 100 *pounds* (45,36 *kg*), por roda, e têm um peso de 0,312 *kg*. A sua escolha em detrimento das outras estudadas, relacionou-se, para além do cumprimento dos requisitos impostos inicialmente, com o facto de estarem disponíveis na empresa para utilização, sem necessidade de serem adquiridas.

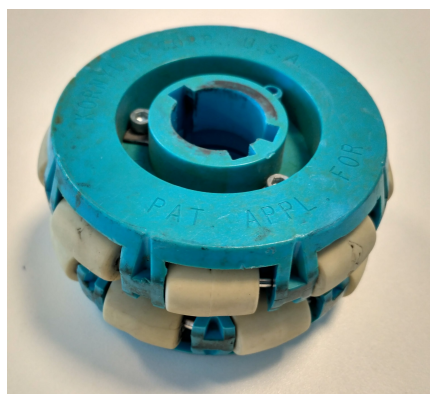


Figura 3.26: Rodas Omnidirecionais Universais Utilizadas.

### 3.7 Desenvolvimento da Estrutura

Para o desenvolvimento da estrutura das bases omnidirecionais, foram abordados dois tipos de configurações. Uma das configurações teria formato retangular (vista de topo) com rodas *mecanum* dispostas em paralelo, duas a duas. A outra configuração teria formato circular (vista de topo) e as rodas omnidirecionais utilizadas seriam as universais, à distância de  $90^\circ$  umas das outras. Na Figura 3.27 estão representados estes dois tipos de estruturas.

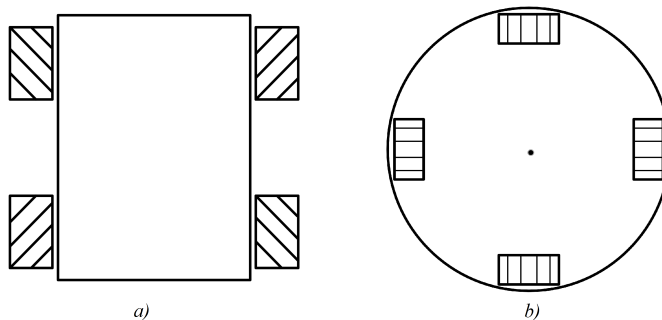


Figura 3.27: Configurações das Bases Desenvolvidas: a) Retangular com 4 Rodas *Mecanum*; b) Circular com 4 Rodas Universais.

Em relação à base com rodas *mecanum*, como mencionado anteriormente, a estrutura da plataforma foi adquirida em conjunto com o restante kit. Esta estrutura é fabricada em alumínio e já vem com as furações e todos os componentes adequados para o acoplamento das rodas e dos motores. Dos motores para as rodas, esta estrutura, possui um sistema de engrenagens guiados por uma corrente, cuja relação entre o número de dentes é tido em conta no cálculo das velocidades de cada uma das rodas.

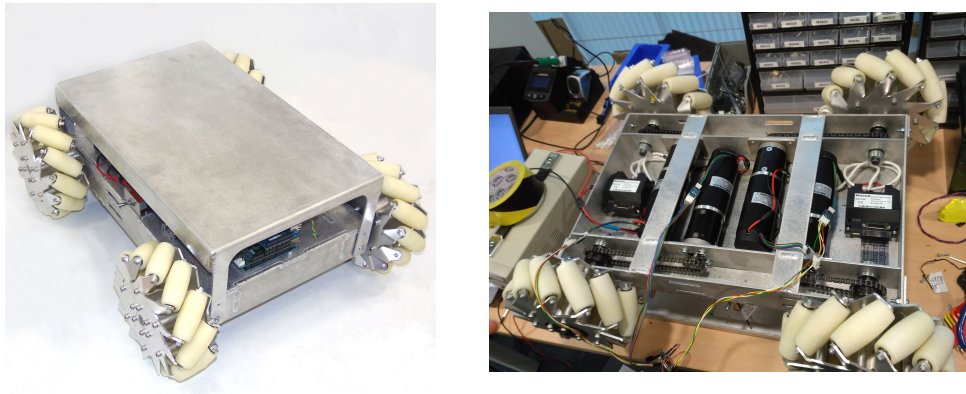


Figura 3.28: Estrutura da Plataforma com Rodas *Mecanum*.

Nas imagens presentes na Figura 3.28 pode-se observar a estrutura metálica já com os componentes base instalados: rodas, motores, *encoders* e controladores.

Para o desenvolvimento da estrutura da base omnidirecional com rodas universais tornou-se necessário ter em conta alguns requisitos, os quais toda a sua organização estrutural deveria considerar, principalmente relativos aos componentes necessários, de acordo com a arquitetura do sistema.

- Forma: Circular;
- Diâmetro Máximo: 700 *mm*;
- Capacidade de Carga:  $\pm 100$  *kg*
- Qtd. de Motores: 4;
- Qtd. de Rodas: 4;
- Qtd. de Controladores de Motores: 2;

Assim, tendo em conta os requisitos mencionados, foi projetado, com auxílio de um *software* CAD (*Computer-Aided Design*) *online* (Onshape), o

primeiro protótipo da estrutura da base omnidirecional (Figura 3.29). Esta base foi projetada tendo em conta todos os componentes necessários e respectivas furações para a sua integração na mesma.

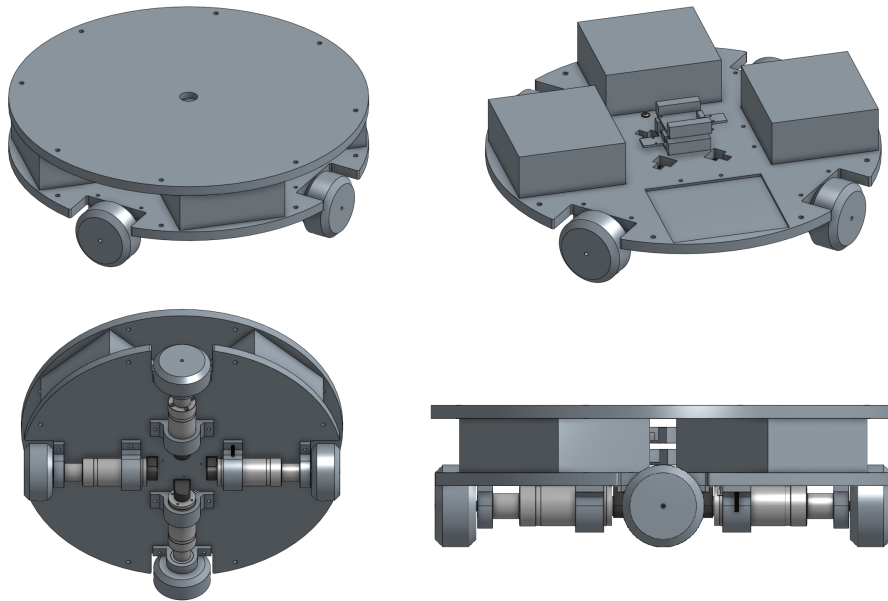


Figura 3.29: Projeto da Base Omnidirecional (Diferentes Vistas).

A plataforma inferior foi desenhada com o intuito de no seu topo (Figura 3.30 lado esquerdo) ser possível integrar quatro baterias (se necessário), dois controladores de motores e ainda sobrar algum espaço para alguns componentes adicionais. Na sua base (Figura 3.30 lado direito) as furações foram projetadas com o objetivo de poder acoplar os quatro motores, através de suportes, assim como possibilitar a ligação dos *encoders* aos controladores. Depois de estruturar todo o posicionamento dos componentes necessários, conseguiu-se um diâmetro da base de  $600\text{ mm}$ , cumprindo-se assim o requisito do diâmetro máximo.

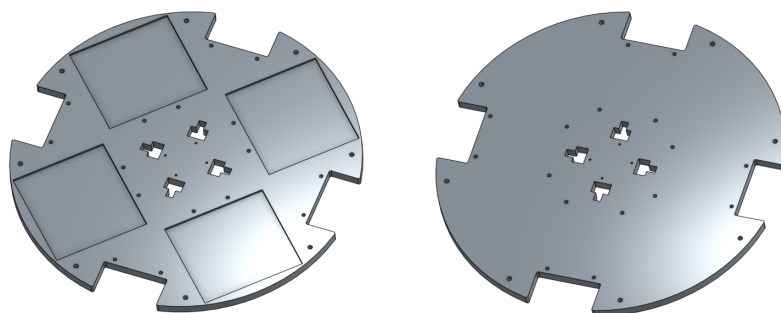


Figura 3.30: Projeto da Base Omnidirecional (Plataforma Inferior).

De modo a realizar o processo de fabrico dos desenhos projetados, foi necessária a conversão destes para a linguagem nativa da ferramenta que, de forma automatizada iria proceder à sua realização. Sendo que se recorreu a uma fresadora CNC (*Computer Numerical Control*) para maquinar a maioria dos componentes desenhados, utilizou-se o *software* Autodesk Fusion 360 para simulação da sua manufatura e conversão dos projetos CAD em linguagem CNC.

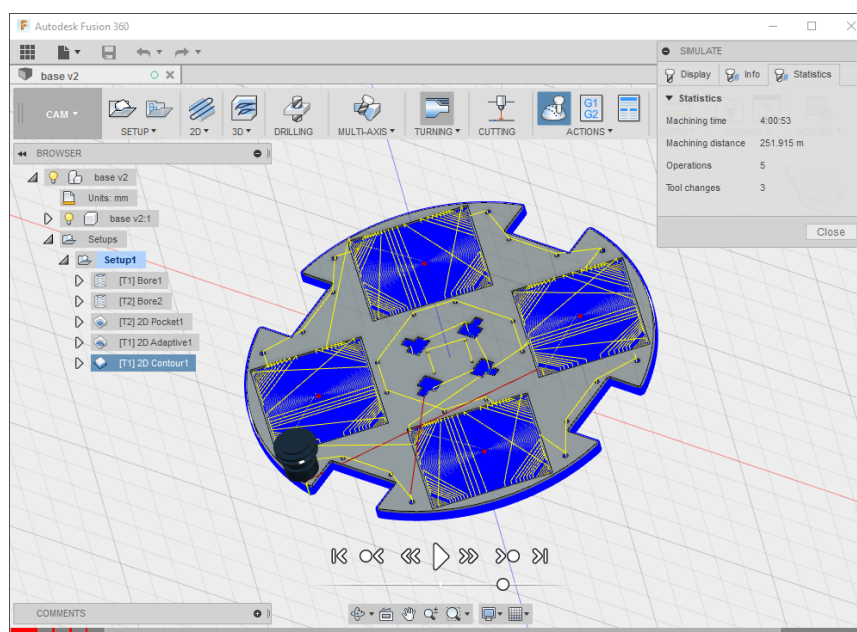


Figura 3.31: Interface do Autodesk Fusion 360.

O Autodesk Fusion 360 é um *software* que permite a integração de módulos de desenho CAD, como modelação de forma livre, renderização, montagem e simulação física. Permite também a tradução desses desenhos para linguagem CNC assim como simulação de todo o processo de fabrico, graças ao seu módulo CAM (*Computer-Aided-Manufacturing*). Na Figura 3.31 está representada a interface do Autodesk Fusion 360, no modo de simulação.

Após tradução para linguagem CNC, foi efetuado o corte de uma placa em MDF (*Medium-Density Fiberboard*), consoante a plataforma previamente desenhada, recorrendo a uma fresadora CNC de três eixos (como mencionado anteriormente), como se pode verificar através da observação da Figura 3.32.

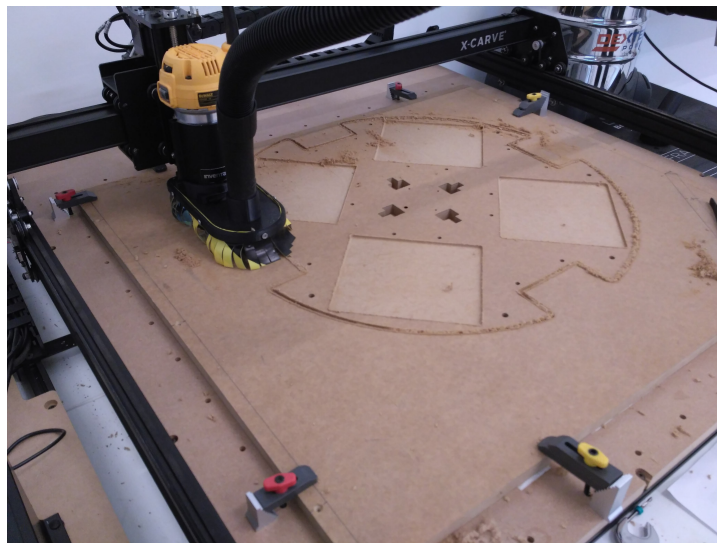


Figura 3.32: Corte CNC da Base Omnidirecional.

Para se poder integrar os quatro motores na plataforma, foram projetados dois suportes para cada motor (Figura 3.33). Na sua idealização e desenho foram utilizadas as mesmas ferramentas de desenho da restante base, mas o processo de fabrico foi diferente.

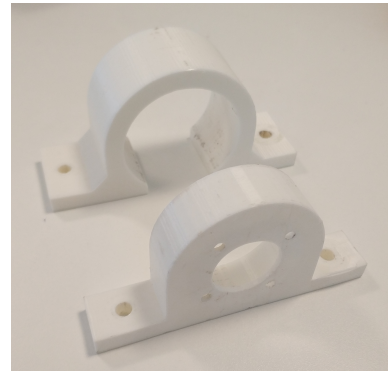
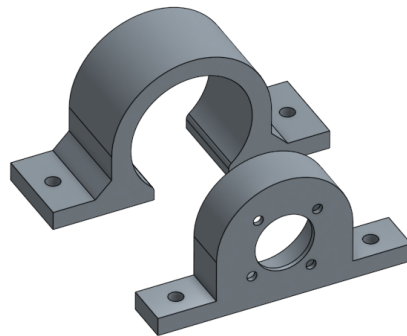


Figura 3.33: Projeto dos Suportes de Motores (à esquerda) e Respetivas Peças (à direita)

Estes suportes foram manufacturados recorrendo a uma impressora 3D, como se observa na Figura 3.34.

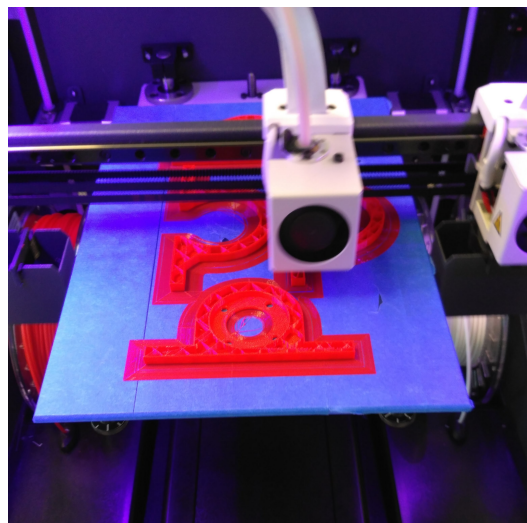


Figura 3.34: Impressão 3D dos Suportes dos Motores.

Do projeto e realização, dos suportes para os motores, e sua aplicação na base, obteve-se o resultado que se pode visualizar na Figura 3.35.

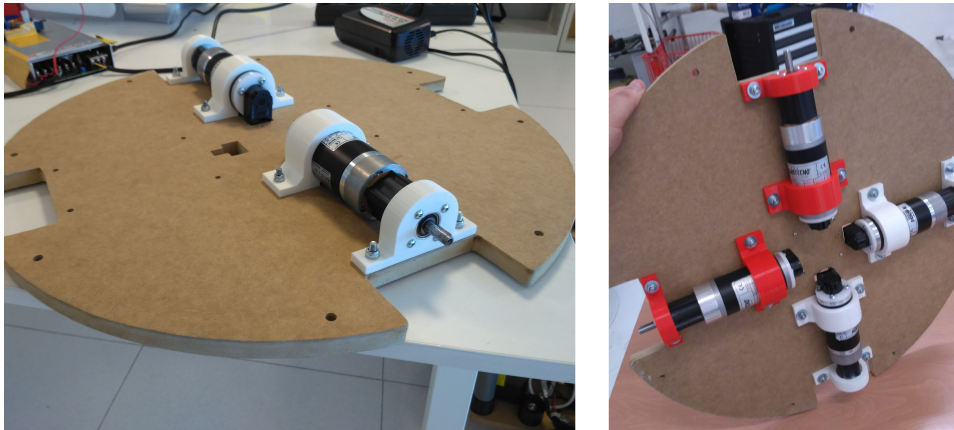


Figura 3.35: Aplicação dos Suportes de Motores na Base.

Com a finalidade de acoplar as rodas omnidirecionais universais aos respectivos motores, visto que estas não possuíam nenhum sistema que possibilitasse a sua interligação aos eixos dos motores, surgiu a necessidade de idealizar uma solução para tal. Assim, foi desenhado um sistema composto por diferentes peças que, consoante a sua forma e especificações, permitiam conectar as rodas aos respectivos eixos dos motores. Na Figura 3.36 está representado o desenho CAD do sistema que foi idealizado.

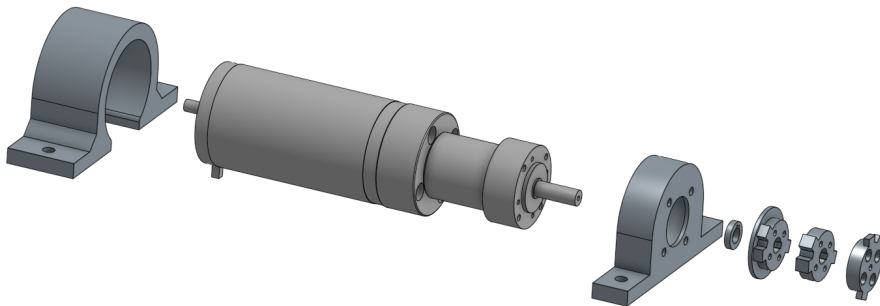


Figura 3.36: Projeto do Sistema de Acoplamento Roda-Eixo.

Depois de projetado este sistema, procedeu-se ao corte CNC destas peças em Acetal, material este que apresenta rigidez e resistência mecânica suficiente para aguentar com toda a carga que o conjunto Roda-Eixo iria suportar, assim como possui elevada resistência ao desgaste. [55] Na Figura 3.37 observa-se o conjunto de todos os componentes do sistema Roda-Eixo.

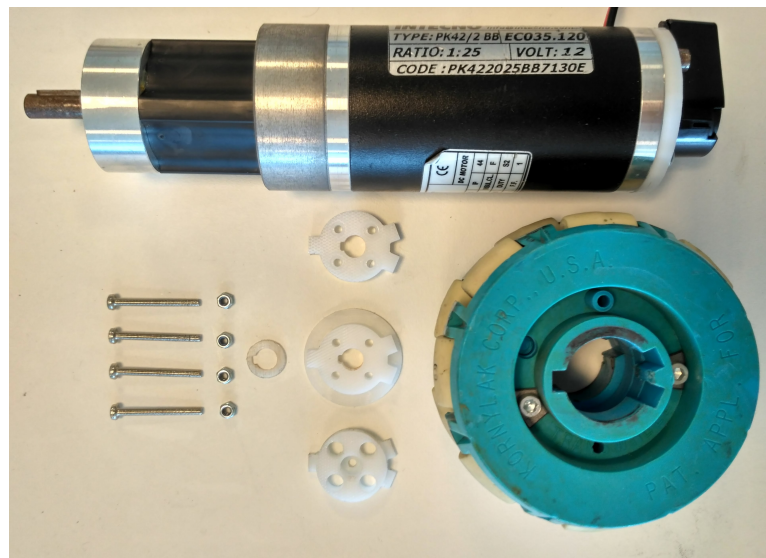


Figura 3.37: Componentes do Sistema de Acoplamento Roda-Eixo.

Por fim, após interligação de todos os componentes e montagem de todas as partes integrantes da estrutura, o resultado final do protótipo da plataforma omnidirecional com rodas universais, pode-se visualizar através da observação da Figura 3.38.

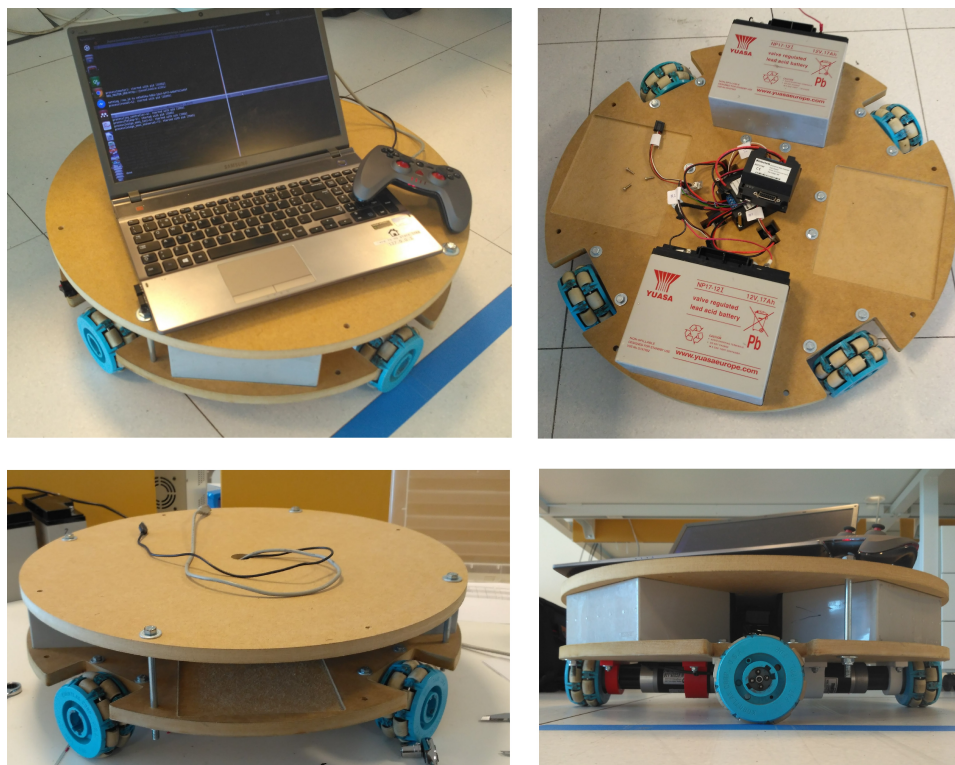


Figura 3.38: Plataforma Omnidirecional com Rodas Universais Desenvolvida.

De referir, que os desenhos técnicos de todas as peças projetadas no desenvolvimento desta plataforma omnidirecional, encontram-se representados no Anexo C.2.

Esta página foi intencionalmente deixada em branco.

## Capítulo 4

# Controlo e Desenvolvimento de *Software*

*No presente capítulo serão apresentados os princípios aplicados relativos ao controlo dos motores das duas plataformas assim como as configurações e software relacionados com os controladores. Serão também mencionados e explicados os modelos de odometria, que permitem obter a estimação do deslocamento e posição da plataforma, nas duas configurações. Após isto, será demonstrado o desenvolvimento de software em ROS, explicando os nós desenvolvidos de acordo com as suas funcionalidades e interação entre si.*

Depois da escolha e montagem de todo o *hardware* necessário para a execução do projeto, o passo seguinte passou pela estruturação e desenvolvimento do *software* que iria conferir a locomoção omnidirecional à plataforma.

Mesmo tendo sido desenvolvidas duas plataformas desenvolvidas com configurações diferentes a estrutura geral na qual assenta o seu sistema de controlo, é similar em ambas. Na Figura 4.1 pode-se visualizar um diagrama de funcionamento do sistema, no qual estão representados os seus elementos principais e de que modo eles interagem entre si.

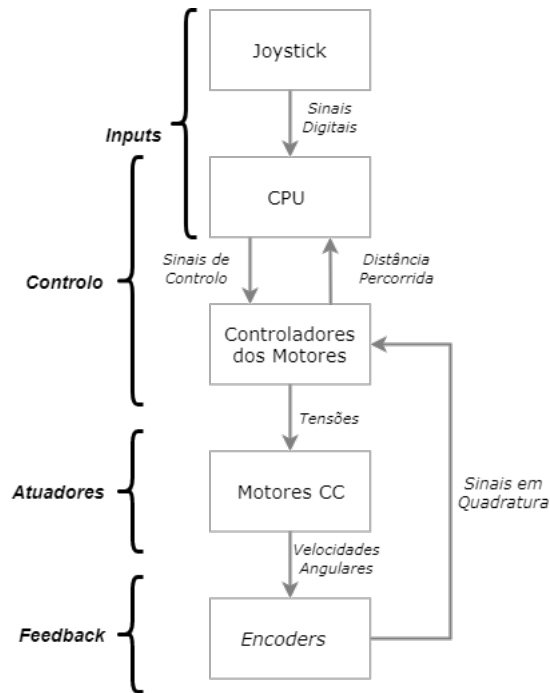


Figura 4.1: Diagrama de Funcionamento do Sistema.

Os *inputs* do sistema que sustenta a plataforma omnidirecional são introduzidos pelo utilizador por intermédio de comandos no CPU ou através de um *joystick*, enviando sinais digitais para o CPU, que os traduzirá em sinais de controlo. O CPU transmite esses sinais de controlo, já processados, para os controladores dos motores que os transformam em valores de tensão, fazendo girar o(s) respetivo(s) motor(es) à velocidade associada. Estes valores de velocidade são captados pelos *encoders*, acoplados aos motores, através de impulsos e são transmitidos para os controladores dos motores sobre a forma de sinais em quadratura que irão dar a informação sobre o deslocamento de cada uma das rodas e permitir estimar a posição da plataforma por intermédio dos cálculos da odometria.

Em termos de *software*, como se pode verificar através da visualização da Figura 4.2, o sistema da plataforma omnidirecional pode-se dividir em duas

## Capítulo 4

partes essenciais: a parte dos controladores dos motores e a parte do computador *Onboard*.

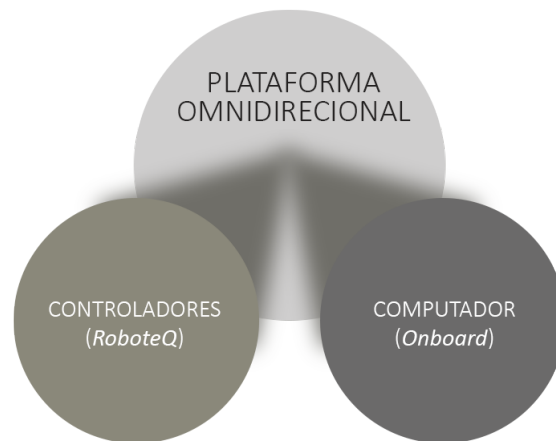


Figura 4.2: Esquema de *Software* da Plataforma Omnidirecional.

Na parte dos controladores dos motores, como o próprio nome indica, são efetuadas todas as configurações necessárias relativas ao controlo assim como são efetuadas as leituras dos valores dos *encoders* que posteriormente são enviadas para o CPU (Computador *Onboard*), por intermédio de um *script*, desenvolvido em MicroBasic, para serem processadas.

A parte de computador *Onboard* é composta por um computador integrado na plataforma que executa um *software*, desenvolvido em ROS, de forma ininterrupta, o qual sustenta a maior parte do processamento necessário para o correto funcionamento do sistema. Este *software*, de um modo geral, é responsável pela leitura dos comandos introduzidos pelo utilizador, cálculo das equações relativas à cinemática, envio dos valores de velocidade relativos aos comandos inseridos e cálculo das equações de odometria representativas da estimação da posição da plataforma.

## 4.1 Controlo

Uma parte fundamental do *software* da plataforma é representada pelo controlo de velocidade. Com o intuito do robô conseguir executar os comandos fornecidos pelo utilizador, é necessário a existência de um mecanismo que traduza esses comandos em sinais compreensíveis pelos motores, de forma a que o seu tempo de reação (tanto de arranque como de paragem) seja tal, que a sua condução possa ser efetuada com segurança. Deste modo, como na maior parte dos sistemas mecânicos, é necessária a implementação de um sistema de controlo cuja resposta possa ser ajustável, de acordo com os requisitos do mesmo. Assim recorreu-se ao controlo PID para os motores da plataforma, nas duas configurações, visto ser um tipo de controlo eficaz em motores CC e pelo *hardware* escolhido, ao nível dos controladores de motores, possuir características e suporte suficiente para a sua aplicação.

### 4.1.1 Controlo PID

O controlo PID é das formas mais comuns de controlo por realimentação. Devido à sua simplicidade e desempenho em diversas aplicações, este tipo de controlo é utilizado em 95% dos processos industriais de malha-fechada. [56] Em relação à robótica móvel, este tipo de controlo é bastante utilizado em sistemas em que existe *feedback* do estado do sistema. A sua vasta utilização deve-se à facilidade de implementação, versatilidade e simplicidade funcional.

Este método de controlo efetua o cálculo de erro entre o valor à saída do processo e o valor de referência (Figura 4.3). O controlador, através do ajuste das entradas que controlam o processo, tenta atenuar o erro da relação entrada-saída.

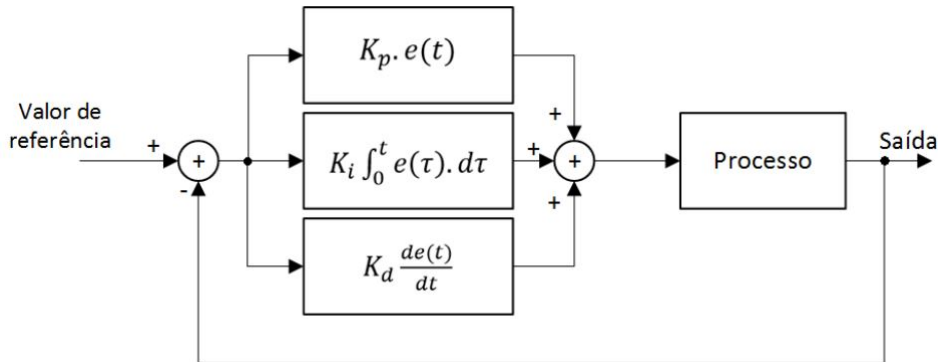


Figura 4.3: Esquema de Controle PID [56].

O cálculo da ação do controlador, envolve três tipos de parâmetros: proporcional (P), integral (I) e derivativo (D). A junção destes parâmetros permite calcular a ação de controle a ser tomada pelo PID, em que a saída de controle é dada pela seguinte equação:

$$u(t) = K_p \cdot e(t) + K_i \int_0^t e(\tau) \cdot d\tau + K_d \frac{de(t)}{dt} \quad (4.1)$$

da qual os ganhos proporcional ( $K_p$ ), integral ( $K_i$ ) e derivativo ( $K_d$ ) são os parâmetros a configurar. Cada um dos ganhos tem um efeito diferente na resposta do processo a controlar, dependendo de cada um, a sua utilização pode melhorar ou piorar a resposta do sistema. Existem diferentes métodos para sintonia dos parâmetros deste tipo de controle. Para processos mais complexos e com necessidade de uma resposta melhor, os métodos de Ziegler-Nichols e Zuang-Atherton, são dois exemplos de métodos que se podem aplicar. No entanto, para processos mais simples, é usual efetuar a sintonia do controle PID através do método manual ou *tentativa-erro*. Dada a complexidade do sistema implementado e os seus requisitos, optou-se por efetuar a sintonia do PID através deste último método.

Mesmo sendo este método manual um método mais convencional, existem certas considerações que se devem ter em conta. As especificações da resposta de um sistema em malha-fechada mudam consoante a alteração dos ganhos do controlador. Para se conseguir um comportamento preciso e suave por parte do sistema é importante perceber-se essas especificações da sua resposta, e de que forma, os ganhos do controlador PID as alteram. Na Figura 4.4 encontra-se representada a resposta de um sistema deste tipo.

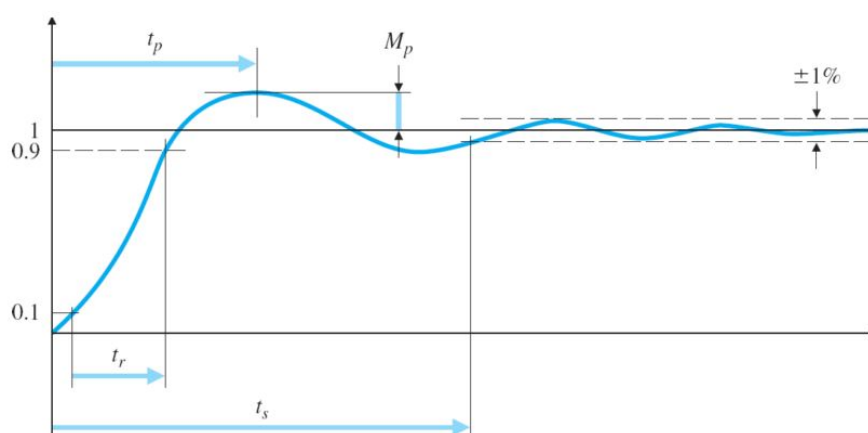


Figura 4.4: Resposta de um Sistema em Malha-Fechada [57].

Em que:

- $t_r$  - Tempo de subida: tempo que o sistema demora a atingir a proximidade do seu valor final pela primeira vez;
- $M_p$  - *Overshoot*: percentagem máxima de sinal que ultrapassa o seu valor final;
- $t_s$  - Tempo de estabelecimento: tempo que o sistema demora a estabilizar no seu valor final;
- $e_{ss}$  - Erro em regime permanente: diferença entre a saída em regime permanente e a entrada de referência.

A relação entre o aumento dos ganhos do controlador PID e as especificações do sistema encontra-se representada na Tabela 4.1

Tabela 4.1: Relação entre os Ganhos do PID e as Especificações do Sistema.

Ganho	Tempo de Subida ( $t_r$ )	Overshoot ( $M_p$ )	Tempo de Estabelecimento ( $t_s$ )	Erro em Regime Permanente ( $e_{ss}$ )	Estabilidade
$K_p$	Diminui	Aumenta	Aumenta Pouco	Diminui	Piora
$K_i$	Diminui Pouco	Aumenta	Aumenta	Diminui Muito	Piora
$K_d$	Diminui Pouco	Diminui	Diminui	Pouca Variação	Melhora

#### 4.1.2 Configuração do Controlador

Para efetuar um controlo eficaz dos motores das plataformas omnidirecionais, foi necessário efetuar a configuração de certos parâmetros no controlador (*RoboteQ* SDC2130). Como mencionado na Secção 3.5.3, a *RoboteQ* disponibiliza um *software* onde se podem configurar todos os parâmetros relativos ao controlador, assim como escolha do modo de operação.

Tendo em conta os requisitos do sistema, o controlador foi configurado no modo *Closed Loop Speed Mode*. Neste modo, como era requerido, o controlador recebe *feedback* de um sensor de velocidade, medindo a velocidade atual e comparando-a com a velocidade desejada. Caso a velocidade real se altere, devido a diferenças de carga na plataformas, o controlador compensa automaticamente o *output* de alimentação para os motores.

Ao utilizar *feedback* medido por *encoders*, o controlador mede a velocidade como sendo o valor em RPM real do motor. Estes valores medidos são escalonados no intervalo de  $-1000$  a  $1000$ , sendo que os valores negativos

estão associados ao sentido negativo de rotação e os positivos ao sentido positivo de rotação. Desta forma, para que o motor atinja a sua velocidade máxima (ou máxima desejada), é necessária configuração do parâmetro *Max Speed* (RPM), parâmetro este que define a velocidade máxima real do motor. Quando o controlador enviar um comando com o valor 1000, o motor mover-se-á à velocidade definida de *Max Speed*. Para além da velocidade máxima também foi necessário configurar o parâmetro *Pulses/Rev*, representativo da resolução dos *encoders*, de forma a obter-se o *feedback* adequado. Na Figura 4.5 a) e b) pode-se observar os valores definidos para a velocidade máxima de controlo dos motores e para resolução dos *encoders*, da base com rodas *mecanum* e da base com rodas universais, respetivamente.

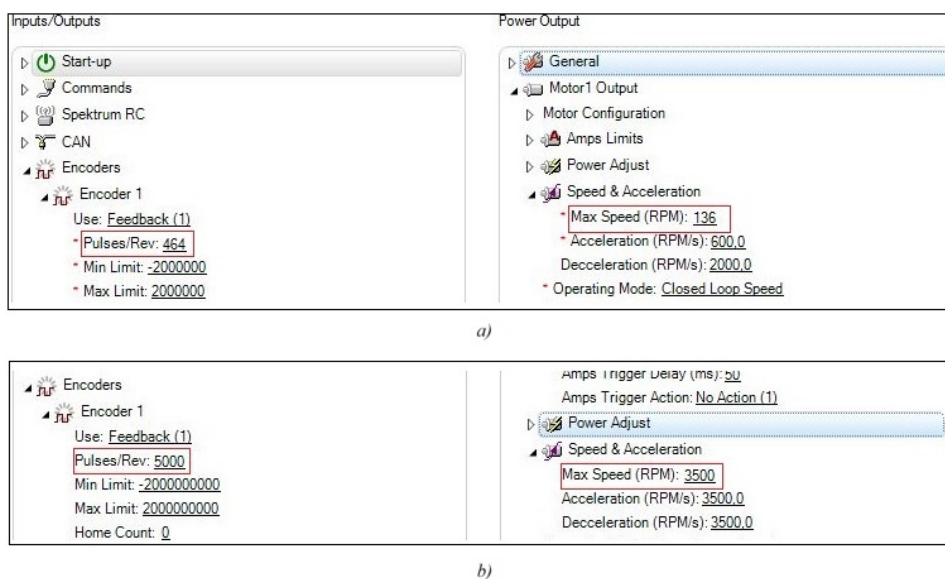


Figura 4.5: Configuração dos Parâmetros *Max Speed* e *Pulses/Rev* no Roborun+.

É de mencionar que, para a base com rodas *mecanum*, o controlo de velocidade é efetuado depois da caixa redutora, pelo facto de, como já foi referido em 3.4.5, não possuímos informação acerca da resolução do *encoder*, tendo sido esta calculada à saída do motor. Também, mesmo tendo sido definida

como velocidade máxima do motor 136 RPM, a roda não irá rodar a essa velocidade. Como existe o sistema de engrenagens entre o motor e a roda, cuja relação foi mencionada também em 3.4.5, a velocidade máxima de cada roda será:

$$Max Vel_{roda} = 136 \times \frac{15}{22} = 92,72 \quad (\text{RPM}) \quad (4.2)$$

Na Figura 4.6 pode-se observar um esquema do sistema de acoplamento da roda ao motor da base com rodas *mecanum* com as respectivas velocidades máximas e resolução de *encoder* calculada, nos diferentes pontos.

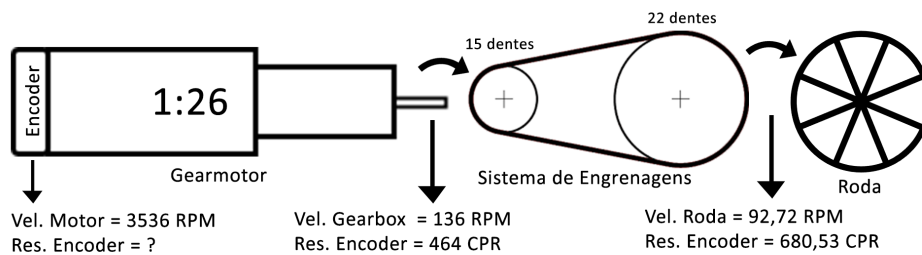


Figura 4.6: Esquema do Sistema de Acoplamento Roda-Motor - Base com Rodas *Mecanum*.

No controlo de velocidade da base com rodas universais, como o valor da resolução do *encoder* era conhecido e a roda estava conectada ao motor por acoplamento direto, como se pode observar no esquema da Figura 4.7, optou-se por efetuar o controlo antes da caixa redutora.

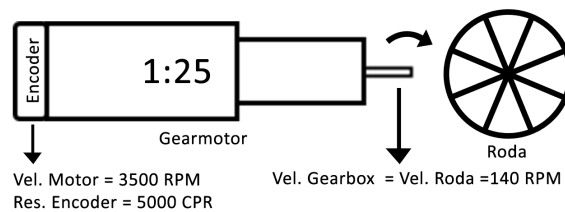


Figura 4.7: Esquema do Sistema de Acoplamento Roda-Motor - Base com Rodas Universais.

### 4.1.3 Descrição do *Loop* de Controlo

O controlador possui um modo de controlo de velocidade (utilizado neste projeto) em malha-fechada que atua através de um algoritmo PID. Esta técnica, como mencionado em 4.1.1, é bastante utilizada em sistemas de controlo e funciona através do ajuste de parâmetros que "modelam" a alimentação de saída, tendo em conta a diferença medida entre a velocidade desejada (definida pelo utilizador) e a velocidade real (medida pelo *encoder*).

A cada 1 milissegundo o controlador mede a velocidade real do motor e subtrai-a à velocidade desejada, de modo a calcular o erro entre as duas. O valor resultante é então multiplicado pelo ganho proporcional ( $K_p$ ), definido pelo utilizador. Desta operação resulta uma das componentes utilizadas para controlar o motor em que, nesta parte do algoritmo, o seu efeito está relacionado com a aplicação de energia no motor, de forma proporcional à diferença entre a velocidade atual e a desejada [58]. Um ganho proporcional mais elevado fará com que o algoritmo aplique um nível de energia maior para um determinado erro, fazendo com que o motor reaja mais rápido às mudanças de comandos e/ou carga aplicada.

A componente derivativa do algoritmo calcula as alterações do erro de 1 em 1 *ms*. Estas alterações serão valores relativamente grandes sempre que ocorram mudanças abruptas no valor da velocidade desejada ou medida. O valor desta alteração é então multiplicado por um ganho derivativo ( $K_d$ ), definido pelo utilizador, que é adicionado ao sinal de saída para os motores. O efeito da parte derivativa fornece um impulso extra de energia ao iniciar o movimento do motor, devido a mudanças no valor de velocidade desejado. Esta componente também ajuda na atenuação de qualquer *overshoot* ou oscilação existentes.

A componente integral do algoritmo efetua a soma do erro ao longo do tempo.

Esta componente ajuda o controlador a alcançar e manter a velocidade desejada quando o erro tende para zero (*i.e.* a velocidade medida está próxima da velocidade real).

Na Figura 4.8 está representado um esquema do algoritmo PID do *RoboteQ*.

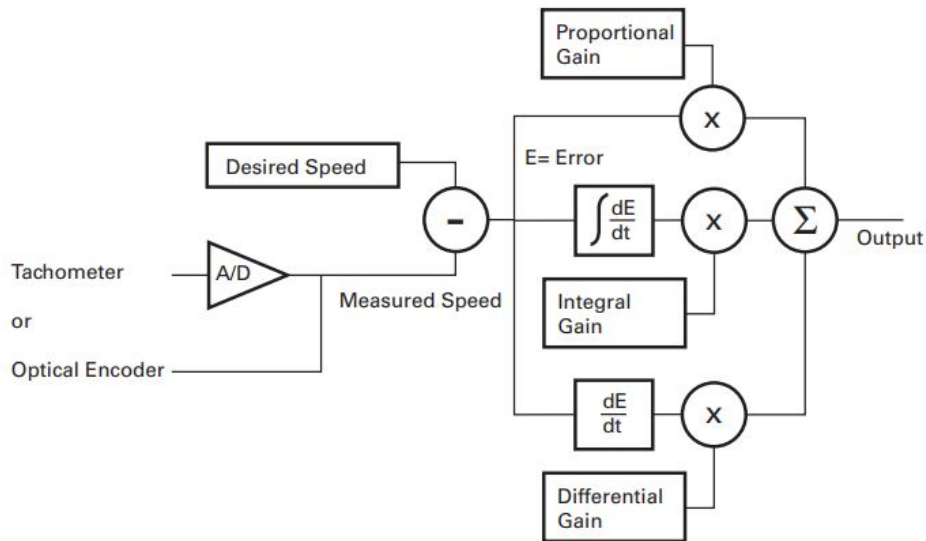


Figura 4.8: Esquema do Algoritmo de Controle PID do RoboteQ [58].

#### 4.1.4 Sintonia/Retificação do PID

Para retificação do controle PID em malha-fechada no controlador, são ajustados os três ganhos, como mencionado em 4.1.1. O principal objetivo da sintonia do controle PID é fazer com que o motor atinja rapidamente a velocidade desejada, sem que ocorra *overshoot* ou grandes oscilações. Para tal, os ganhos PID foram configurados tendo por base a informação da Tabela 4.1 e através da seguinte metodologia:

1. Definir os três ganhos com valor zero;
2. Aumentar o ganho proporcional ( $K_p$ ) até obter uma boa resposta;

3. Diminuir ligeiramente  $K_p$  e aumentar o ganho integral ( $K_i$ ) gradualmente (para melhorar o erro em regime permanente);
4. Adicionar o ganho derivativo ( $K_d$ ) para melhorar a resposta transitória;
5. Testar a resposta a perturbações e mudanças de velocidade;
6. Retificar os ganhos caso necessário.

A aplicação deste método é efetuada com o auxílio do *software* Roborun+, que permite a alteração dos valores dos ganhos e receber *feedback* em formato gráfico da resposta do sistema, como demonstra a Figura 4.9.

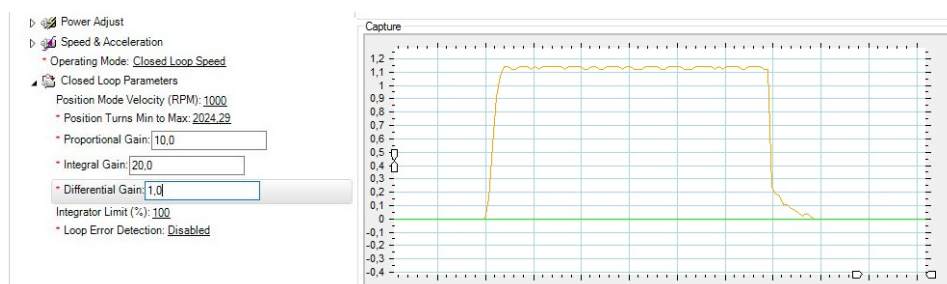


Figura 4.9: Interface para Sintonia do PID - Roborun+.

Assim, da sintonia dos ganhos do controlo PID, pelo método mencionado anteriormente, resultaram os valores presentes na Tabela 4.2.

Tabela 4.2: Ganho dos Controlo PID.

Ganho	Base Rodas	Base Rodas
	<i>Mecanum</i>	Universais
$K_p$	10	0,8
$K_i$	20	1,5
$K_d$	1	0,1

### 4.1.5 *Script* de Envio de *Feedback*

Os valores medidos pelos *encoders*, para além de serem utilizados no *loop* de controlo de velocidade, são necessários para estimação da posição do robô ao longo do tempo, ou seja para a odometria. Desta forma, visto que os cálculos de odometria são efetuados no CPU, de modo a que esta possa receber o *feedback* adequado, foi implementado um *script* simples em MicroBasic, no *RoboteQ*, que envia o valor absoluto dos *encoders* de cada um dos motores, por USB, dos controladores para o CPU, à frequência de 25 *Hz*. O fluxograma deste *script* pode-se observar na Figura 4.10.

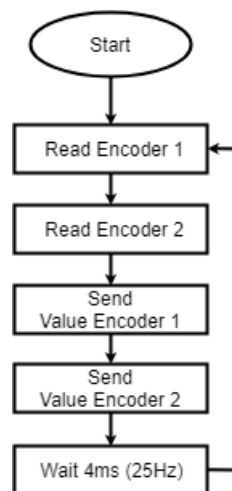


Figura 4.10: Fluxograma do *Script* para Envio de *Feedback*.

Como nas duas configurações são utilizados quatro motores, ou seja, os motores interagem em par com dois controladores, este *script* foi implementado em ambos os controladores de igual forma.

Sendo um dos requisitos principais de *software*, a estimação da posição da plataforma ao longo do seu deslocamento, torna-se necessária a compreensão do método utilizado para tal.

## 4.2 Modelo de Odometria - Quatro Rodas Mecanum

Como mencionado na secção 2.6, a odometria é uma das mais importantes formas de estimação de posição na robótica móvel e baseia-se na integração da informação incremental do movimento rotacional das rodas. Esta permite, a curto prazo, obter uma boa estimativa no deslocamento do robô, mas devido à acumulação de erros, tanto de natureza aleatória como sistemática, não possui grande precisão a longo prazo. No entanto, é um método de implementação rentável e com elevada frequência de amostragem que, quando conjugado com outros métodos de posicionamento absoluto, se torna mais fidedigno na estimação de posições. Na Figura 4.11 está representada a pose do robô, dada pela sua posição  $(x, y)$  e pela orientação  $(\theta)$ , em relação ao referencial cartesiano  $XY$ . [6]

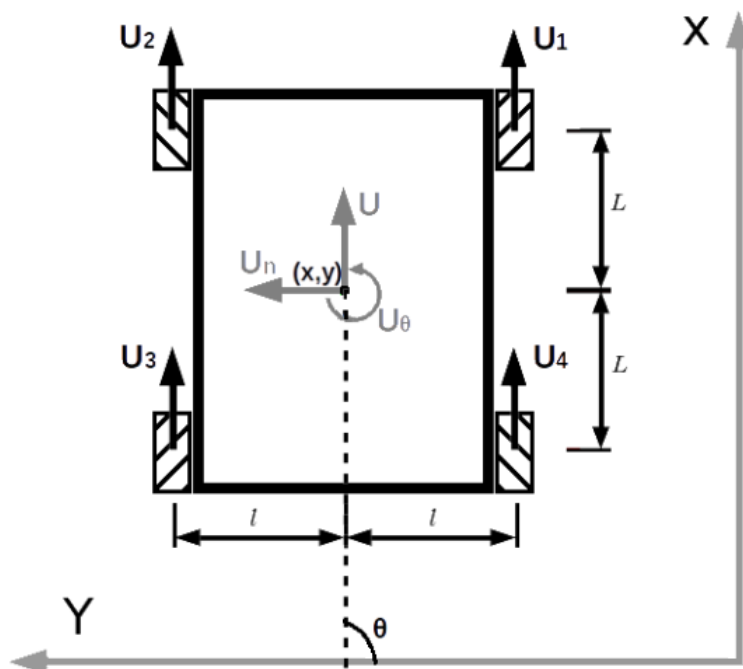


Figura 4.11: Deslocamentos da Plataforma Mecanum com Quatro Rodas.

O deslocamento horizontal de cada uma das rodas é representado por  $U_i$  ( $i = 1, 2, 3, 4$ ) e  $L$  e  $l$  são as distâncias entre o centro da plataforma e as rodas.

Através da informação periódica obtida pelos *encoders*, juntamente com o modelo de odometria, é possível estimar a pose do robô. Na expressão seguinte representa-se a relação entre  $N_i(n)$  e o deslocamento efetuado por cada roda durante o período de amostragem,  $\Delta U_i(n)$ .

$$\Delta U_i(n) = \frac{2\pi \times R}{Imp_r} \times N_i(n) \quad (4.3)$$

Em que:

- $N_i(n)$  - Impulsos do *encoder* durante um determinado período de amostragem ( $T$ );
- $R$  - Raio da roda;
- $Imp_r$  - Número de impulsos por volta completa da roda.

O valor de  $Imp_r$  é calculado tendo em conta a resolução do *encoder* e a relação da caixa redutora do motor, como demonstrado em 4.1.2.

Tendo por base o modelo da cinemática direta apresentado em 2.3.3, a variação dos deslocamento é obtida a partir das equações seguintes.

$$\Delta U(n) = \frac{U_1 + U_2 + U_3 + U_4}{4} \quad (\text{m}) \quad (4.4)$$

$$\Delta U_n(n) = \frac{U_1 - U_2 + U_3 - U_4}{4} \quad (\text{m}) \quad (4.5)$$

$$\Delta U_\theta(n) = \frac{U_1 - U_2 - U_3 + U_4}{4 \times (l + L)} \quad (\text{rad}) \quad (4.6)$$

A atualização da posição do robô ao longo do tempo é efetuada a partir das seguintes equações:

$$\Rightarrow \text{Se } \Delta U_\theta(n) = 0$$

$$x(n+1) = x(n) + \Delta U_n \times \cos(\theta(n)) - \Delta U_n(n) \times \sin(\theta(n)) \quad (\text{m}) \quad (4.7)$$

$$y(n+1) = y(n) + \Delta U_n \times \sin(\theta(n)) + \Delta U_n(n) \times \cos(\theta(n)) \quad (\text{m}) \quad (4.8)$$

$$\theta(n+1) = \theta(n) + \Delta U_\theta(n) \quad (\text{rad}) \quad (4.9)$$

$$\Rightarrow \text{Se } \Delta U_\theta(n) \neq 0$$

$$x(n+1) = x(n) + \frac{1}{\Delta U_\theta(n)} \times [\Delta U(n) \times \sin(\Delta U_\theta(n)) + \Delta U_n(n) \times (\cos(\Delta U_\theta(n)) - 1)] \times \cos(\theta(n) + \frac{\Delta U_\theta(n)}{2}) - \frac{1}{\Delta U_\theta(n)} \quad (4.10)$$

$$\times [\Delta U(n) \times (1 - \cos(\Delta U_\theta(n))) + \Delta U_n(n) \times \sin(\Delta U_\theta(n))] \times \sin(\theta(n) + \frac{\Delta U_\theta(n)}{2}) \quad (\text{m})$$

$$y(n+1) = y(n) + \frac{1}{\Delta U_\theta(n)} \times [\Delta U(n) \times \sin(\Delta U_\theta(n)) + \Delta U_n(n) \times (\cos(\Delta U_\theta(n)) - 1)] \times \sin(\theta(n) + \frac{\Delta U_\theta(n)}{2}) + \frac{1}{\Delta U_\theta(n)} \quad (4.11)$$

$$\times [\Delta U(n) \times (1 - \cos(\Delta U_\theta(n))) + \Delta U_n(n) \times \cos(\Delta U_\theta(n))] \times \sin(\theta(n) + \frac{\Delta U_\theta(n)}{2}) \quad (\text{m})$$

$$\theta(n+1) = \theta(n) + \Delta U_\theta(n) \quad (\text{rad}) \quad (4.12)$$

### 4.3 Modelo de Odometria - Quatro Rodas Universais

O sistema de locomoção omnidirecional da plataforma composto por quatro rodas universais (representado na Figura 4.12), pode ser modelado como dois sistemas diferenciais ortogonais para os cálculos de odometria. O par roda 2 e roda 4 representa um dos sistemas diferenciais e o par roda 1 e roda 3 representa outro. Neste modelo, o deslocamento da plataforma é calculado pela soma do deslocamento realizado pelos dois sistemas diferenciais [32].

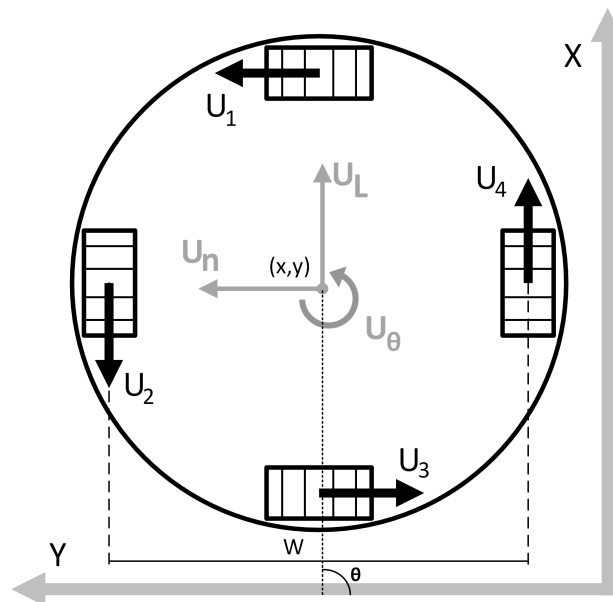


Figura 4.12: Deslocamentos da Plataforma com Quatro Rodas Universais.

Este método é válido uma vez que todas as rodas da plataforma são controladas de forma independente, o que significa que os pares de rodas não estão acoplados entre si, podendo assim ser tratados separadamente nos cálculos de odometria. Deste modo, como este modelo tem por base dois sistemas diferenciais, é importante compreender de que forma a odometria é aplicada a um sistema diferencial normal para estimar a sua posição.

## 4.3.1 Odometria em Sistemas Diferenciais

Um robô com um sistema de direção diferencial possui duas rodas independentes, paralelas entre si, e presas em ambos dos lados do corpo do robô.

Para se entender a aplicação do modelo de odometria a este tipo de configuração, assume-se que os motores que conduzem cada uma das rodas têm acoplados *encoders* que contam  $Imp_r$  impulsos por revolução. A posição do robô em relação ao referencial global é definida pelas coordenadas do ponto médio do eixo traseiro (ponto C na Figura 4.13). Num instante de tempo  $n$ , a localização do robô no plano bidimensional é totalmente definida pelas variáveis  $x_n$ ,  $y_n$  e  $\theta_n$ . Supondo que a posição inicial do robô é conhecida no instante de tempo  $n$ , o objetivo é estimar a sua posição no instante  $n + 1$ .

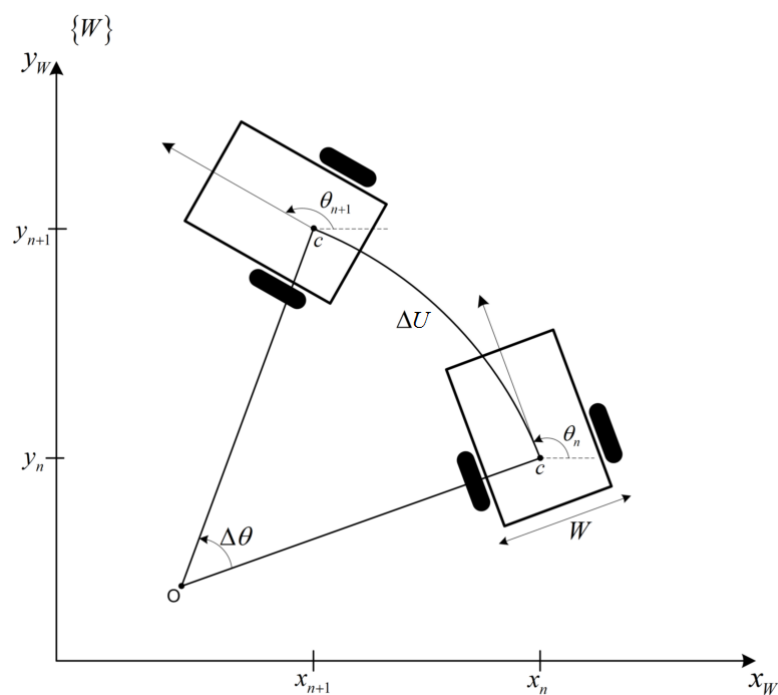


Figura 4.13: Deslocamentos de um Robô Diferencial [32]

Durante um período de amostragem  $T$ , entre os instantes de tempo  $n$  e  $n + 1$ , as distâncias percorridas pelas rodas direita e esquerda,  $\Delta U_r$  e  $\Delta U_l$ , respetivamente, são calculadas a partir das leituras dos *encoders*, da seguinte forma:

$$\Delta U_r = \frac{2\pi \times R}{Imp_r} \times n_r \quad (4.13)$$

$$\Delta U_l = \frac{2\pi \times R}{Imp_r} \times n_l \quad (4.14)$$

Em que  $n_r$  e  $n_l$  são os impulsos contados pelos *encoders* direito e esquerdo, respetivamente,  $R$  é o raio das rodas e  $Imp_r$  é a resolução do *encoder*, em impulsos por revolução.

O deslocamento de translação,  $\Delta U$ , do robô, (i.e. a distância percorrida pelo ponto médio do eixo traseiro) entre os pontos de amostragem é a média das distâncias percorridas pelas duas rodas.

$$\Delta U = \frac{\Delta U_r + \Delta U_l}{2} \quad (4.15)$$

O deslocamento rotacional do robô,  $\Delta \theta$ , é calculado a partir da seguinte equação:

$$\Delta \theta = \frac{\Delta U_r - \Delta U_l}{W} \quad (4.16)$$

Onde  $W$  é a distância entre a roda esquerda e a roda direita.

Tendo calculado os deslocamentos de translação e rotação do robô durante o período de amostragem, a posição do robô no instante  $n + 1$  é estimado pelas seguintes equações:

$$x(n+1) = x(n) + \Delta U \times \cos\left(\theta(n) + \frac{\Delta\theta}{2}\right) \quad (4.17)$$

$$y(n+1) = y(n) + \Delta U \times \sin\left(\theta(n) + \frac{\Delta\theta}{2}\right) \quad (4.18)$$

$$\theta(n+1) = \theta(n) + \Delta\theta \quad (4.19)$$

Como mencionado anteriormente, o sistema omnidirecional com quatro rodas universais pode ser modelado como dois sistemas diferenciais, perpendiculares um ao outro. Uma vez que os dois sistemas diferenciais estão desacoplados, o deslocamento do sistema em geral pode ser obtido através da soma dos dois deslocamentos dos sistemas diferenciais. Assim os cálculos de odometria explicados na secção anterior (4.3.1), para um sistema diferencial, podem ser aplicados para determinar o deslocamento resultante do conjunto de duas configurações diferenciais.

Analogamente ao demonstrado na secção 4.2, para calcular as distâncias percorridas pelas quatro rodas omnidirecionais durante um período de amostragem,  $T$ , tem-se:

$$\Delta U_i(n) = \frac{2\pi \times R}{Imp_r} \times N_i(n) \quad (4.20)$$

Onde  $\Delta U_i$  é a distância percorrida pela roda  $i$  ( $i = 1, 2, 3, 4$ ),  $Imp_r$  é o número de impulsos contados pelo respetivo *encoder*,  $N$  é a resolução do *encoder* e  $R$  o raio da roda.

Os deslocamentos de translação e rotação dos dois pares de rodas (roda 2 com a roda 4 e roda 1 com a roda 3), são calculados a partir das seguintes equações:

$$\Delta U_{(2,4)}(n) = \frac{\Delta U_2 + \Delta U_4}{2} \quad (\text{m}) \quad (4.21)$$

$$\Delta \theta_{(2,4)}(n) = \frac{\Delta U_2 - \Delta U_4}{W} \quad (\text{rad}) \quad (4.22)$$

$$\Delta U_{(1,3)}(n) = \frac{\Delta U_1 + \Delta U_3}{2} \quad (\text{m}) \quad (4.23)$$

$$\Delta \theta_{(1,3)}(n) = \frac{\Delta U_1 - \Delta U_3}{W} \quad (\text{rad}) \quad (4.24)$$

Com isto, o deslocamento rotacional da plataforma em geral,  $\Delta \theta$ , é a média dos deslocamentos produzidos pelos dos sistemas diferenciais.

$$\Delta \theta(n) = \frac{\Delta \theta_{(2,4)} + \Delta \theta_{(1,3)}}{2} \quad (\text{rad}) \quad (4.25)$$

A posição da base robótica no instante de tempo  $n + 1$  (i.e. posição atual) relativa ao referencial geral  $(x, y, \theta)$  é calculado com as equações seguintes:

$$x(n+1) = x(n) + \Delta U_{2,4} \times \cos\left(\theta(n) + \frac{\Delta \theta}{2}\right) + \Delta U_{1,3} \times \cos\left(\theta(n) + \frac{\pi}{2} + \frac{\Delta \theta}{2}\right) \quad (4.26)$$

$$y(n+1) = y(n) + \Delta U_{2,4} \times \sin\left(\theta(n) + \frac{\Delta \theta}{2}\right) + \Delta U_{1,3} \times \sin\left(\theta(n) + \frac{\pi}{2} + \frac{\Delta \theta}{2}\right) \quad (4.27)$$

$$\theta(n+1) = \theta(n) + \Delta \theta \quad (4.28)$$

## 4.4 Desenvolvimento do *Software* em ROS

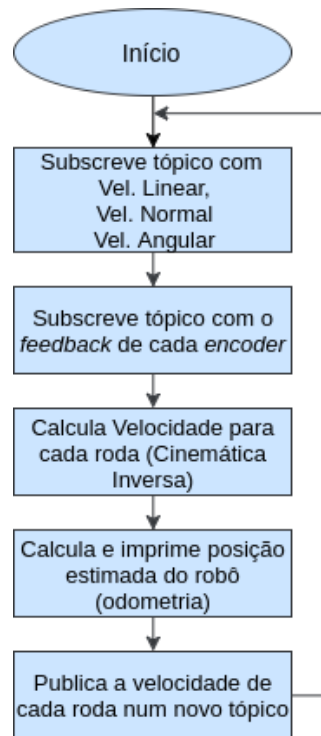
A outra parte fundamental do *software* da plataforma omnidirecional está relacionada com o desenvolvimento, na *framework* ROS, do *software* que é executado no computador integrado no robô. Como referido no início do presente capítulo, este *software* é executado de forma ininterrupta no CPU e é responsável pela interligação de vários elementos do sistema da plataforma, fazendo a ponte entre os comandos introduzidos pelo utilizador (quer através do *joystick* quer diretamente no computador) e os controladores dos motores, assegurando todo o processamento necessário para conferir a omnidirecionalidade à plataforma. Para além disso, é responsável pela receção dos valores de deslocamento lidos pelos *encoders* e respetivo tratamento, de forma a poder efetuar os cálculos de odometria e imprimir as coordenadas relativas à posição estimada da plataforma.

Desta forma foram desenvolvidos em ROS vários *nodes* com diferentes funcionalidades. No Anexo D, encontram-se representados os diagramas de *software* ROS, ilustrativos das interações entre nós, onde os retângulos correspondem aos tópicos subscritos e publicados e as elipses aos nós.

### 4.4.1 Nó `wiigo_omni_mecanum`

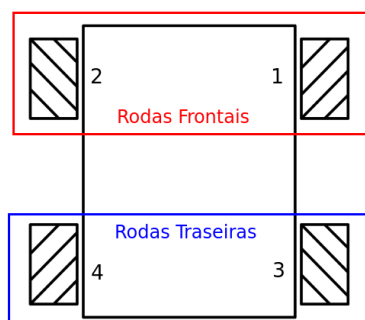
O nó `wiigo_omni_mecanum` é o principal do sistema desenvolvido, para a plataforma com rodas *mecanum*, e é responsável pelos cálculos das velocidades relativas a cada um dos motores, segundo as equações da cinemática, assim como pela estimação da posição da plataforma através dos cálculos das equações de odometria.

Para melhor compreensão do nó desenvolvido apresenta-se na Figura 4.14 um fluxograma representativo da estrutura de *software* do `wiigo_omni_mecanum`.

Figura 4.14: Fluxograma do Nó `wiigo_omni_mecanum`.

Este nó subscreve um tópico com os dados das velocidades linear, normal e angular, de modo a calcular a velocidade para cada uma das rodas de acordo com a Equação 2.27 (mencionada em 2.3.3), representativa da cinemática inversa da configuração omnidirecional com quatro rodas *mecanum*. Depois de calcular as velocidades para cada uma das rodas, estas são publicadas em dois tópicos diferentes, um com as velocidades das duas rodas frontais (Rodas 1 e 2) do robô e outro com as velocidades das rodas traseiras (Rodas 3 e 4). Na Figura 4.15 pode-se visualizar um esquema da numeração atribuída às rodas desta configuração.

Para estimação da posição do robô, este nó subscreve um tópico com os dados relativos ao *feedback* proveniente dos *encoders* de cada uma das quatro rodas. Com estes dados é possível a aplicação do modelo de odometria para este tipo de configuração, mencionado em 4.2. Desta forma, aplicando a

Figura 4.15: Numeração das Rodas da Plataforma com Rodas *Mecanum*.

equação (4.2) a cada uma das rodas, é calculada a variação do respetivo deslocamento por roda. Com isto, calcula-se a variação deslocamentos linear, normal e angular da plataforma, de acordo com as Equações (4.4), (4.5) e (4.6), respetivamente. A posição estimada da plataforma é obtida através da aplicação das Equações (4.7) à (4.12), estimação esta que é atualizada de forma periódica, uma vez que a cada ciclo de programa são efetuadas novas leituras dos *encoders* de forma contínua. Assim, é calculada a posição  $(x, y, \theta)$  da plataforma, correspondente ao seu deslocamento linear, normal e angular ao longo do tempo, respetivamente, segundo o referencial cartesiano apresentado na Figura 4.16.

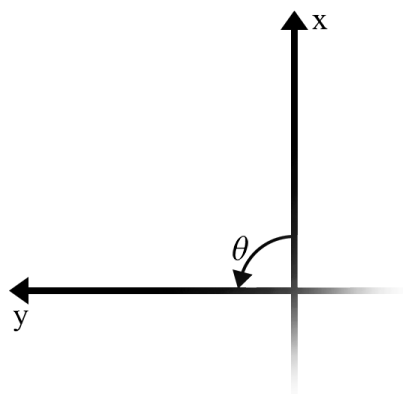


Figura 4.16: Referencial Cartesiano da Posição da Plataforma.

De modo a facilitar a compreensão da orientação angular da plataforma no espaço, este nó efetua a normalização do ângulo  $\theta$ , de estimação da posição do movimento angular, entre  $[-\pi, \pi]$ . Os valores resultantes da aplicação deste modelo são apresentados na consola do computador *onboard* de modo a que utilizador os possa visualizar.

#### 4.4.2 Nó `wiigo_omni_universal`

Da mesma forma que o `wiigo_omni_mecanum` o é na plataforma de configuração com quatro rodas *mecanum*, o `wiigo_omni_universal` é o nó principal do sistema desenvolvido para a plataforma com quatro rodas universais. À semelhança com o mencionado na secção 4.4.1, também este nó é responsável pelos cálculos da cinemática inversa, assim como da estimação da posição do robô por intermédio da odometria.

Como a estrutura e a finalidade deste nó é análoga à do `wiigo_omni_mecanum`, mas com aplicabilidade à configuração com quatro rodas universais, o fluxograma apresentado em 4.14 representa também com clareza as suas funcionalidades, tendo em conta que as equações utilizadas tanto relativas à cinemática como à odometria são a principal diferença.

De forma a efetuar o cálculo da velocidade individual de cada uma das rodas, o nó `wiigo_omni_universal` subscreve o tópico com as velocidades  $V$ ,  $V_n$  e  $\omega$  e aplica a Equação (2.11), apresentada em 2.3.2. Após o seu cálculo estas velocidades são publicadas em dois tópicos: um com as velocidades das rodas 1 e 3 e outro com as velocidades das rodas 2 e 4. Na Figura 4.17 pode-se observar a numeração atribuída a cada uma das rodas nesta configuração.

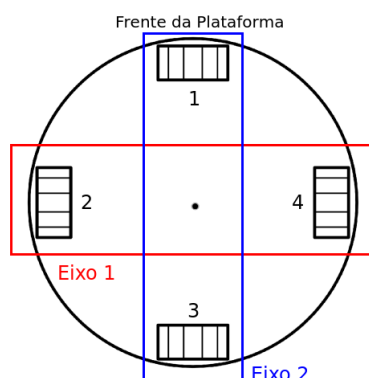


Figura 4.17: Numeração das Rodas da Plataforma com Rodas Universais.

A nomenclatura atribuída ao conjunto de rodas 2-4 e 1-3 como "Eixo 1" e "Eixo 2", respetivamente, foi apenas atribuída por uma questão de facilidade de interpretação acerca de que rodas é que estão associadas a cada controlador. Na realidade não constituem nenhum eixo, pois cada motor é controlado de forma independente.

Com o objetivo de estimar a posição da plataforma no espaço bidimensional, à imagem do *node* mencionado na secção anterior, este também subscreve um tópico com os valores absolutos dos *encoders* acoplados a cada um dos motores. Com estes dados são calculadas as variações de deslocamento associado a cada uma das rodas, pela aplicação da Equação (4.20). Após isto são calculados dos deslocamentos linear, normal e angular da plataforma, de acordo com as Equações (4.21) a (4.25). Com estes, aplicando de forma periódica as Equações (4.26), (4.27) e (4.28), consegue-se obter as coordenadas  $x$ ,  $y$  e  $\theta$ , resultado da estimativa do movimento e posição da plataforma, segundo o referencial da Figura 4.16.

Este nó também efetua a normalização do valor que estima a posição do movimento angular entre  $[-\pi, \pi]$ , para facilitar a sua interpretação. Depois de todos os cálculos efetuados, os valores obtidos através da aplicação deste modelo de odometria são impressos na consola do computador.

### 4.4.3 Nó `wiigo_omni_joy`

O nó `wiigo_omni_joy` foi desenvolvido com a finalidade de movimentar a plataforma robótica. Este nó subscreve um tópico com informações acerca de um *joystick* e de acordo com os botões pressionados ou intensidade dos manípulos analógicos gera um comando de velocidade, composto pelas velocidades linear, normal e angular, que é publicado num tópico. Na Figura 4.18 está representado um fluxograma com a estrutura deste nó.

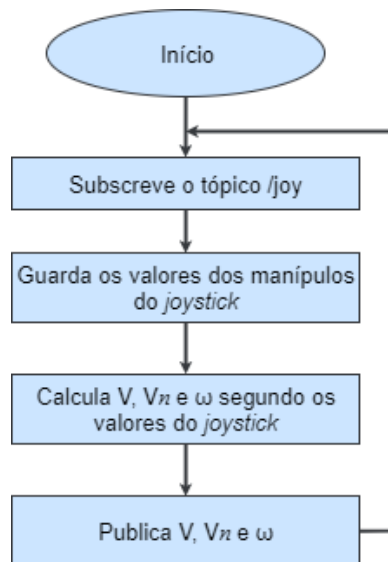

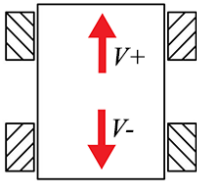
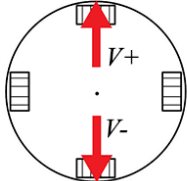

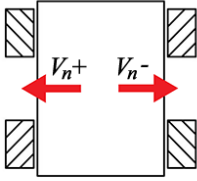
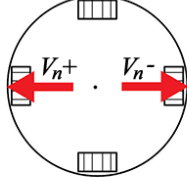

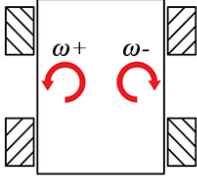
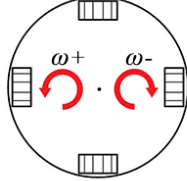


Figura 4.18: Fluxograma do Nó `wiigo_omni_joy`.

O tópico que este nó subscreve é o `/joy`, que pertence a um dos pacotes disponíveis em ROS, e possui toda a informação sobre o estado do *joystick*. O `wiigo_omni_joy` guarda os valores relativos aos botões e manípulos analógicos, os quais utiliza para definir e calcular a velocidade e direção a comandar. Assim definiu-se que o controlo por *joystick* (para ambas as configurações) seria efetuado com os manípulos analógicos. O manípulo analógico esquerdo controla as velocidades lineares ( $V$  e  $V_n$ ) e o manípulo analógico direito a velocidade angular do robô ( $\omega$ ), como representado na Tabela 4.3.

Tabela 4.3: Movimentos das Plataformas Segundo os Comandos do *Joystick*.

<i>Joystick</i>	Plataforma Rodas <i>Mecanum</i>	Plataformas Rodas Universais
		
		
		

Deste modo as três velocidades são calculadas de forma proporcional aos valores dos manípulos analógicos. Caso os manípulos não sejam atuados todos os valores de velocidade são nulos. Estes valores de velocidade são publicado num novo tópico, que será subscrito pelo nó `wiigo_omni_mecanum` ou pelo `wiigo_omni_universal`, dependendo da configuração a que este está associado.

#### 4.4.4 Nós `roboteq_driver`

O conjunto de *nodes* `roboteq_driver` é composto por dois *nodes* com as mesmas funcionalidades, mas dedicados a diferentes partes do *hardware*, o `roboteq_driver_front` e o `roboteq_driver_rear`. Como o próprio nome induz, estes nós estabelecem a comunicação entre o CPU e os respetivos

controladores, das rodas 1 e 2 e das rodas 3 e 4 através de portas serial 0 e 1, respetivamente, no caso da plataforma com rodas *mecanum*.

O `roboteq_driver_front` subscreve o tópico das velocidades das rodas 1 e 2, converte os dados em RPM, escala estes valores num intervalo de  $[-1000; 1000]$ , de modo a serem corretamente interpretados, e envia-os para o *RoboteQ*. É também este nó que recebe os valores absolutos dos *encoders* acoplados às rodas 1 e 2, por intermédio do *script* desenvolvido, mencionado em 4.1.5. Após receção destes dados, os mesmos são publicados num novo tópico, para poderem ser interpretados pelos nós, quer `wiigo_omni_mecanum` (na configuração com rodas *mecanum*) quer `wiigo_omni_universal` (na configuração com rodas universais).

O nó `roboteq_driver_rear`, como referido anteriormente, tem exatamente as mesmas funcionalidades que o primeiro, mas subscreve o tópico das velocidades das rodas 3 e 4 e efetua comunicação com o *RoboteQ* ao qual está associado o *hardware* das rodas 3 e 4.

Como se pode verificar, estes nós são transversais à estrutura de *software* das duas plataformas desenvolvidas, pelo que as suas funcionalidades são requeridas da mesma forma nos dois tipos de configurações. A nomenclatura *front* e *rear* manteve-se, mesmo sendo a sua aplicação mais lógica quando se menciona a plataforma com roda *mecanum*, contudo quando associadas a configuração com quatro rodas universais o `roboteq_driver_front` corresponde ao Eixo 1 e o `roboteq_driver_rear` ao Eixo 2.

#### 4.4.5 *Launch Files*

De forma a que o *software* desenvolvido, para cada uma das plataformas omnidirecionais, funcione corretamente é necessário que todos os nós sejam executados, pois, como se pôde verificar, são vários os exemplos de nós que

publicam tópicos que são subscritos por outros e vice-versa.

Tendo isto em conta, o ROS tem uma ferramenta denominada `roslaunch` que permite a execução de múltiplos nós, local e remotamente (via SSH - *Secure Shell*). Para tal, é necessária a criação de ficheiros de configuração XML (*eXtensible Markup Language*), com a extensão `.launch`, que especifiquem os nós a serem lançados, bem como os parâmetros a definir (caso existam).

Assim foram criados dois *launch files*, um para cada uma das plataformas desenvolvidas, que fazem correr todos os *nodes* relativos a cada uma das configurações.

Para a plataforma de configuração com quatro rodas *mecanum* foi criado o `wiigo_mecanum.launch` que executa os nós mencionados nas secções anteriores, à exceção do nó `wiigo_omni_universal`, pois é relativo à base com rodas universais. Para a plataforma com quatro rodas omnidirecionais universais, foi criado o `wiigo_universal.launch` que executa os mesmos nós que o `wiigo_mecanum.launch`, com a diferença do nó principal.

É de notar, que todo o *software* desenvolvido em ROS foi elaborado com a modularidade suficiente para ser possível de ser integrado ou adaptado à estrutura de *software* já existente do robô wiiGO.

## Capítulo 5

# Testes e Resultados

*Neste capítulo serão apresentados os resultados obtidos após o desenvolvimento do projeto ter sido completo. Também serão mencionados e analisados testes efetuados com as duas plataformas desenvolvidas de forma a averiguar a validade da sua implementação.*

No início do estágio foi idealizado o desenvolvimento de uma plataforma omnidirecional, controlada de forma manual, com capacidade de guardar e mostrar ao utilizador resultados acerca da variação do seu deslocamento assim como da sua posição angular.

Com o decorrer do projeto, decidiu-se pela implementação de duas plataformas com mobilidade omnidirecional, mas com diferentes configurações: uma com quatro rodas *mecanum* e outra com quatro rodas universais.

Desta forma, no final do projeto, foram desenvolvidas as duas plataformas com as características requeridas. Ambas as plataformas são controladas por *joystick* e deslocam-se sem restrições cinemáticas em todas as direções do espaço bidimensional. Também, como requerido, foi desenvolvido um *software*, nas duas configurações que calcula a estimativa do deslocamento da

plataforma ao longo do tempo e mostra a sua posição  $(x, y, \theta)$  ao utilizador no ecrã do computador *onboard*. O resultado final, obtido após interligação de todos elementos de *software* e *hardware*, pode-se visualizar através dos vídeos do funcionamento das duas plataformas desenvolvidas, presentes no CD-ROM que se anexa a esta dissertação .

Por forma a validar a implementação da funcionalidade de estimação de deslocamento e posição, foram realizados testes com as duas plataformas, os quais se apresentam nas secções seguintes.

## 5.1 Testes com a Base de Rodas Mecanum

Foram efetuados testes para validação das equações de odometria, implementadas na base omnidirecional, de modo a verificar a existência de concordância entre o deslocamento real e o calculado pelo o robô. Para tal, foram realizados testes para as distâncias de 1 e 4 metros, assim como rotação de pi ( $\pi$ ) radianos, em dois pisos distintos, um alcatifado e outro liso. Estes testes possibilitaram a análise dos movimentos linear, normal e angular do robô, tanto no sentido positivo como negativo. Na análise dos resultados efetuada, são considerados possíveis erros de medição, pois o método de medição do deslocamento real foi realizado com o auxílio de uma fita métrica, podendo a precisão das medições não ser a mais adequada. No entanto, e tendo em conta o objetivo dos testes em questão, o possível erro não interfere de forma significativa para a análise dos resultados e respetivas conclusões. Na Tabela 5.1, encontram-se representadas figuras exemplificativas dos movimentos efetuados, assim como dos respetivos sentidos, tendo em conta o referencial presente na Figura 5.1.

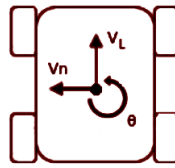


Figura 5.1: Sentidos positivos do Referencial da Plataforma Mecanum.

Tabela 5.1: Movimentos e Sentidos do Robô Consoante Movimento das Rodas.

Movimento	Sentido	
Linear	Positivo	
	Negativo	
Normal	Positivo	
	Negativo	
Angular	Positivo	
	Negativo	

Para a realização dos testes foram efetuados quatro ensaios para cada tipo de movimento, tendo sido depois calculada a média dos resultados obtidos dos quatro ensaios. Os dados recolhidos destes testes foram os valores da odometria, calculados através das equações implementadas no robô, e os valores do deslocamento real da plataforma robótica. Nas tabelas de resultados apresentadas de seguida, estão representados os valores médios dos quatro ensaios realizados para cada movimento e sentido, arredondados à milésima. No Anexo E.1 podem-se observar as tabelas completas, de todos os ensaios realizados de acordo com os vários tipos de movimentos e sentidos.

### 5.1.1 Testes de Deslocamento de 1 m

Para o teste de deslocamento de 1 metro, desenvolveu-se um *node* de teste, que publica o valor de velocidade 0,5 *m/s* durante 2 segundos. No piso alcatifado foram recolhidos os seguintes resultados:

Tabela 5.2: Resultados do Teste de Deslocamento de 1 m - Piso Alcatifado.

Movimento	Sentido	Média Odometria (m)	Média Deslocamento (m)	Média Diferença Odometria/Real (m)	Diferença Deslocamento Expectável e Deslocamento Médio (m)
Linear	Positivo	1,034	1,045	0,018	0,045
	Negativo	1,016	1,048	0,032	0,048
Normal	Positivo	1,024	0,938	0,086	0,062
	Negativo	0,976	0,943	0,046	0,057

Para o movimento linear positivo, obteve-se um valor médio de odometria de 1,034 m e uma média de deslocamento real de 1,045 m, sendo a diferença entre estes dois valores 0,018 m. No movimento linear negativo o valor de odometria foi 1,016 m e a média do deslocamento real 1,048 m, resultando

num erro de 0,032 *m*. Através dos ensaios do movimento normal positivo obteve-se um valor médio de odometria de 1,024 *m* e um valor médio de deslocamento real de 0,938 *m*, dos quais resulta uma diferença média de 0,086 *m*. Para o movimento normal negativo o valor médio de odometria foi 0,976 *m* e o deslocamento médio foi 0,943 *m*, perfazendo um erro de 0,046 *m*.

Em termos gerais, após análise dos resultados deste teste, os valores obtidos foram bastante satisfatórios, tendo sido obtido um erro entre a odometria e valor real na casa dos centímetros para todos os quatro casos. Em relação aos deslocamentos observados, comparativamente com o valor de deslocamento expectável (1 metro), os resultados foram satisfatórios, tendo-se obtido diferenças entre 4 a 6 centímetros, aproximadamente.

No piso liso foram obtidos os seguintes resultados:

Tabela 5.3: Resultados do Teste de Deslocamento de 1 *m* - Piso Liso.

Movimento	Sentido	Média Odometria (m)	Média Deslocamento (m)	Média Diferença Odometria/Real (m)	Diferença Deslocamento Expectável e Deslocamento Médio (m)
Linear	Positivo	1,040	1,083	0,042	0,083
	Negativo	1,013	1,055	0,042	0,055
Normal	Positivo	1,026	1,018	0,024	0,018
	Negativo	1,037	1,079	0,047	0,079

Neste teste o movimento linear positivo apresentou um valor médio de odometria de 1,040 *m* e um deslocamento real médio de 1,083 *m*, resultando em 0,042 *m* de diferença. Em relação ao movimento linear negativo obtiveram-se os valores de 1,013 *m* de odometria e 1,055 *m* de deslocamento real, sendo o erro entre os dois 0,042 *m*. Do movimento normal positivo resultou o valor

de 1,026  $m$  pelas equações de odometria e 1,018  $m$  de deslocamento médio real, sendo a diferença entre os dois de 0,024  $m$ . No movimento normal negativo o valor médio de odometria foi 1,037  $m$  e deslocamento real 1,079  $m$ , resultando em 0,047  $m$  de diferença entre os dois.

Os resultados obtidos através deste teste, em piso liso, foram positivos, apresentando semelhanças com os resultados no piso alcatifado. As diferenças entre os valores de odometria e deslocamento real foram entre os 2 e os 4 centímetros, sendo resultados bastante satisfatórios tendo em conta o ambiente de teste. Comparando os deslocamentos observados com o valor expectável (1 metro), consegue-se verificar que a diferença não foi muito significativa, à semelhança do teste anterior, obtendo-se diferenças de 1 a 8 centímetros, aproximadamente. Contudo, foi possível verificar um deslizamento significativo nas direções perpendiculares às consideradas na medição (p.e. no teste de movimento linear existir deslizamento normal), muito devido às características das rodas *mecanum*, os quais não foram considerados para a recolha destes resultados.

### 5.1.2 Testes de Deslocamento de 4 $m$

Com o intuito de efetuar testes para deslocamentos de 4 metros, alterou-se o *node* desenvolvido para os testes anteriores, de forma a que base omnidirecional se deslocasse à velocidade de 0,5  $m/s$ , durante 8 segundos.

Através dos ensaios no piso alcatifado foram recolhidos os seguintes valores médios:

Tabela 5.4: Resultados do Teste de Deslocamento de 4 m - Piso Alcatifado.

Movimento	Sentido	Média Odometria (m)	Média Deslocamento (m)	Média Diferença Odometria/Real (m)	Diferença Deslocamento Expectável e Deslocamento Médio (m)
Linear	Positivo	4,115	4,205	0,089	0,205
	Negativo	4,164	4,207	0,043	0,207
Normal	Positivo	4,148	3,920	0,228	0,080
	Negativo	4,161	3,918	0,243	0,082

Em relação ao movimento linear positivo a odometria teve o valor de 4,115 m e o deslocamento real foi, em média, 4,205 m, obtendo se uma diferença de 0,089 m entre os anteriores. Dos ensaios do movimento linear negativo resultaram os valores médios de 4,164 m e 4,207 m para a odometria e para o deslocamento real, respetivamente. A diferença entre estes dois valores foi de 0,043 m. No que diz respeito ao movimento normal positivo, os resultados foram de 4,148 m, resultantes das equações de odometria, e 3,920 m de deslocamento médio real, sendo a diferença de 0,228 m. Do movimento normal negativo, obtiveram-se os valores de 4,161 m da odometria e 3,918 m do deslocamento real, diferenciados em 0,243 m.

Tendo em conta os resultados obtidos com este teste, pôde-se observar que a diferença entre os valores calculados pelas equações de odometria e os valores de deslocamento real foi satisfatória nos movimentos lineares, no entanto, nos movimentos normais aumentou significativamente, comparativamente com os testes de deslocamentos menores. Esta diferença de maior significância nos movimentos laterais da base omnidirecional deve-se, muito provavelmente, ao deslizamento dos rolos da rodas *mecanum*, que apresentam um movimento não controlado, pelo que este tipo de movimentos (laterais) são diretamente dependentes do movimento destes rolos.

A diferença entre o deslocamento real e o expectável (4 metros) também aumentou para valores entre 8 e os 20 centímetros, aproximadamente. Este aumento era previsto, pois aumentando a distância, a probabilidade de existirem erros na medição de um movimento contínuo aumenta também.

Na tabela seguinte são apresentados os resultados obtidos dos ensaios com deslocamento de 4 metros no piso liso:

Tabela 5.5: Resultados do Teste de Deslocamento de 4 m - Piso Liso.

Movimento	Sentido	Média Odometria (m)	Média Deslocamento (m)	Média Diferença Odometria/Real (m)	Diferença Deslocamento Expectável e Deslocamento Médio (m)
Linear	Positivo	4,208	4,207	0,004	0,207
	Negativo	4,185	4,193	0,016	0,193
Normal	Positivo	4,163	4,020	0,142	0,020
	Negativo	4,138	4,025	0,113	0,025

Para o movimento linear positivo o valor médio calculado pela odometria foi 4,208 m e o valor médio do deslocamento real foi 4,207 m, sendo a média da diferença entre os dois de 0,004 m. No movimento linear negativo, o valor de odometria foi em média 4,185 m e o deslocamento real foi 4,193 m, sendo a média das diferenças entre os dois 0,016 m. Nos ensaios deste teste, para o movimento normal positivo obteve-se um valor de odometria de 4,163 m e um valor de deslocamento real de 4,020 m, em que o erro existente os dois valores foi 0,142 m. Dos ensaios do movimento normal negativo obtiveram-se os valores de 4,138 m para a média da odometria e 4,025 m para a média dos deslocamentos reais. A média obtida da diferença entre os valores de odometria e os deslocamentos reais foi 0,113 m.

Analogamente aos resultados obtidos no teste anterior, as diferenças entre os

valores de odometria e deslocamento real foram bastante satisfatórias, para os movimentos lineares e menos para os movimentos normais, isto devido aos fatores explicados anteriormente. A diferença entre o deslocamento obtido e o expectável também foi semelhante à do teste anterior, cerca de 20 centímetros para os movimentos lineares, sendo que no caso do movimento normal foi inferior, à volta dos dois centímetros.

### 5.1.3 Teste de Deslocamento de $\pi$ rad

O último teste efetuado foi para validação do movimento angular do robô, para o qual foi modificado o *node* de teste de modo a que a plataforma se movimentasse a  $1 \text{ rad/s}$  durante  $\pi$  segundos, com o objetivo de efetuar o movimento de rotação de  $\pi$  ( $\approx 3.1415926$ ) radianos. A tabela abaixo é representativa dos testes realizados com movimento angular de  $\pi$  radianos, numa superfície alcatifada.

Tabela 5.6: Resultados do Teste de Deslocamento de  $\pi$  rad - Piso Alcatifado.

Sentido	Média Odometria (rad)	Média Deslocamento (rad)	Média Diferença Odometria/Real (rad)	Diferença Deslocamento Expectável e Deslocamento Médio (rad)
Positivo	3,284	3,296	0,035	0,154
Negativo	3,278	3,301	0,053	0,159

Com o teste do movimento angular positivo obteve-se o valor médio de odometria de  $3,284 \text{ rad}$  e um deslocamento médio real de  $3,296 \text{ rad}$ , sendo a diferença média entre os valores de odometria e deslocamento real de  $0,035 \text{ rad}$ . Para o movimento angular negativo foram registados os valores de  $3,278 \text{ rad}$  e  $3,301 \text{ rad}$  das equações de odometria e do deslocamento real,

respetivamente. O erro entre estes ensaios foi em média de  $0,053 \text{ rad}$ .

Da análise dos resultados obtidos com este teste, pode-se verificar que a diferença entre os valores calculados pela odometria e os valores de deslocamento real é muito satisfatória, sendo na ordem das centésimas de radiano. Comparando os valores de deslocamento obtidos com os valores expectáveis, verifica-se que existe um acréscimo de cerca de  $0,15 \text{ rad}$  nos dois sentidos.

Na superfície mais lisa foram recolhidos os seguintes valores médios:

Tabela 5.7: Resultados do Teste de Deslocamento de  $\pi \text{ rad}$  - Piso Liso.

Sentido	Média Odometria (rad)	Média Deslocamento (rad)	Média Diferença Odometria/Real (rad)	Diferença Deslocamento Expectável e Deslocamento Médio (rad)
Positivo	3,278	3,331	0,053	0,189
Negativo	3,281	3,315	0,043	0,173

Para os movimentos angulares, neste tipo de piso, os resultados foram muito semelhantes. Do movimento angular positivo registou-se um valor médio de odometria de  $3,278 \text{ rad}$  e um valor médio de deslocamento de  $3,331 \text{ rad}$ , com uma diferença média entre os dois de  $0,053 \text{ rad}$ . Nos ensaios do movimento angular negativo registou-se a média de  $3,281 \text{ rad}$  das equações de odometria e  $3,315 \text{ rad}$  da média do deslocamento real, com um erro médio de  $0,043 \text{ rad}$ .

Neste tipo de teste, de forma semelhante com o piso de alcatifa, os resultados foram positivos, apresentando uma diferença muito reduzida entre o valor calculado pela odometria o deslocamento real. Em relação à diferença entre o deslocamento expectável ( $\pi$  radianos) e o deslocamento realizado pela plataforma, observou-se um excedente entre os  $0,17$  e  $0,19$  radianos.

Na generalidade os testes efetuados demonstraram resultados positivos, tendo-se observado diferenças pequenas entre os deslocamentos reais da plataforma omnidirecional e os valores calculados pelas equações de odometria implementadas. Dos resultados da diferença entre o valor de deslocamento expectável e o valor de deslocamento efetivamente realizado, observou-se que quanto maior for o deslocamento em questão, maior será essa diferença. Em relação aos resultados obtidos no piso liso, é notável um maior deslizamento do robô. Este deslizamento era expectável, devido às características que permitem a omnidirecionalidade da plataforma, nomeadamente as características das rodas *mecanum*. Neste tipo de superfície também se pôde constatar que o robô realizava menos tração quanto maior fosse a velocidade imposta nas rodas, assim como em movimentos rápidos e de consecutivas mudanças de direção. No entanto, fora estes casos mais específicos, conseguiu-se obter resultados muito semelhantes com os obtidos nos ensaios na superfície de alcatifa.

Da análise dos dados obtidos dos diversos testes, tendo em conta o erro obtido, foi possível validar as equações de odometria implementadas na plataforma omnidirecional.

É de mencionar que um dos propósitos do desenvolvimento desta plataforma é que, num futuro próximo, esta possua mobilidade omnidirecional autónoma para seguimento de pessoas. Após validação das equações de odometria, esta base omnidirecional foi sujeita a um teste piloto no local onde iria atuar. Após este teste foi concluído que, tendo em conta a superfície onde se irá deslocar e o material dos rolos das rodas *mecanum*, a plataforma apresentava um deslizamento considerável. Para resolução desta situação as rodas desta plataforma foram substituídas por outras com as mesmas dimensões e configuração, mas com o material dos rolos diferente (borracha), como se pode observar na Figura 5.2.



Figura 5.2: Rodas Mecanum com Rolos de Borracha Aplicadas.

Após novos testes no local, verificou-se que o deslizamento reduziu de uma forma bastante significativa, apresentando o robô o comportamento esperado.

## 5.2 Testes com a Base de Rodas Universais

De forma a validar as equações de odometria estudadas e implementadas na plataforma omnidirecional com quatro rodas universais desenvolvida, efetuaram-se testes de concordância entre o deslocamento realizado pela base e o calculado pelas equações de odometria. Analogamente aos testes efetuados com a base de rodas *mecanum*, realizaram-se testes para deslocamentos de 1 e 4 metros e também de rotação  $\pi$  radianos, tanto no sentido positivo como negativo, em dois tipos de superfície: uma composta por um piso alcatifado e outra por um piso liso. Da mesma forma que nos testes da base *mecanum*, o método utilizado para medição do deslocamento real da plataforma foi efetuado com recurso a uma fita métrica, podendo existir algum erro associado a esta medição. Contudo, a inclusão deste erro nos testes em questão não invalida a sua viabilidade. Nesta configuração, foi definido um referencial representativo dos sentidos dos movimentos lineares, normais e angulares da base omnidirecional, referencial este que se encontra representado na Figura 5.3.

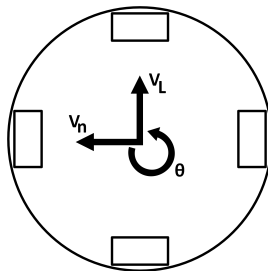


Figura 5.3: Sentidos positivos do Referencial da Plataforma com Rodas Universais.

Na Tabela 5.8 observam-se os movimentos efetuados pela base, tendo em conta o movimento das rodas, nos seus diferentes tipos e sentidos, tendo por base o referencial da Figura 5.3.

Tabela 5.8: Movimentos e Sentidos do Robô Consoante Movimento das Rodas.

Movimento	Sentido	
Linear	Positivo	
	Negativo	
Normal	Positivo	
	Negativo	
Angular	Positivo	
	Negativo	

Na realização destes testes foram feitos quatro ensaios para cada tipo de movimento, calculando-se de seguida a média dos quatro ensaios. Os valores recolhidos foram referentes aos valores de odometria, calculados pelas equações implementadas no robô, e aos deslocamentos reais efetuados pela plataforma. Os valores apresentados nas tabelas de resultados correspondem aos valores médios dos quatro ensaios, arredondados às milésimas de metro,

sendo que no anexo E.2, podem-se consultar as tabelas com todos os ensaios efetuados para todos os tipos de movimentos.

### 5.2.1 Teste de Deslocamento de 1 $m$

No primeiro teste, foi desenvolvido um *node* em ROS de teste, o qual publicava o valor de velocidade  $0,5 \text{ m/s}$  durante 2 segundos. No piso de alcatifa obtiveram-se os seguintes resultados:

Tabela 5.9: Resultados do Teste de Deslocamento de 1  $m$  - Piso Alcatifado.

Movimento	Sentido	Média Odometria (m)	Média Deslocamento (m)	Média Diferença Odometria/Real (m)	Diferença Deslocamento Expectável e Deslocamento Médio (m)
Linear	Positivo	0,998	0,883	0,115	0,118
	Negativo	1,006	0,884	0,123	0,117
Normal	Positivo	1,004	0,939	0,065	0,061
	Negativo	1,001	0,921	0,081	0,079

Neste teste para o movimento linear positivo obteve-se um valor médio da odometria de  $0,998 \text{ m}$  e um valor médio de deslocamento real de  $0,883 \text{ m}$ , sendo a média da diferença entre os anteriores de  $0,115 \text{ m}$ . No movimento linear negativo a media da odometria foi  $1,006 \text{ m}$  e do deslocamento efetivo foi  $0,884 \text{ m}$ , resultando numa diferença média de  $0,123 \text{ m}$ . No caso do deslocamento normal positivo a média dos valores de odometria obtidos a partir dos quatro ensaios foi de  $1,004 \text{ m}$  e do deslocamento real foi de  $0,939 \text{ m}$ , em que a média das diferenças entre a odometria e o deslocamento real fez o valor de  $0,065 \text{ m}$ . Em relação ao deslocamento normal negativo das equações de odometria resultou um valor médio de  $1,001 \text{ m}$  e de deslocamento real  $0,921 \text{ m}$ , dos quais resultou a média do erro entre os dois de  $0,081 \text{ m}$ .

Após análise dos resultados deste teste, obteve-se um erro entre os valores de odometria e os valores de deslocamento real de cerca de 6 a 12 *cm* para todos os diferentes casos. No que diz respeito aos deslocamentos observados em comparação com o valor expectável (1 metro), o valor do erro foi semelhante, variando entre 6 e 11 *cm*.

No piso liso obtiveram-se os seguintes resultados:

Tabela 5.10: Resultados do Teste de Deslocamento de 1 *m* - Piso Liso.

Movimento	Sentido	Média Odometria (m)	Média Deslocamento (m)	Média Diferença Odometria/Real (m)	Diferença Deslocamento Expectável e Deslocamento Médio (m)
Linear	Positivo	1,007	0,943	0,064	0,057
	Negativo	0,983	0,949	0,052	0,051
Normal	Positivo	1,007	0,957	0,050	0,043
	Negativo	1,007	0,961	0,046	0,040

Neste caso, para o movimento linear positivo, o valor médio de odometria foi de 1,007 *m* e o deslocamento real médio foi 0,943 *m*, obtendo-se uma diferença média de 0,064 *m* entre os dois. No deslocamento linear negativo a média das equações de odometria foi 0,983 *m* e do seu deslocamento efetivo foi de 0,949 *m*, resultando num erro médio de 0,052 *m*. Em relação ao movimento normal positivo foram registados valores médios de 1,007 *m* e 0,957 *m*, da odometria e do deslocamento real, respetivamente, e destes ensaios resultou uma diferença média de 0,050 *m* entre os dois. Para o movimento normal negativo foi observado o valor médio das equações de odometria de 1,007 *m* e o valor médio do deslocamento efetuado pela plataforma de 0,961 *m*, perfazendo um erro médio de 0,046 *m*.

Os resultados observados através da realização deste teste foram satisfató-

rios, tendo apresentado melhorias no erro, em comparação com o teste em superfície de alcatifa. As diferenças entre os valores de odometria e deslocamento real obtidas foram desde os 4 aos 6 *cm*, resultados estes considerados positivos. Em relação à comparação entre o valor de deslocamento expectável e o deslocamento realizado pela base omnidirecional, verificaram-se valores médios de diferença entre os 4 e os 5 *cm*.

### 5.2.2 Teste de Deslocamento de 4 *m*

Com a finalidade de realizar testes para deslocamentos de 4 metros, modificou-se o *node* desenvolvido de forma a que a base omnidirecional se deslocasse realizando movimentos linear e normal, à velocidade 0,5 *m/s*, pelo período de 8 segundos.

Dos ensaios realizados no piso alcatifado, foram recolhidos os seguintes resultados:

Tabela 5.11: Resultados do Teste de Deslocamento de 4 *m* - Piso Alcatifado.

Movimento	Sentido	Média Odometria (m)	Média Deslocamento (m)	Média Diferença Odometria/Real (m)	Diferença Deslocamento Expectável e Deslocamento Médio (m)
Linear	Positivo	3,980	3,491	0,490	0,510
	Negativo	3,977	3,568	0,410	0,433
Normal	Positivo	3,944	3,796	0,148	0,204
	Negativo	3,999	3,729	0,270	0,271

Dos ensaios do movimento linear positivo as equações de odometria implementadas ditaram um valor médio 3,980 *m*, enquanto que o valor médio do deslocamento real foi de 3,491 *m*. A média da diferença entre os dois anteriores resultou num valor de 0,490 *m*. Para o caso do movimento linear negativo

obtiveram-se os valores de 3,977 *m* e 3,568 *m*, da odometria e do deslocamento real, respetivamente, o que resultou num erro médio de 0,410 *m* entre as duas variáveis a testar. Do movimento normal positivo registaram-se os valores de 3,944 *m* e 3,796 *m* para as médias de odometria e deslocamento real, respetivamente, e a sua diferença apresentou um valor médio de 0,148 *m*. Em relação ao movimento normal negativo observou-se um valor médio de odometria de 3,999 *m* e um valor médio de deslocamento real de 3,729 *m*, resultando num valor médio da diferença entre os dois de 0,270 *m*.

Considerando os resultados obtidos através da realização deste teste, pôde-se observar que as diferenças entre os valores calculados através das equações de odometria e os deslocamentos efetivamente realizados pela base aumentaram significativamente, apresentando valores entre 14 e os 49 *cm*. Neste teste, os resultados foram mais favoráveis para os deslocamentos do movimento normal do que para os do movimento linear.

A diferença entre o deslocamento expectável (4 metros) e o deslocamento real também registou um aumento significativo, apresentando valores médios desde 20 *cm* até cerca de 50 *cm*. Este aumento do erro para distâncias maiores era esperado, pois tendo-se verificado a sua existência em menor quantidade numa distância inferior, com o aumento do deslocamento é de esperar que o erro aumente também.

Na tabela seguinte apresentam-se os resultados médios obtidos dos ensaios de deslocamento de 4 metros, em superfície lisa:

Tabela 5.12: Resultados do Teste de Deslocamento de 4 m - Piso Liso.

Movimento	Sentido	Média Odometria (m)	Média Deslocamento (m)	Média Diferença Odometria/Real (m)	Diferença Deslocamento Expectável e Deslocamento Médio (m)
Linear	Positivo	4,005	3,762	0,243	0,238
	Negativo	4,005	3,779	0,226	0,221
Normal	Positivo	4,005	3,839	0,166	0,161
	Negativo	4,006	3,831	0,174	0,169

No movimento linear positivo, o valor de odometria foi em média 4,005 m e o deslocamento real foi 3,762 m, sendo a média das diferenças entre os dois 0,243 m. Dos resultados deste teste para o movimento linear negativo o valor médio calculado pela odometria foi 4,005 m e o valor médio do deslocamento real foi 3,779 m, apresentando uma diferença em média de 0,226 m. Dos ensaios do movimento normal positivo obtiveram-se os valores de 4,005 m das equações de odometria e 3,839 m para o deslocamento efetivamente realizado. A média da diferença entre os valores de odometria e os de deslocamento real foi 0,166 m. Já para o movimento normal negativo, da odometria observou-se um valor médio de 4,006 m e do deslocamento real um valor médio de 3,831 m, ensaio este em que a média da diferença entre os valores calculados e os obtidos do deslocamento real foi de 0,174 m.

Dos resultados obtidos deste teste, em comparação com o mesmo teste em piso de alcatifa, pode-se observar que a diferença entre o valor calculado pelas equações implementadas e o deslocamento real da plataforma, diminuiu significativamente, apresentando valores entre 16 cm e 23 cm. Esta melhoria de resultados neste tipo de piso está relacionada com o facto do material das rodas utilizadas, apresentar melhor aderência em contacto com a superfície mais lisa, do que com a superfície alcatifada. No segundo tipo de superfície

as rodas, por vezes, afundam de uma forma parcial, fazendo com que a tração não seja tão eficaz, e efetuam movimento de rotação sobre o eixo do motor mas sem realizar deslocamento por parte do robô.

Em relação à diferença entre o deslocamento real e o expectável, os valores foram muito idênticos aos anteriores, pelo facto dos valores obtidos pelas equações de odometria terem sido bastante aproximados do valor de deslocamento esperado, mesmo tendo sido o deslocamento real mais discrepante.

### 5.2.3 Teste de Deslocamento de $\pi$ rad

Da mesma forma que nos testes com a base de rodas *mecanum*, o último teste efetuado foi o de validação do movimento angular do robô. Para este teste também foi modificado o *node* de teste de modo a que a base se movimentasse sobre si própria à velocidade de 1 *rad/s* durante  $\pi$  segundos, de forma a efetuar um movimento rotacional de  $\pi$  radianos.

Na Tabela 5.13 estão representados os resultados dos testes efetuados para o movimento angular de  $\pi$  radianos, na superfície com alcatifa:

Tabela 5.13: Resultados do Teste de Deslocamento de  $\pi$  rad - Piso Alcatifado.

Sentido	Média Odometria (rad)	Média Deslocamento (rad)	Média Diferença Odometria/Real (rad)	Diferença Deslocamento Expectável e Deslocamento Médio (rad)
Positivo	3,224	3,194	0,032	0,052
Negativo	3,233	3,230	0,031	0,088

Através do teste do movimento angular positivo, obtiveram-se os valores médios de 3,224 *rad* e 3,194 *rad* das equações de odometria e do deslocamento

real. A média do erro obtido neste ensaio foi de  $0,032 \text{ rad}$ . Deste mesmo movimento (angular) mas no sentido negativo, observou-se, da média dos ensaios realizados, o valor de  $3,233 \text{ rad}$  da odometria e de  $3,230 \text{ rad}$  do deslocamento efetivo. A diferença média entre os dois fatores a considerar foi de  $0,031 \text{ rad}$ .

Dos resultados obtidos através deste teste, verificou-se que os valores foram muito positivos, apresentando pequenas discrepâncias entre os valores calculados pelas equações de odometria implementadas e o deslocamento real. Em relação ao deslocamento expectável em comparação com o deslocamento real, os resultados também foram satisfatórios, apresentando uma diferença de  $0,052 \text{ rad}$  no sentido positivo e de  $0,088 \text{ rad}$  no sentido negativo.

Dos ensaios efetuados na superfície lisa, foram recolhidos os seguintes resultados médios:

Tabela 5.14: Resultados do Teste de Deslocamento de  $\pi \text{ rad}$  - Piso Liso.

<b>Sentido</b>	<b>Média Odometria (rad)</b>	<b>Média Deslocamento (rad)</b>	<b>Média Diferença Odometria/Real (rad)</b>	<b>Diferença Deslocamento Expectável e Deslocamento Médio (rad)</b>
Positivo	3,199	3,199	0,014	0,058
Negativo	3,212	3,193	0,019	0,052

Em relação aos movimento angulares realizados neste tipo de piso, os resultados foram muito semelhantes aos obtidos no piso anterior. Para o movimento angular positivo registou-se um valor médio de odometria de  $3,199 \text{ rad}$  e um valor médio de deslocamento de  $3,199 \text{ rad}$ , tendo-se obtido, da média dos ensaios realizados, a diferença de  $0,014 \text{ rad}$ . Nos ensaios do movimento angular negativo registou-se a média de  $3,212 \text{ rad}$  a partir das equações de

odometria e 3,193 *rad* da média do deslocamento real, perfazendo um erro médio de 0,019 *rad*.

De forma semelhante ao sucedido nos testes deste tipo de movimento no piso com alcatifa, os resultados obtidos foram positivos. A diferença entre o valor calculado pela odometria e o deslocamento real foi bastante reduzida, na ordem das centésimas de radiano. Em relação à diferença entre o valor de deslocamento expectável e o deslocamento efetivamente realizado pela plataforma, verificaram-se diferenças de 0,058 *rad* no sentido positivo e 0,052 *rad* no sentido negativo.

Após a realização de todos os testes, dos três diferentes movimentos, nos dois pisos distintos, pode-se afirmar que este tipo de plataforma com quatro rodas universais, tendo em conta o material das suas rodas, apresenta um comportamento melhor na superfície lisa do que na superfície alcatifada.

Os problemas de deslizamento observados nos testes com a base de rodas *mecanum* não se verificaram de forma tão evidente na configuração com este tipo de rodas. Contudo, neste tipo de configuração, na generalidade dos casos, a diferença entre o resultado das equações de odometria e o deslocamento efetuado na realidade pela plataforma, foi significativamente maior do que no testes da secção 5.1. A exceção ocorreu nos testes rotacionais (movimento angular) em que, da mesma forma que para a base *mecanum*, foram muito satisfatórios. Isto dever-se-á ao facto do movimento angular ser o único movimento nesta configuração, no qual as quatro rodas estão a ser controladas, e em rotação ao mesmo tempo, permitindo que exista mais tração para movimentar a plataforma sem deslizamento nem rotação em falso, por parte das rodas.

Por fim, mediante a análise dos dados obtidos dos diferentes tipos de movimentos realizados pela base omnidirecional e, tendo em conta o erro obtido

nos diferentes ensaios, foi possível validar a implementação das equações de odometria na plataforma omnidirecional desenvolvida.

Esta página foi intencionalmente deixada em branco.

## Capítulo 6

# Conclusões e Trabalho Futuro

*Ao longo desta dissertação foram apresentadas as opções tomadas para a realização da base omnidirecional, assim como as convenientes justificações. Com o objetivo de explicar, de forma resumida, toda essa informação, serão mencionadas neste capítulo as principais conclusões relativas ao projeto desenvolvido assim como algumas propostas de trabalho futuro, para melhoria do mesmo. Por fim, será mencionada uma pequena apreciação pessoal, por parte do autor, conclusiva da presente dissertação.*

Como mencionado na Secção 1.1, todo o conteúdo apresentado nesta dissertação resultou do trabalho efetuado durante um estágio curricular, com a duração de seis meses, na *Follow Inspiration*. Desde o estudo das mais variadas temáticas até ao desenvolvimento e implementação de *software* e *hardware*, respetivamente, passando por variadas fases de testes, este projeto foi objeto de diferentes fases, que no fim culminaram na elaboração da presente dissertação.

O objetivo principal, definido inicialmente, estava relacionado com o desenvolvimento de um protótipo de uma plataforma robótica omnidirecional, de modo a conferir locomoção sem restrições holonómicas ao robô wiiGO Indus-

try. Para além de apresentar mobilidade omnidirecional, um dos requisitos do sistema a desenvolver, passava por integrar um método que estimasse a sua posição durante o seu percurso, relativamente ao ponto onde começou a sua marcha.

Desta forma, foi idealizado um sistema composto por quatro rodas omnidirecionais, atuadas por quatro motores de forma individual, cujo controlo seria efetuado por intermédio de controladores de motores e o processamento a cargo de um CPU *Onboard*. Deste sistema fariam parte sensores que efetuariam as medidas do deslocamento de cada uma das rodas, por forma a poder calcular o deslocamento da plataforma.

Após o desenvolvimento e implementação deste sistema idealizado surgiu, por parte da empresa, a necessidade de criação de uma nova plataforma, com os mesmos requisitos, mas com algumas diferenças estruturais, essencialmente ao nível do tipo de rodas a utilizar.

Assim, tendo por base os objetivos propostos, inicialmente e ao longo do estágio, foram concretizadas duas plataformas omnidirecionais funcionais, de configurações diferentes, controladas por *joystick* e dotadas de um sistema de estimação de posição ao longo do tempo, sistema este que foi testado e validado, como demonstrado no Capítulo 5. Tendo como objetivo a integração, ou adaptação, destas plataformas robóticas ao sistema já desenvolvido pela empresa (o wiiGO), todas as decisões de quer de *hardware* como de estruturação de *software*, foram tomadas de forma a que esse processo seja simples ou até mesmo imediato.

À data da escrita deste documento, a plataforma omnidirecional com quatro rodas *mecanum* já se encontra em fases de testes, de forma a verificar a adaptabilidade das suas funcionalidades no local onde irá atuar. É também de mencionar que após validação do sistema de controlo manual desta pla-

## Capítulo 6

taforma, foi desenvolvido (pela empresa), sobre o mesmo, um sistema que permite que esta apresente mobilidade omnidirecional autónoma, através de SLAM (*Simultaneous Localization and Mapping*). Este sistema recorre aos valores de odometria calculados e, através da sensores que fazem o varrimento laser do ambiente, confere à base omnidirecional um mecanismo de mapeamento, perceção e localização, que lhe permite deslocar-se no espaço de forma autónoma, sem colidir com nenhum obstáculo estático ou dinâmico.

Em relação à plataforma omnidirecional com rodas universais desenvolvida, encontra-se em estudo, por parte da empresa, de forma a verificar se a sua aplicação ao wiiGO, ou a outro projeto, é viável e/ou rentável, visto que foi um protótipo funcional desenvolvido na integra na empresa, cujo funcionamento foi validado.

Como propostas de trabalho futuro seria interessante, da mesma forma que foi efetuado para a base com rodas *mecanum*, tornar a plataforma com rodas universais autónoma também. Para isso seria necessária a implementação de sensores laser e implementação de algoritmos de SLAM, de um modo geral, visto que o restante *software* já se encontra estruturado e desenvolvido. Também seria necessária a implementação de um computador dedicado *onboard* assim como o circuito de alimentação dimensionado para o mesmo, bem como para os sensores, visto que durante o desenvolvimento desta dissertação foi utilizado um computador portátil como CPU *onboard*. Refere-se que, após o desenvolvimento proposto relativo à plataforma com roda *mecanum*, este último ponto mencionado, relativo ao CPU e circuito de alimentação, foi implementado pela equipa de desenvolvimento da *Follow Inspiration*.

Como considerações finais, após a conclusão deste projeto torna-se relevante mencionar que a oportunidade de estágio que me foi concedida pela *Follow Inspiration* contribuiu de uma forma muito positiva e enriquecedora para o meu desenvolvimento pessoal e profissional, pelo facto de me ter proporcio-

nado experiências e desafios, na área da Engenharia e não só, os quais espero ter estado à altura. Tendo sido a minha intenção inicial a de efetuar um estágio em ambiente empresarial em projetos de inovação, sinto, no fim desta caminhada, que foi uma aposta ganha em todos os sentidos. O contacto com situações e problemas reais e o contacto e dinâmica com uma equipa de trabalho multidisciplinar bem organizada, permitiram-me ter uma perspetiva diferente acerca de trabalho em cooperação.

Para além disso, a aplicação de conhecimentos apreendidos ao longo do meu percurso académico e a necessidade constante de aprendizagem de novas matérias, que foi surgindo no decorrer do projeto, foram fatores extremamente estimulantes e recompensadores, que em muito contribuíram para o meu crescimento enquanto profissional.

Acima de tudo, é bastante satisfatório concluir que, no final, toda a dedicação e empenho depositados neste projeto deram frutos, e podem vir a representar parte de um produto a ser aplicado a curto prazo, pela empresa à qual deixo aqui expresso o meu agradecimento.

# Referências Documentais

- [1] C. V. R. Coutinho, *Robótica Móvel - Sistema de Condução Autónoma*. Tese de mestrado, Instituto Superior de Engenharia de Lisboa, 2014.
- [2] M. Silva, “ROBIN: Robótica Industrial,” pp. 1–9, 2012.
- [3] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Springer, 1st editio ed., 2008.
- [4] H. Secchi, *Uma Introdução aos Robôs Móveis*. Instituto de Automática – INAUT, 2012.
- [5] F. Inspiration, “Follow Inspiration - wiiGO tests continues.” <http://www.followinspiration.pt/post21.html>, 2015. (Acedido em: 2017-06-14).
- [6] F. J. M. Sá, “Sistema de navegação para plataforma móvel omnidireccional,” Master’s thesis, Faculdade de Engenharia da Universidade do Porto, 2016.
- [7] I. Conduraru, I. Doroftei, and A. Conduraru, “A Mobile Robot with Modified Mecanum Wheels,” no. October, 2015.
- [8] H. Oliveira, “Análise do Desempenho e da Dinâmica de Robôs Omnidireccionais de Três e Quatro Rodas,” Master’s thesis, Faculdade de Engenharia Universidade do Porto, 2007.

- [9] P. Editora, “holonomia in Dicionário infopédia da Língua Portuguesa com Acordo Ortográfico.” <https://www.infopedia.pt/dicionarios/lingua-portuguesa/holonomia>. (Acedido em: 2017-04-05).
- [10] R. Siegwart, *Introduction to Autonomous Mobile Robots*, vol. 53. 2011.
- [11] H. Yu, “Omni-directional mobility using active split offset castors,” pp. 1–7, 1997.
- [12] I. Doroftei, V. Grosu, and V. Spinu, “Omnidirectional Mobile Robot – Design and Implementation,” *Bioinspiration and Robotics: Walking and Climbing Robots*, no. September, pp. 511–528, 2007.
- [13] H. Asama, M. Sato, L. Bogoni, H. Kaetsu, A. Matsumoto, and I. Endo, “Development of an Omni-Directional Mobile Robot with 3 DOF Decoupling Drive Mechanism,” *IEEE International Conference on Robotics and Automation*, pp. 1925–1930, 1995.
- [14] K. Kanjanawanishkul, “Omnidirectional wheeled mobile robots : wheel types and practical applications,” *Int. J. Advanced Mechatronic Systems*, vol. 6, no. 6, 2015.
- [15] H. P. Oliveira, A. J. Sousa, A. P. Moreira, and P. Costa, “Modeling and Assessing of Omni-directional Robots with Three and Four Wheels,” *CONTEMPORARY ROBOTICS - Challenges and Solutions*, pp. 109–138, 2009.
- [16] T. Braünl, *Embedded robotics*. second ed., 2006.
- [17] M. Olimpiu, D. Mândru, I. Ardelean, and A. Ple, “Design and Development of an Autonomous Directional Mobile Robot with Mecanum Wheels,” 2014.
- [18] P. Oliveira, A. J. Sousa, A. P. Moreira, and P. J. Costa, “Precise Modeling of a Four Wheeled Omni-directional Robot,” *Proceedings of the 8th*

- Conference on Autonomous Robot Systems and Competitions*, pp. 57–62, 2008.
- [19] K.N.Wakchaure, “Kinematics Modelling of Mecanum Wheeled Mobile Platform,” *Mechatronics and Machine Vision in Practice*, no. M2VIP 2008 15th International Conference on, pp. 657 – 662, 2008.
- [20] H.-c. Huang and C.-c. Tsai, “Adaptive Robust Control of an Omnidirectional Mobile Platform for Autonomous Service Robots in Polar Coordinates,” pp. 439–460, 2008.
- [21] B. Siciliano and O. Khatib, *Springer Handbook Robotics*. second ed., 2016.
- [22] I. Fetch Robotics, “AMRs vs AGVs: What’s the difference?.” <http://fetchrobotics.com/amrs-vs-agvs/>, 2017. (Acedido em: 2017-04-27).
- [23] C. Sprunk, B. Lau, P. Pfaff, and W. Burgard, “An accurate and efficient navigation system for omnidirectional robots in industrial environments,” *Autonomous Robots*, 2016.
- [24] K. Aktiengesellschaft, “KUKA omniMove.” <https://www.kuka.com/en-de/products/mobility/mobile-platforms/kuka-omnimove/>. (Acedido em: 2017-04-21).
- [25] Kuka, “Masterful maneuvering in confined spaces : KUKA omniMove .,” 2015.
- [26] M. Wada and H. H. Asada, “Design and control of a variable footprint mechanism for holonomic omnidirectional vehicles and its application to wheelchairs,” *IEEE Transactions on Robotics and Automation*, vol. 15, no. 6, pp. 978–989, 1999.

- [27] J.-T. Zou, “The Development of the Omnidirectional Mobile Home Care Robot,” *Mobile Robots – Current Trends*, pp. 345–362, 2011.
- [28] R. Tan, S. Wang, and Y. Jiang, “Motion Control of a Cushion Robot Considering Load Change and Center of Gravity Shift,” *Journal of System Design and Dynamics*, vol. 6, no. 5, pp. 740–753, 2012.
- [29] T. Fukuda, J. Huang, P. Di, and K. Sek, “Motion Control and Fall Detection of Intelligent Cane Robot,” in *Intelligent Assistive Robots: Recent Advances in Assistive Robotics for Everyday Activities*, vol. 106, pp. 317–337, 2015.
- [30] Toyota Motor Corporation, “Toyota Shifts Home Helper Robot R&D into High Gear with New Developer Community and Upgraded Prototype.” <http://newsroom.toyota.co.jp/en/detail/8709541/>, 2015. (Acedido em: 2017-06-07).
- [31] A. Martinez and E. Fernández, *Learning ROS for Robotics Programming*. 2013.
- [32] S. Ginzburg, “Design and Development of an Autonomous Navigation System For an Omni-Directional Four-Wheeled Mobile Robot,” Master’s thesis, University of Ontario Institute of Technology, 2012.
- [33] O. Cordero Morales, “Puesta en marcha y documentación del robot móvil para exteriores Andábata,” bachelor’s thesis, Escuela Técnica Superior de Ingeniería Industrial, 2016.
- [34] S. Ginzburg and S. Nokleby, “Indoor Localization of an Omni-Directional Wheeled Mobile Robot,” *Canadian Society for Mechanical Engineering*, vol. 37, no. 4, pp. 1043–1056, 2013.
- [35] A. M. Harrington and C. Kroninger, “Characterization of Small DC Brushed and Brushless Motors,” March 2013.

- [36] AspenCore, “Electronics Tutorials: Pulse Width Modulation.” <http://www.electronics-tutorials.ws/blog/pulse-width-modulation.html>, 2017. (Acedido em: 2017-07-04).
- [37] S. W. Colton and D. Frey, “Design and Prototyping Methods for Brushless Motors and Motor Control,” master of science in mechanical engineering, Massachusetts Institute of Technology, 2010.
- [38] Think RC, “Brushed and Brushless Motors.” <http://www.thinkrc.com/faq/brushless-motors.php>. (Acedido em: 2017-07-05).
- [39] Dicionário Priberam da Língua Portuguesa, “torque.” <https://www.priberam.pt/dlpo/torque>, 2013. (Acedido em: 2017-07-05).
- [40] K. Kowalewsky and L. Electric, “Top tips for working with gearmotors,” pp. 24-27, 2009.
- [41] Anaheim Automation, “Gearbox Guide.” <http://www.anaheimautomation.com/manuals/forms/gearbox-guide.php>, 2017. (Acedido em: 2017-07-05).
- [42] M. Dynamics, “Which Gearbox?.” <https://www.motiondynamics.com.au/gearboxes-which-type.html>, 2015. (Acedido em: 2017-07-05).
- [43] Robin R. Murphy, *An Introduction to AI Robotics*. Massachusetts Institute of Technology, 2000.
- [44] L. Tamagawa seiki Co, “Rotary Encoders FA-CODER.” <http://www.tamagawa-seiki.com/english/encoder/about-encoder.html>. (Acedido em: 2017-07-04).
- [45] G. Monari and A. Abacus, “Understanding Resolution In Optical And Magnetic Encoders,” 2013.
- [46] J. Kimbrell, “Fundamentals of Industrial Encoder Sensing,” pp. 1-9, April 2013.

- [47] A. Technologies, “Datasheet AEDM-5xxx,” 2016.
- [48] A. Bhatt, “DC Motor Control Using H Bridge.” <https://www.engineersgarage.com/contribution/dc-motor-control-using-h-bridge>, 2012. (Acedido em: 2017-07-20).
- [49] A. Tantos, “H-Bridges – the Basics.” <http://modularcircuits.tantosonline.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/>, 2011. (Acedido em: 2017-07-20).
- [50] RoboteQ, “RoboteQ SDC21xx Datasheet,” 2016.
- [51] RoboteQ, “MicroBasic Overview.” <http://www.roboteq.com/index.php/technology-menu/microbasic-technology>. (Acedido em: 2017-06-21).
- [52] D. Engineering, “Sabertooth 2x25 V2 User’s Guide,” April 2012.
- [53] D. Engineering, “Sabertooth 2X25 Quick Start Guide,” Julho 2007.
- [54] D. Engineering, “Kangaroo x2 Motion Controller,” Março 2013.
- [55] U. LLC, “Acetal Plastic.” <https://plastics.ulprospector.com/generics/2/acetal>, 2017. (Acedido em: 2017-07-14).
- [56] G. F. D. Pinto, “Controlo de um Motor Diesel com Turbo- Compressor Utilizando Redes Neurais,” Master’s thesis, Instituto Superior de Engenharia do Porto, 2016.
- [57] R. d. S. Barbosa, “SISCA - Sistemas de Controlo Avançado: Controlo PID - Parte 1,” 2016.
- [58] RoboteQ, “Advanced Digital Motor Controllers - User Manual,” 2012.

*Anexo A. Datasheets do  
Hardware Utilizado*



## A.1. Datasheets Motores CC

### A.1.1. Gearmotor IG52-04

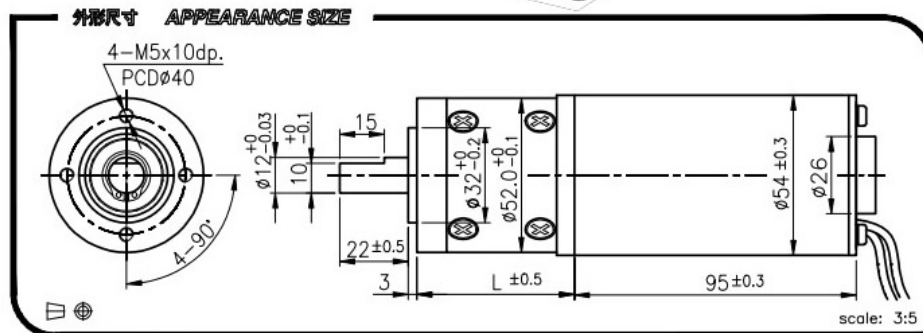
直流馬達 (DC Carbon-brush motors)

IG-52  
GEARED MOTOR  
SERIES

# IG-52GM 03&04 TYPE



REDUCTION RATIO	L	REDUCTION RATIO	L
1/3~1/4	53.0	1/43~1/113	84.0
1/12~1/26	68.5	1/150~1/936	99.5



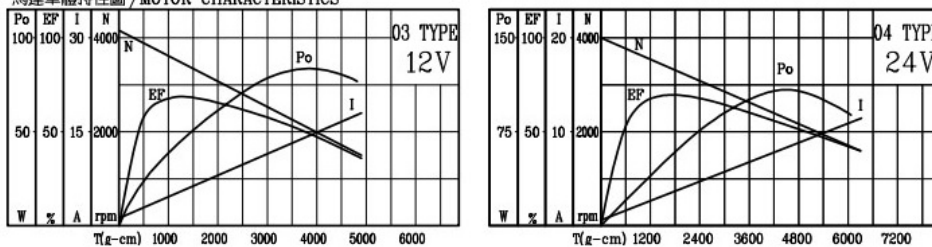
GEARED MOTOR TORQUE/SPEED

減速比 Reduction ratio	1/3	1/4	1/12	1/15	1/19	1/26	1/43	1/53	1/66	1/81	1/100	1/113	1/150	1/230	1/285	1/353	1/488	1/546	1/676	1/936
12V 定格扭力(Kg-cm) Rated torque	2.5	3.1	7.7	9.5	11.8	16	23	28	35	44	54	60	67	100	100	100	100	100	100	100
12V 定格回轉數(rpm) Rated speed	1030	835	295	238	192	139	84	68	55	44	36	32	24	15.5	12.8	10.4	7.6	6.7	5.6	4.0
24V 定格扭力(Kg-cm) Rated torque	3.6	4.5	11	13.5	17	23	33	41	51	62	78	88	97	100	100	100	100	100	100	100
24V 定格回轉數(rpm) Rated speed	1000	815	285	230	185	136	82	67	54	44	35	31	23.5	15.6	12.9	10.5	7.7	6.8	5.7	4.1

馬達單體型式 / MOTOR DATTA

定格電壓 Rated volt (V)	定格扭力 Rated torque (g-cm)	定格回轉數 Rated speed (rpm)	定格電流 Rated current (mA)	無負荷回轉數 No load speed (rpm)	無負荷電流 No load current (mA)	定格出力 Rated output (W)	重量 Weight (g)
12	900	3620	≤ 4100	4000	≤ 1200	33.5	920
24	1300	3550	≤ 2850	4000	≤ 700	48.6	920

馬達單體特性圖 / MOTOR CHARACTERISTICS



## A.1.2. Planetary Gearbox IG52

行星齒輪 (Planetary gear)

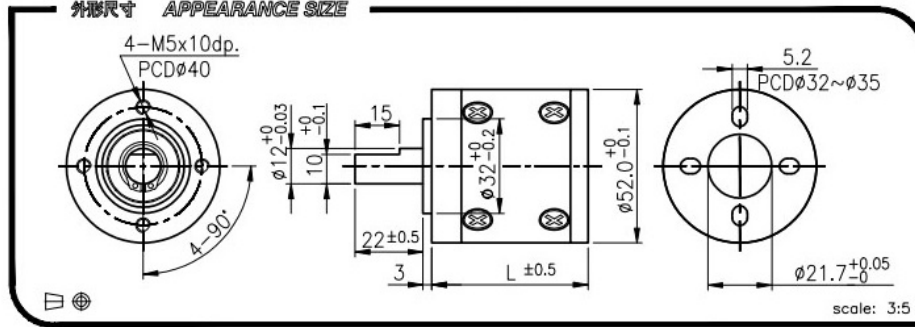
IG-52  
GEARHEAD  
SERIES

# IG-52



外殼材質 Housing material	金屬 metal	背隙 (無負荷時) Backlash at no-load	≤ 3°
輸出軸承型式 Bearing at output	滾珠軸承 Ball bearings		
徑向負荷 Radial load (10mm from flange)	≤ 20 kgf		
軸向負荷 Shaft axial load	≤ 10 kgf		
主軸最大承受壓入力 Shaft press fit force,max.	≤ 30 kgf		
徑向間隙 Radial play of shaft	≤ 0.04 mm		
軸向間隙 Thrust play of shaft	≤ 0.3 mm		

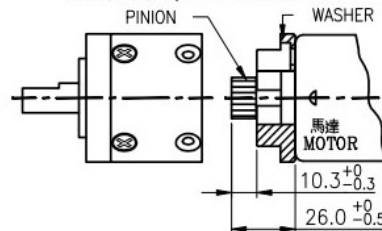
外形尺寸 APPEARANCE SIZE



馬達齒輪型式 / PINION SPECIFICATIONS

模數 Module	1.0		
齒數 No. of teeth	8	12	16
壓力角 Pressure angle	20°		
孔徑 Hole diameter	ø3.98, ø5.98, ø6.34		
減速比 Reduction ratio	1/21, 1/26 1/126, 1/156 1/756, 1/936	1/4, 1/15, 1/19 1/53, 1/66, 1/81 1/113, 1/230, 1/285 1/353, 1/488, 1/546 1/676	1/3 1/12 1/43 1/150

馬達裝卸方法 / MOTOR INSTALLATION



標準減速比型式 / GEARBOXES SPECIFICATIONS

★ 使用相對溼度: 20%~85%RH    ★ 使用溫度範圍: -10°C~+60°C  
Operating relative humidity    Operating temperature range

減速比 Reduction ratio	精確減速比 Exact reduction ratio	定格容許扭力 Rated tolerance torque	瞬間容許扭力 Max momentary tolerance torque	效率 Efficiency	L	重量 (滾珠軸承) Weight (g) Ball bearings
1/3,1/4	3 1/2, 4 1/3	10 kgf-cm Max.	30 kgf-cm	80%	53.0	475
1/12,1/15	12 1/4, 15 1/6	30 kgf-cm Max.	90 kgf-cm	70%	68.5	660
1/19,1/21,1/26	18 7/9, 21, 26	50 kgf-cm Max.	150 kgf-cm	70%	68.5	660
1/43,1/53,1/66	42 7/8, 53 1/12, 65 13/16	100 kgf-cm Max.	300 kgf-cm	60%	84.0	850
1/81,1/100	81 10/27, 100 2/7	100 kgf-cm Max.	300 kgf-cm	60%	84.0	850
1/113,1/126	112 2/3, 126	100 kgf-cm Max.	300 kgf-cm	60%	84.0	850
1/150	150 1/16	100 kgf-cm Max.	300 kgf-cm	50%	99.5	1020
1/230,1/285	230 1/36, 284 43/54	100 kgf-cm Max.	300 kgf-cm	50%	99.5	1020
1/353,1/488	352 49/81, 488 2/9	100 kgf-cm Max.	300 kgf-cm	50%	99.5	1020
1/546,1/676	546, 676	100 kgf-cm Max.	300 kgf-cm	50%	99.5	1020
1/756,1/936	756, 936	100 kgf-cm Max.	300 kgf-cm	50%	99.5	1020

### A.1.3. Gearmotor EC035.120



#### EC035.120 - EC035.240

##### Caratteristiche

Costruzione	Tubolare, senza ventilazione
Grandezza	Ø 52 mm
Potenza	55 W S2 (35 W S1)
Magneti	2
Supporti	Cuscinetti a sfera
Fori di montaggio	4
Alimentazione	Bassa tensione, 12 o 24 Vcc
Spazzole	N° 2 interne di composto grafiterame
Cavo di alimentazione	Lunghezza: 200 mm
Opzioni	Encoder

##### Features

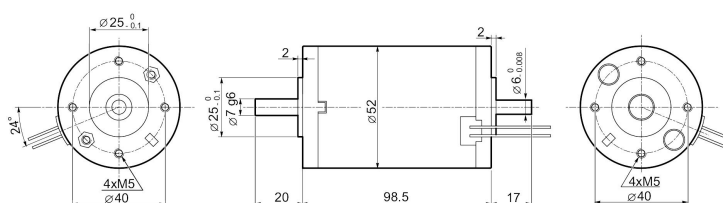
Construction	Tubular, without fan
Size	Ø 52 mm
Power	55 W S2 (35 W S1)
Magnets	2
Bearings	Ball bearings
Mounting holes	4
Power supply	Low voltage, 12 or 24 Vdc
Brushes	2 inside brushes made of graphite/copper composite
Electric cable	Length: 200 mm
Options	Encoder

Tipo Type	S	Pn [W]	V [V]	I [A]	IC	FF	Mn [Nm]	n <sub>1</sub> [min <sup>-1</sup> ]	IP	Kg
EC035.120	S1	35	12	5.2	F	1	0.11	3000	44	0.8
	S2 9'	55		8.0			0.18			
EC035.240	S1	35	24	2.6	F	1	0.11		44	0.8
	S2 9'	55		4.0			0.18			

Azionamenti Drives → [H2](#)

##### Dimensioni

##### Dimensions



Encoder → [I2](#)

Per montaggio encoder serve flangia AS 204  
Encoder assembling needs flange AS 204



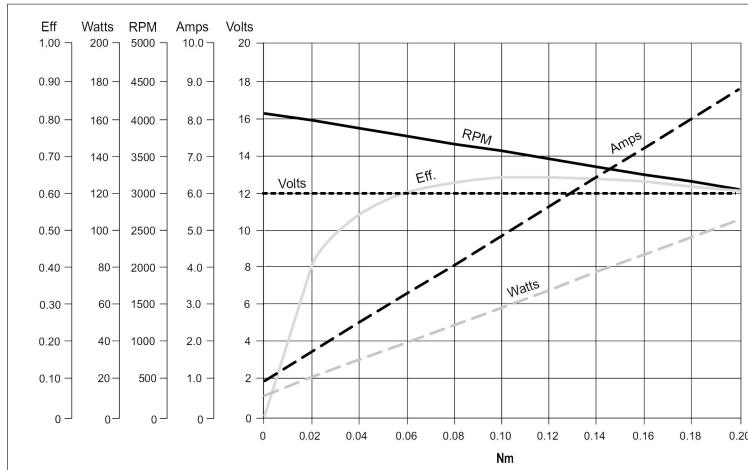
**EC035.120 - EC035.240**

DC

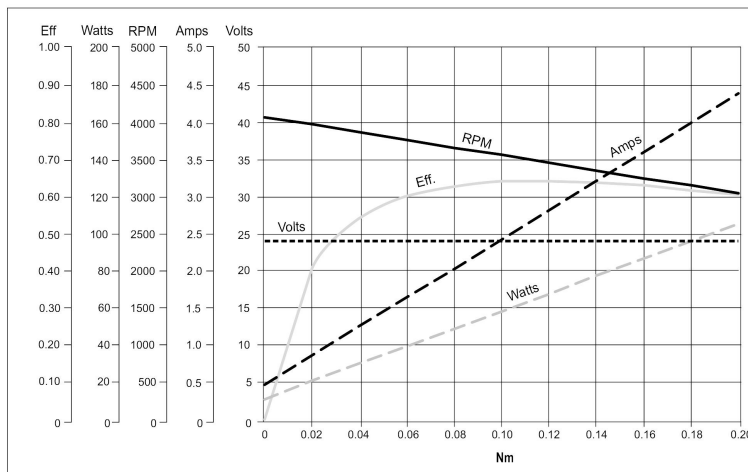
Prestazioni

Performances

**EC035.120**



**EC035.240**



A9

## A.1.4. Planetary Gearbox PK42BB



PK- con motore C.C.

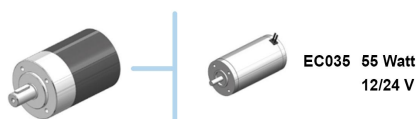
PK- with DC motor



PK42BB								EC035			
Ns	in	ir	M <sub>2</sub> [Nm]	Rd	R <sub>2</sub> [N]	A <sub>2</sub> [N]	Kg	S1 12/24V		S2 12/24V	
								[Nm]	[rpm]	[Nm]	[rpm]
1	4	3.70	2	0.80	160	50	0.4	0.33	946	0.53	811
	7	6.75						0.59	519	0.97	444
	14	13.73						1.13	255	1.85	218
2	25	25.01	5	0.75	230	80	0.5	2.06	140	3.38	120
	46	45.56						3.76	77	6.15	66
	68	68.06						5.24	51	8.58	44
3	93	92.70	10	0.70	300	110	0.6	7.14	38	11.68	32
	169	168.84						13	21	21.27	18
	308	307.54						23.68	11	38.75	10

Nota: le caselle in colore grigio indicano il superamento della coppia massima sopportata dal riduttore per il servizio in S1.  
Per vedere tutti i rapporti di riduzione disponibili, vedere tabella a pag. B3

N.B.: boxes in grey indicate that maximum torque withstood by gear reducer for service in S1 is exceeded.  
See the table on page B3 for all available ratios.



Tipo Type	Servizio Duty	Pn [W]	V [V]	I [A]	IC	FF	Mn [Nm]	n <sub>1</sub> [rpm]	IP	Kg
EC035.120	S1	35	12	5.2	F	1	0.11	3500	44	0.8
	S2 9'	55		8.0			0.18	3000		
EC035.240	S1	35	24	2.6			0.11	3500		
	S2 9'	55		4.0			0.18	3000		

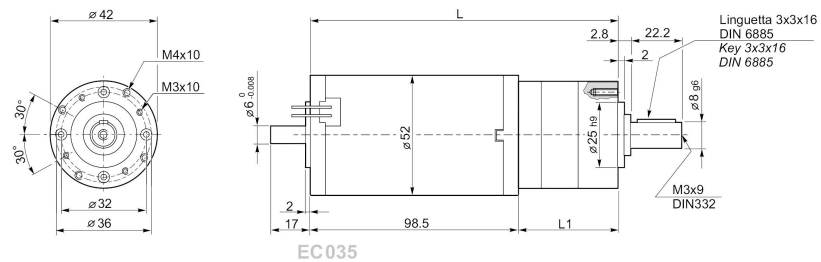


**MICRO MOTORIDUTTORI C.C. EPICICLOIDALI**  
**D.C. MICRO PLANETARY GERMOTORS**

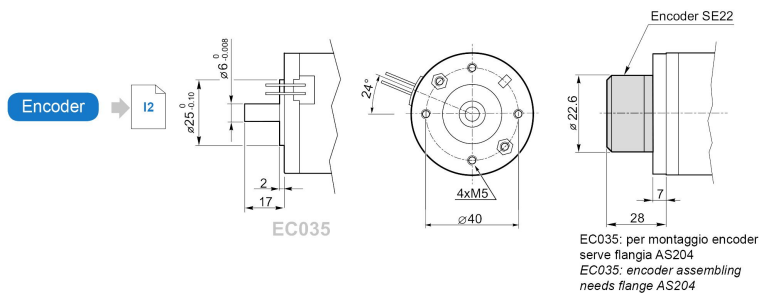


PK- con motore C.C.

PK- with DC motor



DC  
 PK-P-PM



PK42BB	EC035		
	Ns	L1	L
	1	60	158.5
	2	73	171.5
	3	86	184.5

## A.2. Datasheets Encoders

### A.2.1. Two Channel Hall Effect Encoder

# Magnetic Encoders

- 雙向霍爾作用譯碼器  
Two Channel Hall Effect Encoder
- 單向霍爾作用譯碼器  
One Channel Hall Effect Encoder

★ 使用相對溼度: 20%~85%RH    ★ 使用溫度範圍: -10°C~+60°C  
Operating relative humidity    Operating temperature range

**外形尺寸 APPEARANCE SIZE**

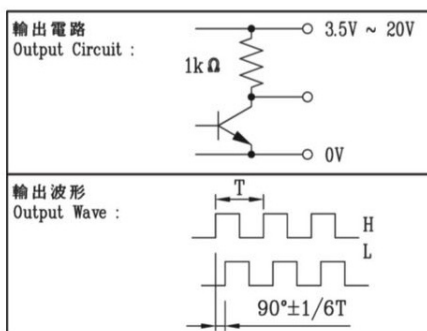
φA	L	COUNTS POLES OF PER TURN (PPR)	
		current	limit.
★ φ12	★ 6.5	2, 6 (1, 3)	6 (3)
★ φ16	★ 6.5	2, 6 (1, 3)	6 (3)
★ φ20	★ 8.5	2, 6 (1, 3)	6 (3)
φ27.3	12.6	2, 6, 14, 26 (1, 3, 7, 13)	26 (13)
φ32.3	14.3	14, 26 (7, 13)	26 (13)
φ35.3	13.5	14, 26 (7, 13)	26 (13)
φ42.5	15.5	2, 10, 38 (1, 5, 19)	38 (19)

■ UL1061 AWG26  
 UL1007 AWG26  
 suggested connectors  
 JST PHR-6 AMP 175788  
 P=2.0-6P Molex 51065  
 JST PH  
 ■ JST ZHR-6 JST ZH  
 P=1.5-6P  
 ★ WITHOUT CAP

#### 電氣特性

#### ELECTRICAL CHARACTERISTICS

規格特性 CHARACTERISTICS	代號 SYMBOL	測試條件 TEST CONDITIONS	極小 MIN.	基準 REF.	最大 MAX.	單位 UNITS
輸入電壓 Supply Voltage	Vcc	---	3.5	-	20	V
輸出飽和電壓 Output Saturation Voltage	Vce(sat)	Vcc=14V ; Ic=20mA	-	300	700	mV
輸出漏電流 Output Leakage Current	Icex	Vce=14V ; Vcc=14V	-	< 0.1	10	μV
輸入電流 Supply Current	Ice	Vcc=20V Output open	-	5	10	mA
輸出上升時間 Output Rise Time	tr	Vcc=14V ; RL=820Ω ; CL=20pF	-	0.3	1.5	μS
輸出下降時間 Output Fall Time	tf	Vcc=14V ; RL=820Ω ; CL=20pF	-	0.3	1.5	μS



Two Channel Encoder	
Connections :	
1. Black	: -MOTOR
2. Red	: +MOTOR
3. Brown	: HALL SENSOR Vcc
4. Green	: HALL SENSOR GND
5. Blue	: HALL SENSOR A Vout
6. Purple	: HALL SENSOR B Vout
One Channel Encoder	
Connections :	
1. Black	: -MOTOR
2. Red	: +MOTOR
3. Brown	: HALL SENSOR Vcc
4. Green	: HALL SENSOR GND
5. Blue	: HALL SENSOR A Vout
6. Purple	: EMPTY

## A.2.2. Encoder AEDM-5810

### AEDM-5xxx Quick Assembly High Resolution Three Channel Optical Encoders



### Data Sheet



#### Description

Avago Technologies' AEDM-5xxx series encoders are high resolution and high performance three channel optical incremental encoders. These encoders emphasize on high reliability, high resolution, and easy assembly.

Each encoder contains a LED source with lens, an ASIC with integrated photo detector arrays, LED current regulation, interpolator and output driver circuitry, and a code wheel which rotates between the emitter and detector ASIC. The outputs of the AEDM-5xxx are two square waves in quadrature. In addition, a third channel offers index signal, which is a 90 electrical degree, true index pulse which is generated once for each full rotation of the code wheel. The AEDM-5xxx series utilizes a film code wheel allowing for resolutions up to 5000 CPR.

These bearing-less housed encoders may be quickly and easily mounted to a motor. For larger diameter motors, the AEDM-59xx series feature external mounting ears option.

The quadrature signals and the index pulse are accessed through five 0.025 inch square pins located on 0.1 inch centers.

Standard resolutions between 2000 and 5000 counts per revolution are presently available. Consult local Avago sales representatives for other resolutions.

#### Features

- Two channel quadrature output with index pulse
- High Resolution - up to 5000 CPR
- Operating temperature – 40°C to +85°C
- Quick and easy assembly
- No signal adjustment required
- Cost Effective solution
- Small size
- TTL compatible output
- Single 5V supply with  $\pm 10\%$  tolerance
- External mounting ears option available for larger motors.

---

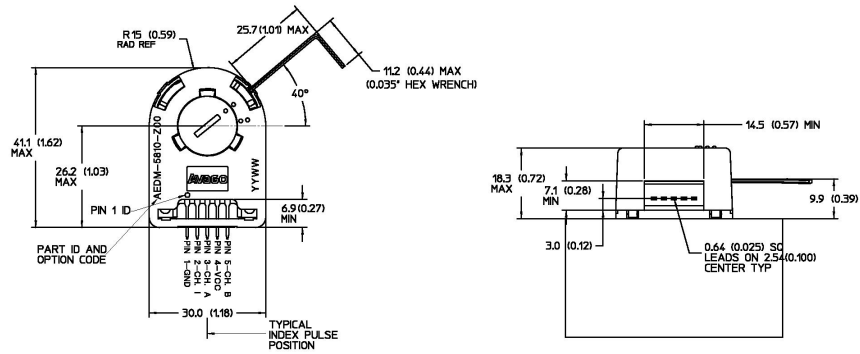
Note: Avago Technologies encoders are not recommended for use in safety critical applications such as. ABS braking systems, power steering, life support systems and critical care medical equipment. Please contact our sales representative if clarifications are needed.

---

**CAUTION:** It is advised that normal static precautions be taken in handling and assembly of this component to prevent damage and/or degradation which may be induced by ESD.

## Package dimensions

### AEDM-58xx

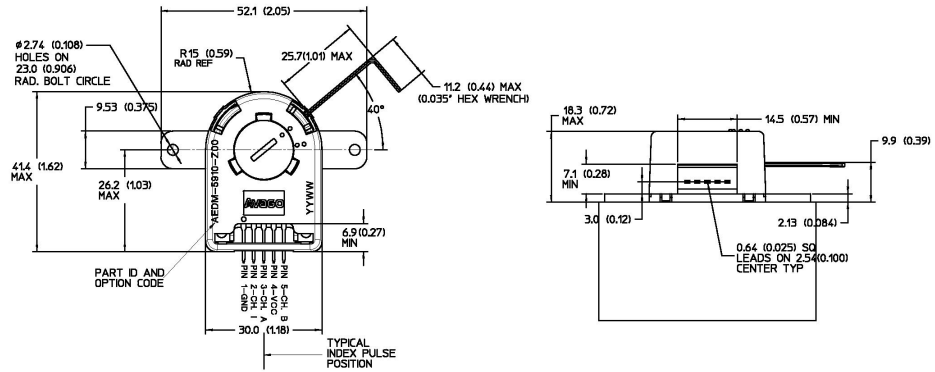


#### NOTES:

1. LEAVE CLEARANCE FOR TURNING AND REMOVING THE HEX WRENCH.
2. TYPICAL DIMENSIONS IN MILLIMETERS (INCHES).

\*Note: For the AEDM-58XX Pin #2 is CH. I, the index output.

### AEDM-59xx



#### NOTES:

1. LEAVE CLEARANCE FOR TURNING AND REMOVING THE HEX WRENCH.
2. TYPICAL DIMENSIONS IN MILLIMETERS (INCHES).

\*Note: For the AEDM-59XX Pin #2 is CH. I, the index output.

### Absolute Maximum Ratings

Parameter	Symbol	Minimum	Maximum	Units
Storage Temperature	T <sub>s</sub>	-40	+85	°C
Operating Temperature	T <sub>A</sub>	-40	+85	°C
Supply Voltage	V <sub>CC</sub>	-0.5	7	V
Output Voltage	V <sub>O</sub>	-0.5	V <sub>CC</sub>	V
Output Current per Channel	I <sub>OUT</sub>		±5	mA
Velocity			12,000**	RPM

Note \*\*: Velocity 12000 RPM applies to Encoder Resolution of 5000 CPR

### Recommended Operating Conditions

Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
Temperature	T <sub>A</sub>	-40		+85	°C	
Supply Voltage	V <sub>CC</sub>	4.5	5.0	5.5	Volt	Ripple < 100 mVp-p
Load Capacitance	C <sub>L</sub>			100	pF	
Output Frequency	f			500	kHz	2000 – 2500 CPR
				1,000	kHz	4000 – 5000 CPR
Shaft Perpendicularity Plus Axial Displacement				± 0.15 (± 0.006)	mm (in.)	
Spatial play (including shaft eccentricity)				Ø0.40 (0.016)	mm (in.)	

### Encoding Characteristics

Encoding characteristics are applicable per the Recommended Operating Conditions, unless otherwise specified. Values are for the worst case error over the full rotation.

#### Encoder Resolution ≥ 4000 CPR

Parameter	Symbol	Min	Typ.*	Max	Units
Pulse Width Error	ΔP		21	45	°e
Logic State Width Error	ΔS		17	45	°e
Phase Error	Δφ		12	25	°e
Cycle Error	ΔC		36	60	°e
Index Pulse Width	P <sub>0</sub>	65	90	115	°e

#### Encoder Resolution ≤ 2500 CPR

Parameter	Symbol	Min	Typ.*	Max	Units
Pulse Width Error	ΔP		20	30	°e
Logic State Width Error	ΔS		15	25	°e
Phase Error	Δφ		10	15	°e
Cycle Error	ΔC		17	30	°e
Index Pulse Width	P <sub>0</sub>	72	90	108	°e

\* Typical values specified at V<sub>CC</sub> = 5.0V and 25°C and for single ended, unless otherwise stated

## Electrical Characteristics

Electrical Characteristics over Recommended Operating Range

Parameter	Symbol	Min.	Typ.*	Max.	Units	Notes
Supply Current	$I_{CC}$		25	80	mA	No load
High Level Output Voltage	$V_{OH}$	2.4			V	$I_{OH} = -5$ mA
Low Level Output Voltage	$V_{OL}$			0.4	V	$I_{OL} = 5$ mA.
Rise Time	$t_r$		10		ns	$C_L = 100$ pF
Fall Time	$t_f$		10		ns	$C_L = 100$ pF

\*Typical values specified at  $V_{CC} = 5.0$  V and 25°C.

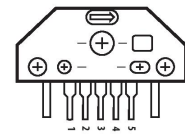
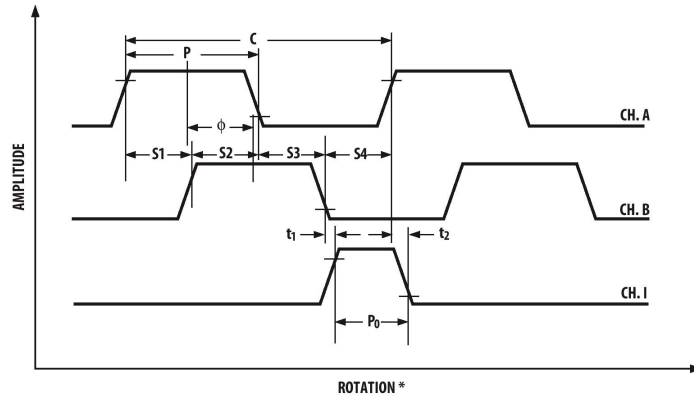
## Mechanical Characteristics

Parameter	Symbol	Dimension	Tolerance <sup>[1]</sup>	Units
Shaft Diameters		Per ordering information	g6	mm (in.)
Moment of Inertia	J	0.6 (8.0 x 10 <sup>-6</sup> )		g-cm <sup>2</sup> (oz-in-s <sup>2</sup> )
Required Shaft Length <sup>[2]</sup>		14.0 (0.55)	± 0.5 (± 0.02)	mm (in.)
Bolt Circle <sup>[3]</sup>	2 screw mounting	19.05 (0.750)	± 0.13 (± 0.005)	mm (in.)
	3 screw mounting	20.90 (0.823)	± 0.13 (± 0.005)	mm (in.)
	external mounting ears	46.0 (1.811)	± 0.13 (± 0.005)	mm (in.)
Mounting Screw Size <sup>[4]</sup>	2 screw mounting	M 2.5 or (2-56)		mm (in.)
	3 screw mounting	M 1.6 or (0-80)		mm (in.)
	external mounting ears	M 2.5 or (2-56)		mm (in.)
Encoder Base Plate Thickness		0.33 (0.130)		mm (in.)
Hub Set Screw		(2-56)		(in.)

### Notes:

1. These are tolerances required of the user.
2. The AEDM-5805, 5905 provide an 8.9 mm (0.35 inch) diameter hole through the housing for longer motor shafts. See Ordering Information.
3. The recommended mounting screw torque for 2 screw and external ear mounting is 1.0 kg-cm (0.88 in-lbs). The recommended mounting screw torque for 3 screw mounting is 0.50 kg-cm (0.43 in-lbs).

**Output Waveforms**



**\* SPIN DIRECTION FOLLOWS ARROW MARKING ON ENCODER**

Direction of Rotation: When the codewheel rotates in the counter-clockwise direction (as viewed from the encoder end of the motor), channel A will lead channel B. If the codewheel rotates in the clockwise direction, channel B will lead channel A.

### Mounting Considerations

The AEDM-5xxx Series high resolution encoders must be aligned using the aligning pins as specified in Figure 2, or using the HEDS-8910 Alignment Tool as shown in Encoder Mounting and Assembly.

The AEDM-58xx can be mounted to a motor using either the two screw or three screw mounting option as shown in Figure 1. The optional aligning pins shown in Figure 2 can be used with either mounting option

The AEDM-59xx have external mounting ears which may be used for mounting to larger motor base plates. Figure 3 shows the necessary mounting holes with optional aligning pins and motor boss.

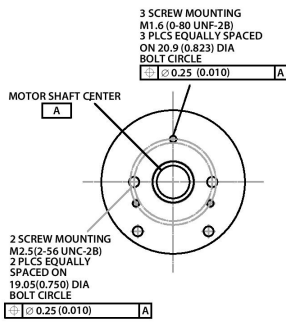


Figure 1. Mounting Holes

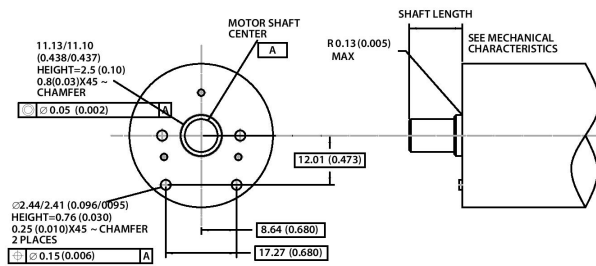


Figure 2. Optional Mounting Aids

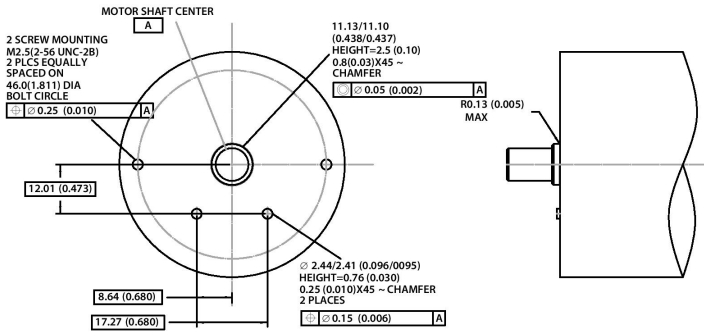


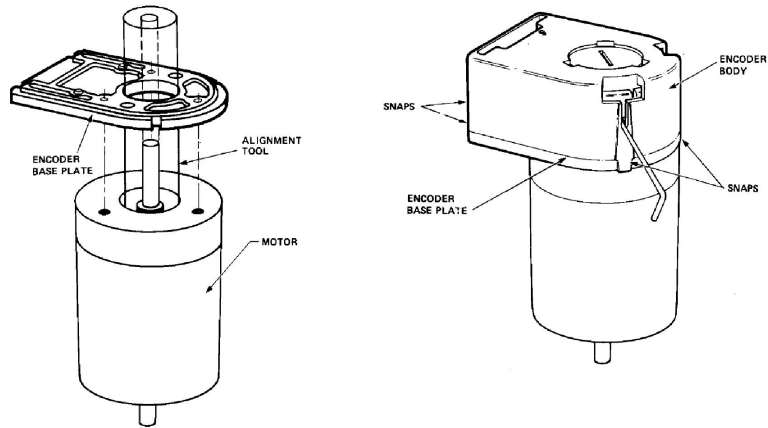
Figure 3. Mounting with external ears

Index Pulse Width (PO): The number of electrical degrees that an index output is high during one full shaft rotation. This value is nominally 90°e or 1/4 cycle.

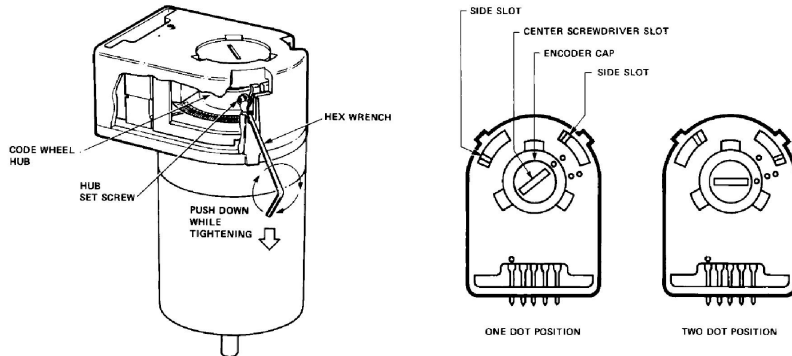
## Encoder Mounting and Assembly

### Assembly process for AEDM-58xx and AEDM-59xx

1. Slide the alignment tool onto the motor shaft. With alignment tool in place, mount encoder baseplate onto motor as shown above. Ensure that the holes in the baseplate align with the mounting holes in the motor body. Inset and tighten mounting screws\* and remove alignment tool.
2. Snap encoder body onto base plate locking all 4 snaps.



- 3a. Inset the hex wrench into the body of the encoder. Ensure that it is properly seated into the code wheel hub set screws. Then apply a downward force on the end of the hex wrench. This sets the code wheel gap by levering the code wheel hub to its upper position.
- 3b. While continuing to apply a downward force, rotate the hex wrench in the clockwise direction until the hub set screw is tight\*\* against the motor shaft. The hub set screw attaches the code wheel to the motor's shaft.
- 3c. Remove the hex wrench by pulling it straight out of the encoder body.



4. Use the center screwdriver slot, or either of the two side slots, to rotate the encoder cap dot clockwise from the one dot position to the two dot position. Do not rotate the encoder cap counterclockwise beyond the one dot position. The encoder is ready for use!

\* Adhesive should be applied to the screw-baseplate interface to prevent screws loosening due to effect of vibration and high temperature on plastic. High temperature rated adhesive such as HERNON Ultra 317 is recommended.

\*\* The recommended torque to tighten the setscrew is 15-18 ozf.inch.

## Connectors

### HEDS-890x connector dimensions

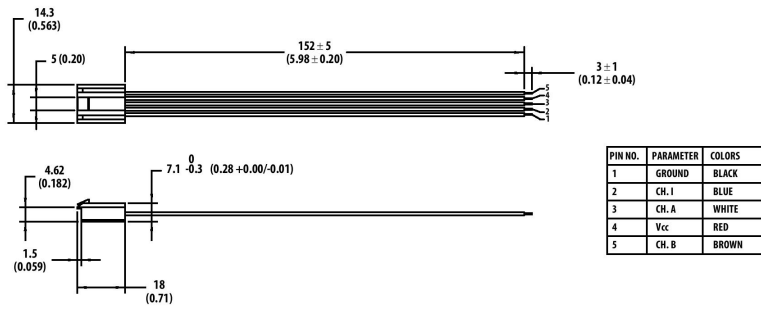


Figure 4. HEDS-8902 and HEDS-8903 Connectors

## Connectors

Manufacturer	Part Number
AMP	103686-4
	640442-5
Avago *	HEDS-8902 (2 ch.) with 4-wire leads
	HEDS-8903 (3 ch.) with 5-wire leads
Molex	2695 series with 2759 series term.

\* Avago connectors are designed to mechanically lock into the AEDM-5XXX Series.

### A.3. Esquema de Ligações Encoders - RoboteQ SCD

**How to wire encoders to RoboteQ SDC**

Connector Pin	Power	Dout	Com	RC	Ana	DInput	Enc	Default Config
1		DOUT1						Unused
2		DOUT2						Unused
3			TKOut					Unused
4				RC5	ANA5 (1)(2)	DIN5	ENCZA	RS232Tx Data
5								Unused
6			Rin					RS232Rx Data
7				RC4	ANA4	DIN4		Unused
8				RC1	ANA1 (1)(2)	DIN1	ENC1A	RCRadio1
9				RC3	ANA3	DIN3		Unused
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								
32								
33								
34								
35								
36								
37								
38								
39								
40								
41								
42								
43								
44								
45								
46								
47								
48								
49								
50								
51								
52								
53								
54								
55								
56								
57								
58								
59								
60								
61								
62								
63								
64								
65								
66								
67								
68								
69								
70								
71								
72								
73								
74								
75								
76								
77								
78								
79								
80								
81								
82								
83								
84								
85								
86								
87								
88								
89								
90								
91								
92								
93								
94								
95								
96								
97								
98								
99								
100								

DB15 Connector  
TE-104-631

Note: In Roborun+ Software if one of the motors is counting backward invert A and B in the motor that is running backward.




Note: if you are using optical encoders no need of pull up board

SuperDroid Robots, Inc

Wiring RoboteQ SDC Motors and encoders





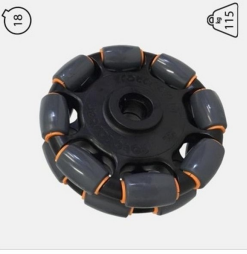

PAGE 1 OF 2

Anexo B. Resultados do  
Estudo das Rodas  
Omnidirecionais Universais

			
<b>Nome</b>	4" DuraOmni Wheel (am-3047)	6" DuraOmni Wheel (am-3154)	8" Duale Plastic Omni Wheel (am-0463)
<b>Fabricante</b>	AndyMark	AndyMark	AndyMark
<b>Diametro</b>	4" (101.6 mm)	6" (152.4 mm)	8" (203.2 mm)
<b>Material do Corpo</b>	Policarbonato	Policarbonato	Policarbonato
<b>Capacidade de carga</b>	54.4 Kg	54.4 Kg	54.4 Kg
<b>Peso</b>	0.26 Kg	0.45 Kg	0.98 Kg
<b>Nº de Rolos</b>	8	10	36
<b>Material dos Rolos</b>	TPU (Poliuretano Termoplástico)	TPU (Poliuretano Termoplástico)	SBR (Estireno Butadieno)
<b>Durometro</b>	77A	77A	80A
<b>Preço</b>	24.00\$	35.00\$	64.00\$
<b>Escala de dureza dos materiais</b>			
<b>Tipos de material dos rolos</b>	<p>SBR (Estireno Butadieno)</p> <p>Poliuretano</p> <p>Polipropileno</p> <p>Nylon</p>		
<b>Características</b>	<p>Menor preço e maior volume de elastômeros disponível</p> <p>Ligeiramente mais resistente ao calor quebra por flexão</p> <p>Pouca resistência à chama</p> <p>Faixa de temperatura: -18° C a 107°C</p> <p>Existem duas classes - poliéster e poliéter</p> <p>Alta resistência à tração e à abrasão (desgaste por atrito)</p> <p>Ao contrário do poliéter, o poliéster pode ser afetado pela água quente e alta umidade</p> <p>Faixa de temperatura: -40° C a 82°C</p> <p>Alta resistência à fratura por fadiga ou fadiga</p> <p>Boa resistência ao impacto acima de 15 °C</p> <p>Boa absorção de umidade</p> <p>Boa resistência à tração</p> <p>Alta resistência à fadiga</p> <p>Boa resistência ao impacto</p> <p>Baixo coeficiente de atrito</p> <p>Alta temperatura de fusão</p>		

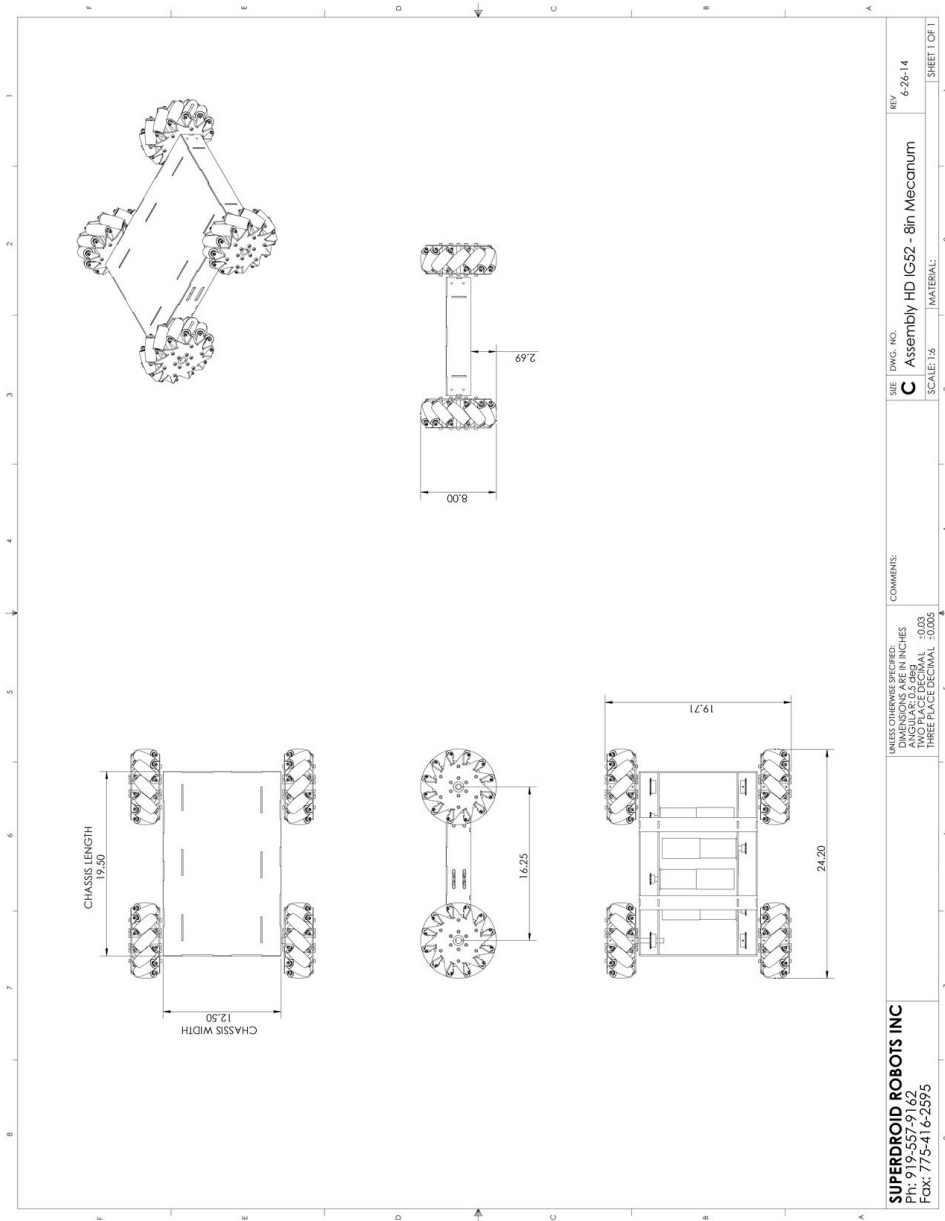
3060 Angram Drive | Eden Prairie, MN 55441 | Phone: 952-934-2301 | Toll Free: 888-776-7761 | www.plasticsintl.com

			
100mm Double Aluminum Omni Wheel w/Bearing Rollers	127mm Double Aluminum Omni Wheel	4 inch Nexus Robot Double Plastic Omni Wheel	8" Aluminum Omni Wheel
Active Robots	Active Robots	Nexus Robot	AndyMark
100 mm	127mm	100 mm	8" (203.2 mm)
Nylon	Liga de Aluminio	Nylon	Aluminio
30 Kg	30 Kg	20 Kg	50 pound (22.68 kg)
0.335 Kg	0.480 Kg	0.290kg	0.431 kg
18	22	18	18
Borracha	Borracha	Borracha	Borracha
?	?	?	?
53.34€ (62.98€)	45.00€ (53.13€)	28.80\$	56.00\$
			
152mm Omnidirectional Wheel (Brass Bearing for Rollers)	127mm Heavy Duty Aluminum Omni Wheel w/ Bearing Rollers	203mm Double Aluminum Omni Directional Wheel w/ Bearing Rollers	
Nexus Robot	Nexus Robot	Nexus Robots	
152 mm	5" (127 mm)	8" (203 mm)	
Liga de Aluminio	Liga de Aluminio	Liga de Aluminio	
40 kg	80 kg	50 kg	
0.630 Kg	0.860 kg	1.3 kg	
28	10	36	
Borracha	Nylon	Borracha	
?	?	?	
70.40\$	120.00\$	135.00\$	

			
100MM-ALUMINIUM-OMNI-WHEEL-WITH-PU-14182Z	203mm Aluminium Omni wheel with PU roller(80kg Load)14194	Transwhheel FXA457 (4202KX) CAT-TRAK	Transwhheel FXA457 (4202KXU) CAT-TRAK
Nexus Robots	Nexus Robots	Komyjak	Komyjak
100 mm	203 mm	4" (101.6 mm)	4" (101.6 mm)
Liga de Alumínio	Liga de Alumínio	Plástico de alta resistencia	Plástico de alta resistencia
80 kg	80 kg	100 pounds (45.36kg)	100 pounds (45.36kg)
0.816 kg	4.5 kg	0.312 kg	0.312 kg
6	10	16	16
Poliuretano	Poliuretano Revestido	Polipropileno revestido com borracha sintética	Ureiano em eixos de aço inoxidável
?	?	?	?
188.00€	259.00€	24.84\$	35.29\$
			
Rotacaster 125mm Double Polyurethane Roller / Shielded Bearing	Rotacaster 125mm Triple Polyurethane Roller / Sealed Bearing 10mm		
Rotacaster	Rotacaster		
125 mm	125 mm		
Polimeros	Polimeros		
115kg	130kg		
0.315 kg	0.315 kg		
16	24		
Poliuretano	Poliuretano		
85A	85A		
34.50\$	46.05\$		

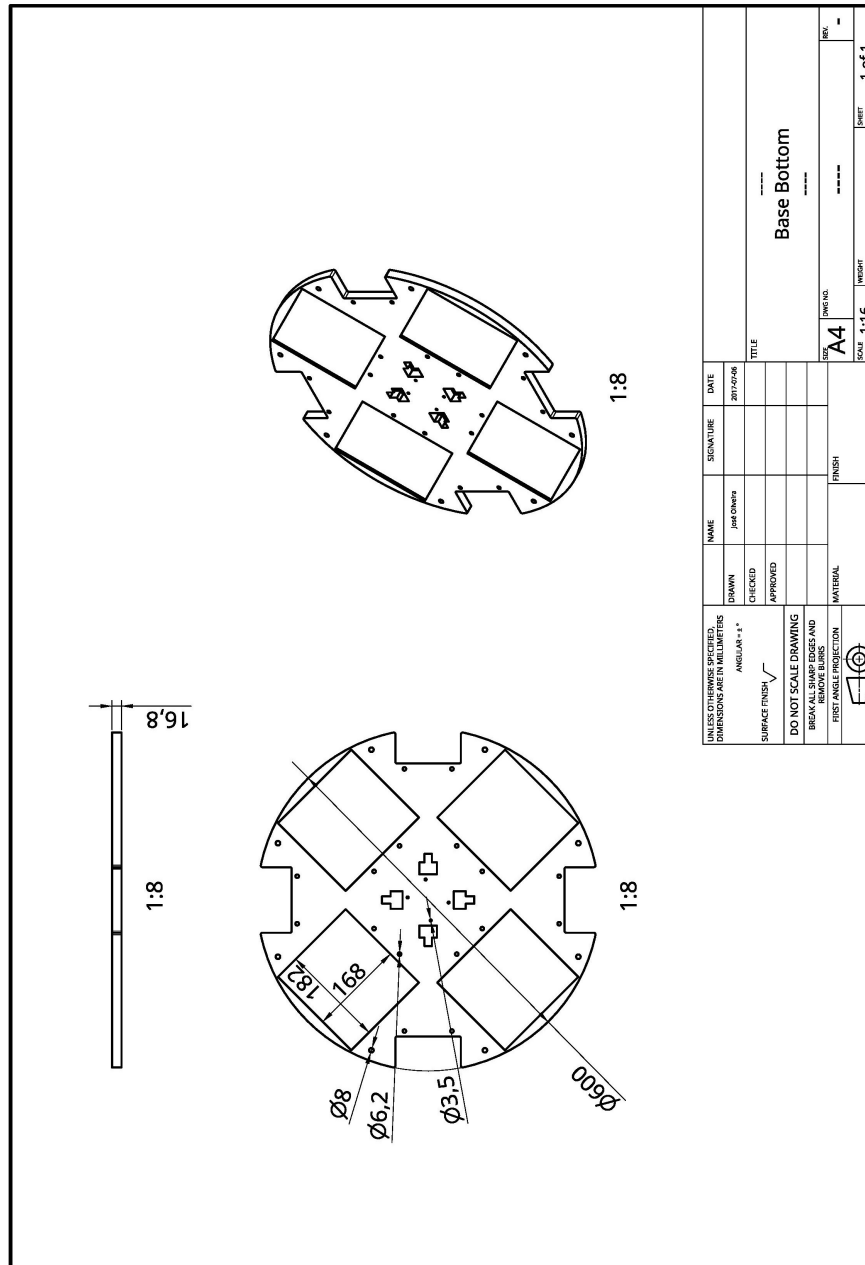
## Anexo C. Desenhos Técnicos

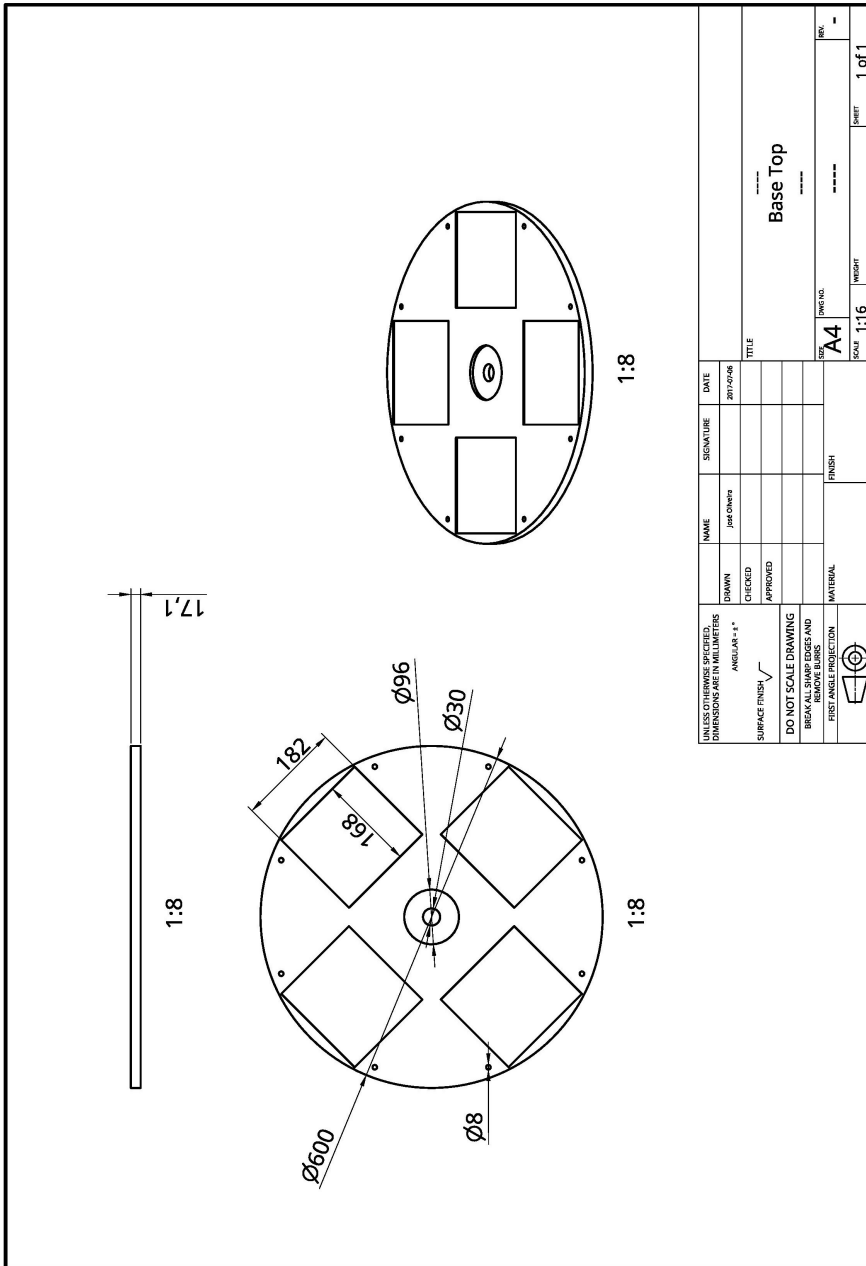
# C.1. Desenhos Técnicos da Base Omnidirecional com Rodas Mecanum

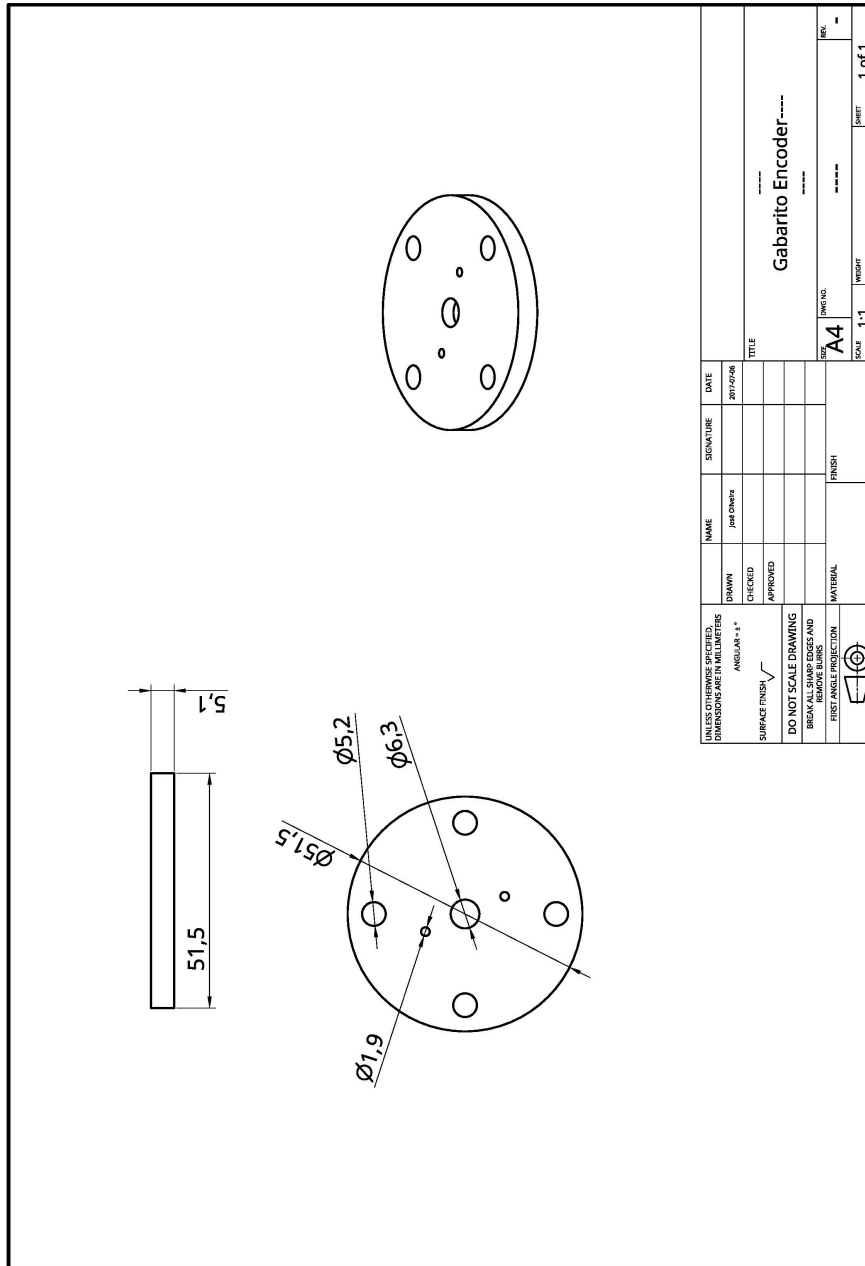




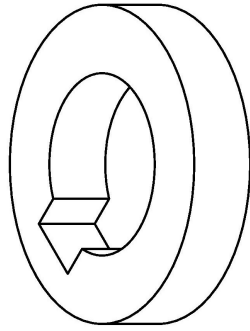
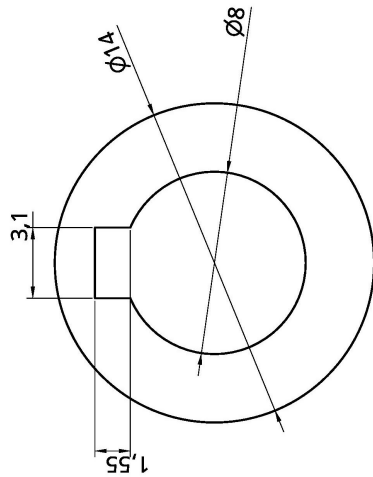
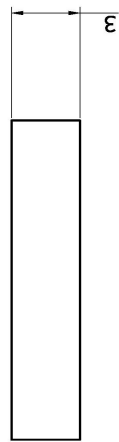
## C.2. Desenhos Técnicos da Base Omnidirecional com Rodas Universais Desenvolvida



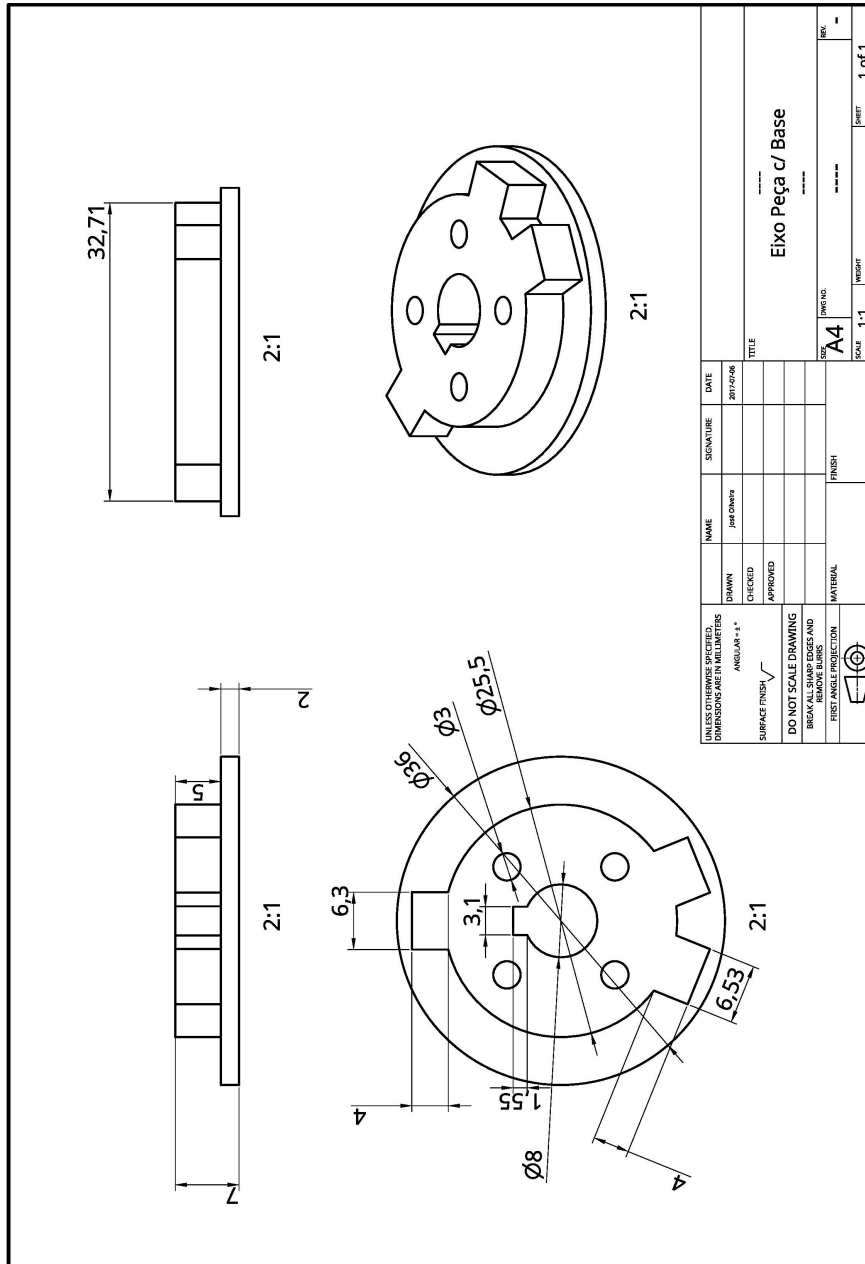


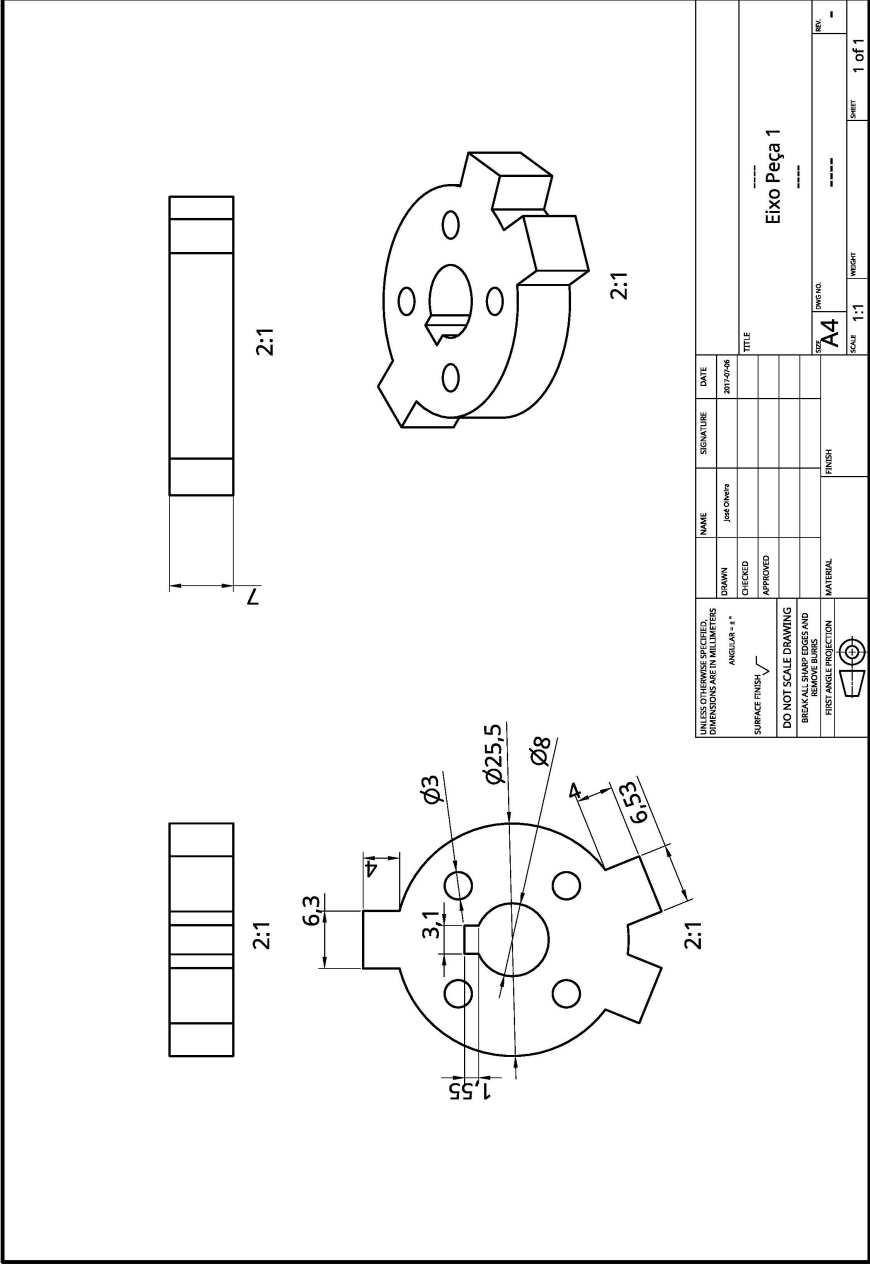


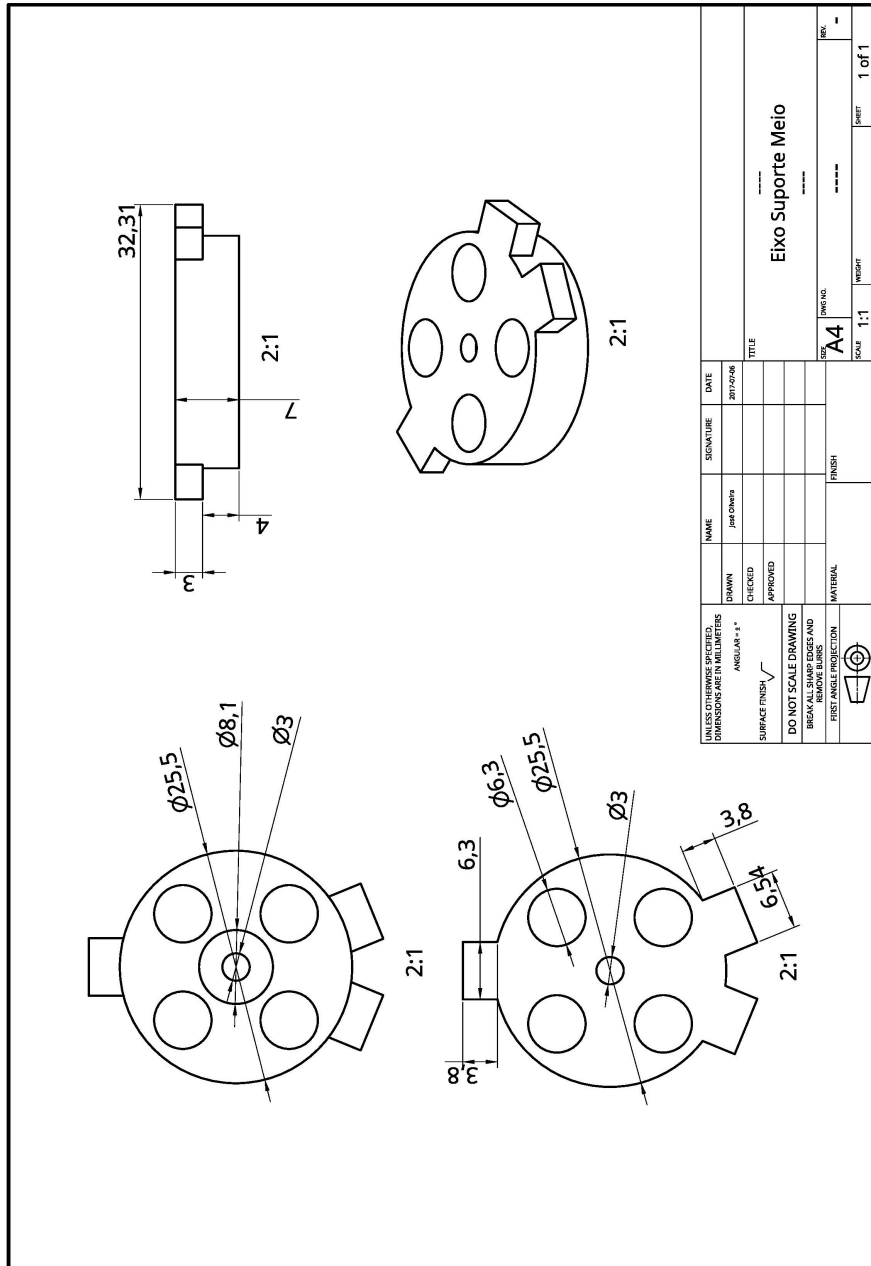
UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN MILLIMETERS		NAME	SIGNATURE	DATE
ANGULAR ±		jos chaves		2017/04
SURFACE FINISH				
DO NOT SCALE DRAWING				
BREAK DIMENSIONS AND REMOVE DIMS				
FIRST ANGLE PROJECTION				
MATERIAL		FINISH		SIZE
				A4
				SCALE
				1:1
				WEIGHT
				SHEET
				1 of 1
TITLE				
Gabarito Encoder				
REV				
-				

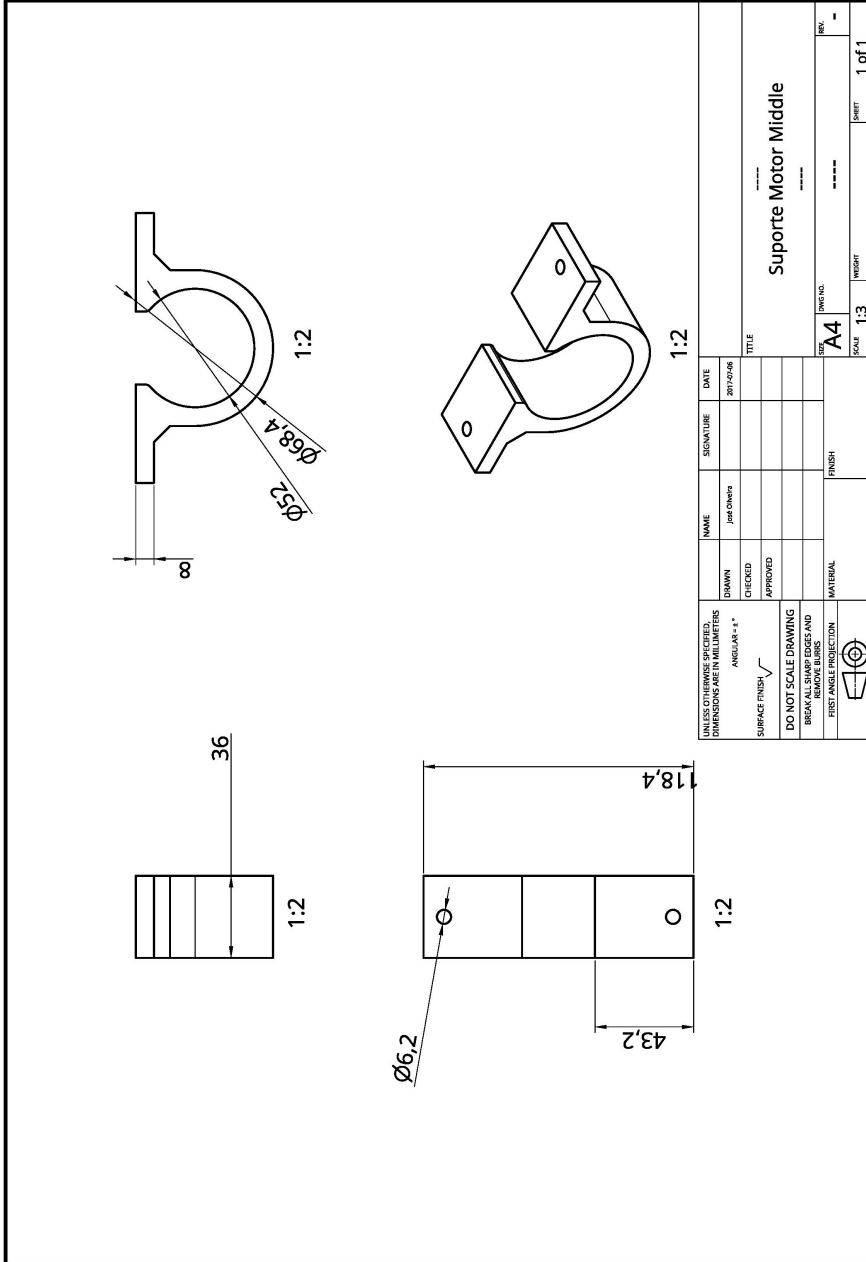


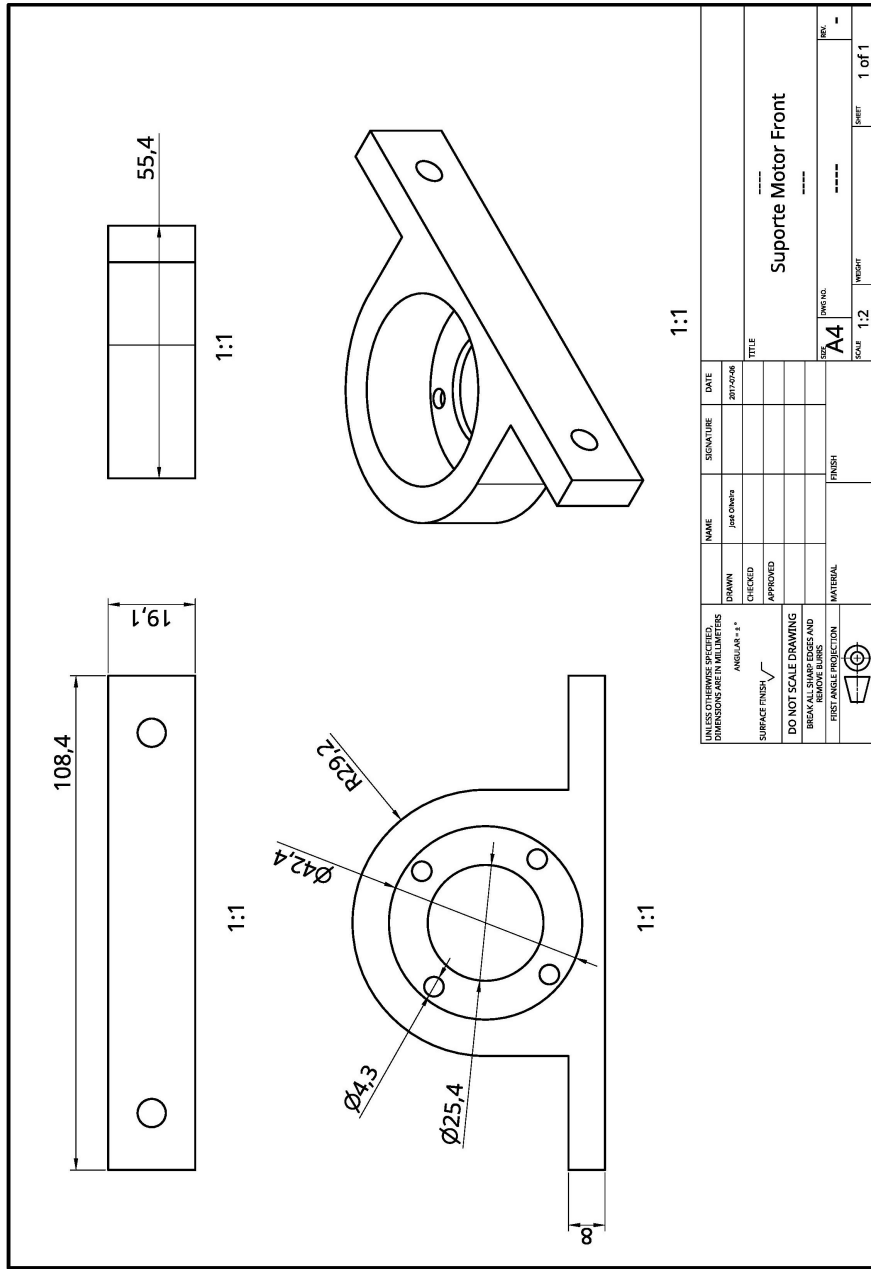
UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN MILLIMETERS		NAME	SIGNATURE	DATE
ANGULARS - 1°		jos chaves		2017/04/04
SURFACE FINISH				
DO NOT SCALE DRAWING		CHECKED		
BREAKS, DIMENSIONS AND HATCHES		APPROVED		
FIRST ANGLE PROJECTION				
MATERIAL				
FINISH				
SCALE		5:1		WEIGHT
FIRST ANGLE PROJECTION		A4		SHEET
TITLE		Eixo Espaçador		1 of 1





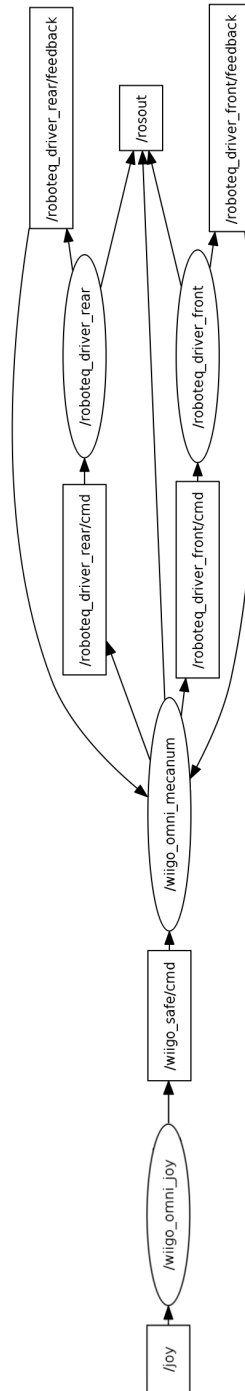




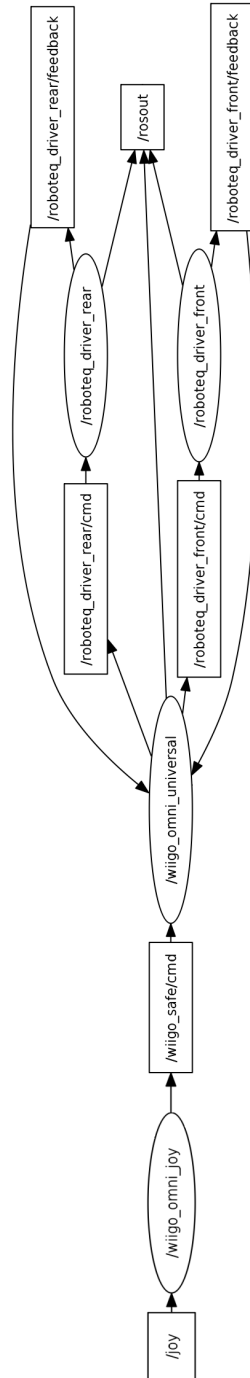


# Anexo D. Diagramas do *Software* desenvolvido em ROS

## D.1. Diagrama de *Software* ROS da Plataforma com Rodas *Mecanum*



## D.2. Diagrama de *Software* ROS da Plataforma com Rodas Universais



Esta página foi intencionalmente deixada em branco.

# Anexo E. Resultados dos Testes de Odometria



# E.1. Resultados dos Testes Realizados com a Base de Rodas *Mecanum*

## E.1.1. Ensaios de Deslocamento de 1 m

Teste 1m - menos atrito Vel = 0.5 m/s --> 2sec		DATA: 22/03/17									
Sentido	Odometria (m)	Deslocamento Real (m)		Diferença Odometria/Real		Média Odometria (m)		Média Deslocamento (m)		Diferença Deslocamento Expectativa e Deslocamento médio (m)	
x (+ positivo)	1.0123	1.0530	0.0607	Δx  = 1.040	Δx  = 1.083	0.042	0.083	1.045	1.048	0.043	0.045
	1.0466	1.0660	0.0564								
	1.0494	1.0600	0.0306								
	1.0466	1.0710	0.0214								
	-1.0358	1.0530	0.0172								
x (- negativo)	-1.0515	1.0510	0.0005	Δx  = 1.013	Δx  = 1.055	0.042	0.055	1.048	1.048	0.086	0.092
	-0.9720	1.0510	0.0790								
	-0.9927	1.0650	0.0723								
	1.0459	1.0150	0.0309								
	1.0507	1.0200	0.0307								
y (+ positivo)	1.0236	1.0200	0.0036	Δy  = 1.026	Δy  = 1.018	0.024	0.018	0.938	0.938	0.086	0.092
	0.9841	1.0160	0.0309								
	-1.0582	1.0000	0.0486								
	-1.0254	1.0000	0.0767								
	-1.0501	1.0000	0.0500								
y (- negativo)	-1.0256	1.0150	0.0106	Δy  = 1.037	Δy  = 1.079	0.047	0.079	0.943	0.943	0.0763	0.057
Teste 1m Vel = 0.5 m/s --> 2sec		DATA: 22/03/17									
Sentido	Odometria (m)	Deslocamento Real (m)		Diferença Odometria/Real		Média Odometria (m)		Média Deslocamento (m)		Diferença Deslocamento Expectativa e Deslocamento médio (m)	
x (+ positivo)	1.0500	1.0440	0.0060	Δx  = 1.034	Δx  = 1.045	0.018	0.045	1.045	1.048	0.032	0.048
	1.0497	1.0450	0.0047								
	0.9857	1.0450	0.0593								
	1.0466	1.0460	0.0036								
	-0.9590	1.0460	0.0890								
x (- negativo)	-1.0237	1.0450	0.0214	Δx  = 1.016	Δx  = 1.048	0.032	0.048	1.048	1.048	0.086	0.092
	-1.0518	1.0520	0.0002								
	-1.0388	1.0460	0.0074								
	1.0495	0.9350	0.1145								
	1.0501	0.9350	0.1151								
y (+ positivo)	0.9917	0.9370	0.0547	Δy  = 1.024	Δy  = 0.938	0.086	0.092	0.938	0.938	0.086	0.092
	1.0054	0.9460	0.0594								
	-0.9175	0.9440	0.0265								
	-0.9497	0.9340	0.0157								
	-1.0233	0.9470	0.0763								
y (- negativo)	-1.0128	0.9460	0.0669	Δy  = 0.976	Δy  = 0.943	0.046	0.057	0.943	0.943	0.0763	0.057

## E.1.2. Ensaios de Deslocamento de 4 m

Teste 4m Vel = 0.5 m/s --> 8sec		DATA: 22/03/17									
Sentido	Odometria (m)	Deslocamento Real (m)	Diferença Odometria/Real	Média Odometria (m)		Média Deslocamento (m)		Média Diferença Odometria/Real		Diferença Deslocamento Esperado e Deslocamento médio (m)	
				Δx  =	Δy  =	Δx  =	Δy  =				
x (+ positivo)	4.1888	4.2020	0.0133			4.205		0.089		0.205	
	4.1200	4.2060	0.0860								
	4.0765	4.2040	0.1275								
	4.0753	4.2060	0.1307								
	4.1054	4.2040	0.0986								
x (- negativo)	-4.1690	4.2090	0.0400	4.164		4.207		0.043		0.207	
	-4.1750	4.2060	0.0310								
	-4.2081	4.2090	0.0009								
	4.2001	3.9770	0.2831								
	4.1309	3.9210	0.2099								
y (+ positivo)	4.1059	3.9210	0.1849	4.148		3.920		0.228		0.080	
	4.1582	3.9210	0.2332								
	-4.2207	3.9760	0.3027								
	-4.2315	3.9500	0.3115								
	-4.1142	3.9210	0.1932	4.161		3.918		0.243		0.082	
-4.0778	3.9740	0.1638									
Teste 4m - menos atrito Vel = 0.5 m/s --> 8sec		DATA: 24/03/17									
Sentido	Odometria (m)	Deslocamento Real (m)	Diferença Odometria/Real	Média Odometria (m)		Média Deslocamento (m)		Média Diferença Odometria/Real		Diferença Deslocamento Esperado e Deslocamento médio (m)	
				Δx  =	Δy  =	Δx  =	Δy  =				
x (+ positivo)	4.2304	4.2240	0.0064			4.207		0.004		0.207	
	4.2147	4.2120	0.0027								
	4.1764	4.1790	0.0026	4.208		4.207		0.004		0.207	
	4.2056	4.2130	0.0034								
	4.2092	4.2130	0.0038								
x (- negativo)	-4.1743	4.2130	0.0387	4.185		4.193		0.016		0.193	
	-4.1444	4.1500	0.0056								
	-4.2111	4.1950	0.0161								
	4.2320	4.0450	0.1870								
	4.1468	4.0740	0.1328	4.163		4.020		0.142		0.020	
y (+ positivo)	4.1678	4.0190	0.1488								
	4.1038	4.0030	0.1008								
	-4.1146	4.0110	0.1036								
	-4.0360	4.0360	0.0000								
	-4.2320	4.0360	0.1670	4.138		4.025		0.113		0.025	
-4.1680	4.0760	0.1500									

### E.1.3. Ensaios de Deslocamento de $\pi$ rad

Teste Rotação $\pi$ Vel = 1,0 rad/s --> $\pi$ sec		DATA:23/03/17	
Sentido	Média Odometria (rad)		Média Deslocamento (rad)
	Odometria (rad)	Deslocamento Real (rad)	
$\theta$ (+ positivo)	3,2223	3,1776	3,296
	3,2660	3,2756	
	3,3195	3,2916	
	3,3088	3,2996	
	3,2004	3,2966	
$\theta$ (- negativo)	3,2733	3,2966	3,301
	3,3183	3,2846	
	3,3204	3,2946	
Teste Rotação $\pi$ - menos altito Vel = 1,0 rad/s --> $\pi$ sec			
Média Odometria (rad)		Média Deslocamento (rad)	
$\Delta\theta$   = 3,284		$\Delta\theta$   = 3,278	
Diferença Odometria/Real		Diferença Deslocamento Esperado e Deslocamento médio (rad)	
0,0953		0,154	
0,0104			
0,0270			
0,0992			
0,0533			
0,0337			
0,0256			
Teste Rotação $\pi$ - menos altito DATA:24/03/17			
Média Odometria (rad)		Média Deslocamento (rad)	
$\Delta\theta$   = 3,278		$\Delta\theta$   = 3,331	
Diferença Odometria/Real		Diferença Deslocamento Esperado e Deslocamento médio (rad)	
0,0269		0,189	
0,0545			
0,0015			
0,1273			
0,1522			
0,0010			
0,0084			
0,0085			



## E.2. Ensaios Realizados com a Base de Rodas Universais

### E.2.1. Ensaios de Deslocamento de 1 m

Teste 1m Vel = 0.5 m/s --> 2sec		DATA:30/06/17		Média Odometria (m)		Média Deslocamento (m)		Média Diferença Odometria/Real		Diferença Deslocamento Expectativa e Deslocamento médio (m)	
Sentido	Odometria (m)	Deslocamento Real (m)	Diferença Odometria/Real	Δx  =	Δy  =	Δx  =	Δy  =	0.983	0.115	0.883	0.118
x (+ positivo)	0.985	0.903	0.082								
	1.006	0.876	0.13								
	1.005	0.876	0.13								
x (- negativo)	0.983	0.875	0.118								
	-1.008	0.923	0.083								
	-1.006	0.89	0.116								
y (+ positivo)	-1.006	0.862	0.144								
	-1.006	0.859	0.147								
	1.003	0.933	0.07								
y (- negativo)	1.004	0.941	0.063								
	1.004	0.935	0.069								
	1.004	0.946	0.058								
Teste 1m - menos atrito Vel = 0.5 m/s --> 2sec	-1.003	0.925	0.078								
	-1.003	0.92	0.083								
	-1.004	0.92	0.084								
	-0.995	0.917	0.078								
Teste 1m - menos atrito Vel = 0.5 m/s --> 2sec		DATA:30/06/2017		Média Odometria (m)		Média Deslocamento (m)		Média Diferença Odometria/Real		Diferença Deslocamento Expectativa e Deslocamento médio (m)	
Sentido	Odometria (m)	Deslocamento Real (m)	Diferença Odometria/Real	Δx  =	Δy  =	Δx  =	Δy  =	0.943	0.064	0.949	0.051
x (+ positivo)	1.006	0.937	0.069								
	1.007	0.949	0.058								
	1.007	0.936	0.071								
x (- negativo)	1.007	0.95	0.057								
	-1.007	0.948	0.059								
	-0.912	0.946	0.036								
y (+ positivo)	-1.007	0.951	0.056								
	-1.006	0.949	0.057								
	1.007	0.961	0.046								
y (- negativo)	1.007	0.957	0.05								
	1.007	0.956	0.051								
	1.007	0.954	0.053								
Teste 1m - menos atrito Vel = 0.5 m/s --> 2sec	-1.007	0.96	0.047								
	-1.007	0.961	0.046								
	-1.007	0.96	0.047								

## E.2.2. Ensaios de Deslocamento de 4 m

Teste 4m Vel = 0.5 m/s --> 8sec		DATA:30/06/17												
Sentido	Odometria (m)		Deslocamento Real (m)		Diferença Odometria/Real		Média Odometria (m)		Média Deslocamento (m)		Média Diferença Odometria/Real		Diferença Deslocamento Expectavel e Deslocamento médio (m)	
x (+ positivo)	3.93		3.538		0.392		3.980		3.491		0.490		0.510	
	3.983		3.405		0.578									
	4.005		3.502		0.503									
	4.003		3.517		0.486									
	-4.004		3.566		0.438									
x (- negativo)	-4.005		3.596		0.409							0.433		
	-3.93		3.58		0.35									
	-3.97		3.528		0.442									
	3.904		3.771		0.133									
	4.005		3.792		0.223									
y (+ positivo)	4.005		3.815		0.19		3.944		3.796		0.148		0.204	
	3.86		3.815		0.045									
	-4.005		3.745		0.26									
	-3.982		3.794		0.188									
	-4.005		3.799		0.206									
y (- negativo)	-4.004		3.577		0.427							0.271		
Teste 4m - menos atrito Vel = 0.5 m/s --> 8sec		DATA:30/06/17												
Sentido	Odometria (m)		Deslocamento Real (m)		Diferença Odometria/Real		Média Odometria (m)		Média Deslocamento (m)		Média Diferença Odometria/Real		Diferença Deslocamento Expectavel e Deslocamento médio (m)	
x (+ positivo)	4.005		3.744		0.261		4.005		3.762		0.243		0.238	
	4.005		3.756		0.249									
	4.005		3.762		0.243									
	4.003		3.766		0.239									
	-4.005		3.804		0.201									
x (- negativo)	-4.005		3.764		0.241							0.221		
	-4.005		3.766		0.239									
	-4.005		3.791		0.224									
	4.005		3.84		0.165									
	4.005		3.859		0.166									
y (+ positivo)	4.005		3.853		0.172		4.005		3.839		0.166		0.161	
	4.005		3.845		0.16									
	-4.006		3.827		0.179									
	-4.005		3.853		0.172									
	-4.006		3.83		0.176									
y (- negativo)	-4.005		3.835		0.17							0.169		

### E.2.3. Ensaios de Deslocamento de $\pi$ rad

Teste Rotação $\pi$ Vel = 1.0 rad/s --> $\pi$ sec		DATA: 30/06/17	
Sentido	Odometria (rad)	Deslocamento Real (rad)	Diferença Odometria/Real
	$\theta$ (+ positivo)	3.201	3.204
3.232		3.172	0.06
3.232		3.21	0.022
3.231		3.19	0.041
3.232		3.189	0.033
$\theta$ (- negativo)	3.233	3.196	0.035
	3.233	3.241	0.008
	3.233	3.28	0.047
Média Odometria (rad) 3.224			
Média Deslocamento (rad) 3.194			
Média Diferença Odometria/Real 0.032			
Diferença Deslocamento Expectavel e Deslocamento médio (rad) 0.052			
Teste Rotação $\pi$ - menos atrito Vel = 1.0 rad/s --> $\pi$ sec			
DATA: 30/06/17			
Sentido	Odometria (rad)	Deslocamento Real (rad)	Diferença Odometria/Real
	3.202	3.193	0.009
$\theta$ (+ positivo)	3.182	3.198	0.016
	3.19	3.203	0.013
	3.221	3.203	0.018
	3.212	3.186	0.026
	3.211	3.186	0.013
$\theta$ (- negativo)	3.202	3.193	0.009
	3.224	3.196	0.028
Média Odometria (rad) 3.199			
Média Deslocamento (rad) 3.189			
Média Diferença Odometria/Real 0.014			
Diferença Deslocamento Expectavel e Deslocamento médio (rad) 0.058			