



**Diogo Gil Alves de Carvalho**

Licenciado em Ciências da Engenharia  
Eletrotécnica e de Computadores

## **Controlo remoto de um robô com recurso ao sistema Android**

Dissertação para obtenção do Grau de  
Mestre em Engenharia Eletrotécnica e de Computadores

Orientador: Yves Rybarczyk, Professor auxiliar, FCT/UNL

Júri:  
Presidente: João Murta Pina  
Arguente: João Rosas



FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE NOVA DE LISBOA

**Março, 2015**



## **Controlo remoto de um robô com recurso ao sistema Android**

Copyright © Diogo Gil Alves de Carvalho, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.



*À minha família e amigos*



## AGRADECIMENTOS

Todo o apoio que me foi dedicado ao longo deste projeto teve um impacto fundamental na sua realização. Deixo aqui os meus agradecimentos aos que mais me apoiaram e foram fundamentais para o meu sucesso.

Ao Professor Yves Rybarczyk, meu orientador, que esteve sempre disponível quando precisei da sua ajuda, aconselhando com as melhores abordagens ao tema, e demonstrando ser um orientador exemplar.

A todos os meus amigos e colegas, Diogo Cardoso, Hugo Santos, João Marques, João Carreiro, Jorge Boavida, Bruno Ferreira, Fábio Xavier, Horácio Pires, Luís Oliveira, que me acompanharam ao longo desta caminhada académica e sempre me fizeram acreditar de que era capaz.

Agradeço especialmente à minha família, Maria da Conceição Alves, Jorge Carvalho e Miguel Carvalho, pois foram o maior suporte nesta etapa tão importante minha vida.



## RESUMO

---

A interação homem-máquina é muitas vezes um processo excessivamente artificial. Para o homem constitui um desafio permanente a otimização dessa interação capaz de controlar a máquina de modo mais eficiente através de uma maior simplicidade de procedimentos.

Este estudo sugere uma forma facilitadora desse controlo. Neste caso, pretende-se controlar um robô por meio de tele-operação, sem contacto visual direto.

Um robô incorporando uma câmara é controlado através de um smartphone, que para tal, irá exibir na sua interface a visão do robô e um painel tátil para o controlar.

No que diz respeito ao controlo, foram desenvolvidos dois modos de condução, um manual e outro com a velocidade assistida, recorrendo à lei de potência 2/3.

Deste modo, deseja-se mostrar que o modo de condução de velocidade assistida facilita a condução ao homem. Com essa finalidade, foi implementado um sistema assente na construção de um robô e no desenvolvimento de uma aplicação Android para controlo do mesmo. Com recurso a este sistema, foi elaborada uma experiência, da qual foram recolhidos dados a fim de testar a hipótese que um robô que implementa leis biológicas no seu modo de funcionamento, facilita a interação homem-máquina.

**Palavras-chave:** Lei de potência 2/3, smartphone, Android, Condução assistida, tele-operação, visão móvel, robô NXT.

---



## ABSTRACT

---

The interaction between man and machine is frequently an excessively artificial process.

The optimization of that interaction, which will enable the control of the machine in a more efficient way, through an increasing simplicity in procedure, constitutes a permanent challenge.

This study suggests a facilitating way to achieve that control. In this case, the intention is to control a robot by means of teleoperation with no direct visual contact. A robot with an incorporated camera is controlled through a smartphone, which will display the picture of the robot and a touch control panel in its interface.

In what comes to the control, two driving modes were developed: a manual and an assisted speed mode, which resorts to the  $2/3$  power law.

This way, we want to demonstrate that the assisted speed driving mode facilitates this human activity. Having this purpose in mind, a system based on the construction of a robot and the development of an Android application to control it has been implemented. Using this system an experiment was conducted in order to gather data to test the hypothesis that a robot that implements biological laws in its mode of operation, facilitates human-machine interaction.

**Keywords:** Power Law  $2/3$ , smartphone, Android, Assisted driving, teleoperation, mobile vision, NXT robot.

---



# CONTEÚDO

<b>Lista de Figuras</b>	<b>xv</b>
<b>Lista de Tabelas</b>	<b>xvii</b>
<b>Listings</b>	<b>xix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	1
1.2 Estado de Arte . . . . .	2
1.2.1 Robôs móveis . . . . .	2
1.2.2 Sistema operativo móvel . . . . .	6
1.2.3 Desenvolvimento de Aplicações Android . . . . .	9
1.2.4 Visão móvel vs Visão estática . . . . .	10
1.2.5 Leis biológicas usadas na interação homem-máquina . . . . .	13
1.3 Proposta . . . . .	15
<b>2 Implementação</b>	<b>17</b>
2.1 Introdução . . . . .	17
2.2 Arquitetura do sistema . . . . .	19
2.3 Construção do robô . . . . .	21
2.4 Comunicação com o robô NXT . . . . .	24
2.5 Transmissão de vídeo . . . . .	25
2.6 Orientação automática da IP câmara . . . . .	26
2.7 Condução em modo manual . . . . .	29
2.8 Condução em modo biológico . . . . .	31
2.9 Interface . . . . .	34
2.9.1 Fundo . . . . .	34
2.9.2 Botão "Calibrate" . . . . .	35
2.9.3 Botão "Connect/Disconnect" . . . . .	35
2.9.4 Botão "Reverse" . . . . .	36
2.9.5 Botão "Test Drive" . . . . .	36
2.9.6 Botão "Bio" e botão "Manual" . . . . .	37
2.9.7 Cronometro . . . . .	37

2.9.8	Velocímetro . . . . .	37
2.9.9	Painel tátil de condução . . . . .	38
<b>3</b>	<b>Experimentação</b>	<b>39</b>
3.1	Introdução da experiência . . . . .	39
3.2	Calculo da velocidade do robô . . . . .	41
3.3	Material/Método . . . . .	47
3.3.1	Percurso . . . . .	47
3.3.2	Participantes . . . . .	48
3.3.3	Procedimento . . . . .	48
3.4	Resultados . . . . .	52
3.4.1	Tempo . . . . .	52
3.4.2	Colisões . . . . .	53
3.4.3	Paragens . . . . .	55
3.4.4	Suavidade de trajetória - pequeno raio de curvatura . . . . .	56
3.4.5	Suavidade de trajetória - grande raio de curvatura . . . . .	58
3.5	Discussão . . . . .	60
<b>4</b>	<b>Conclusão e trabalho futuro</b>	<b>63</b>
	<b>Bibliografia</b>	<b>65</b>
<b>A</b>	<b>Anexo</b>	<b>67</b>

## LISTA DE FIGURAS

1.1	Especificações do Lego Mindstorms NXT 2.0 [10] . . . . .	2
1.2	Atuadores: lâmpadas e motores . . . . .	3
1.3	Sensores: sensor de toque, sensor de som, sensor de luz e sensor de ultra-som (da esquerda para a direita) . . . . .	3
1.4	Controlo de um robô através de um smartphone . . . . .	4
1.5	MyDFRobotShop Rover . . . . .	4
1.6	Finch Programmable Mobile Robot Platform[7] . . . . .	5
1.7	Quota de mercado de sistemas operativos móveis no mundo[11] . . . . .	6
1.8	Comparação entre os sistemas operativos[5] . . . . .	8
1.9	Arquitetura do sistema Android . . . . .	9
1.10	Painel esquerdo - um ciclista se aproximando de uma curva. A cabeça do ciclista está direcionada para o interior da curva. Painel direito - a vista da bicicleta com o ponto de fixação, marcado com um disco preto. . . . .	11
1.11	Exemplificação do modo de funcionamento de uma câmara móvel [21] . . . . .	11
1.12	Comparação entre as condições de visão móvel e fixa, ao longo de 4 sessões[18] . . . . .	12
1.13	Lei de Fitts - embora a distancia entre o bordo do alvo e o ponto de partida do movimento sejam os mesmos nas duas situações, o tempo para atingir o alvo será mais curto à direita do que à esquerda, porque o diâmetro do alvo é maior na configuração azul. . . . .	13
1.14	Representações logarítmicas e padronizadas dos testes: no (A) a "condição controle,"e (B) o modelo de "human-like"de antecipação[22] . . . . .	14
2.1	Modos de interação entre elementos . . . . .	19
2.2	Arquitetura do sistema . . . . .	20
2.3	Robô com as rodas em modo diferencial . . . . .	21
2.4	Esquemático do robô com as rodas em modo diferencial. As setas com apenas um sentido indicam as ligações dos motores as rodas e ao eixo da câmara. As setas com dois sentidos mostram a rotação dos mesmos. A diferença deste robô para o outro é o facto de que para este curvar terá que imprimir mais velocidade a uma roda do que a outra. . . . .	22
2.5	Robô com as rodas em modo direcional. . . . .	22

2.6	Esquemático do robô com as rodas em modo direcional. As setas com apenas um sentido indicam as ligações dos motores as rodas e ao eixo da câmara. As setas com dois sentidos mostram a rotação dos mesmos. A diferença deste robô para o outro é o facto de que para este curvar é utilizado um eixo que faz virar as rodas dianteiras. . . . .	23
2.7	Página onde se recolhe a imagem da IP câmara. . . . .	26
2.8	Comportamento da IP câmara consoante a curvatura. As setas vermelhas representam a direção da visão do robô, enquanto que as setas azuis representam a direção do robô a cada instante. . . . .	27
2.9	As nove áreas e o seus ângulo representativos . . . . .	28
2.10	Interface da condução manual . . . . .	29
2.11	Funcionamento do modo manual . . . . .	30
2.12	Interface da condução biológico . . . . .	31
2.13	Funcionamento do modo biológico . . . . .	32
2.14	Diagrama do algoritmo que atribui a velocidade ao robô . . . . .	33
2.15	Interface . . . . .	34
2.16	Botão "Calibrate" . . . . .	35
2.17	Botão "Connect/Disconnect" . . . . .	35
2.18	Botão "Reverse" . . . . .	36
2.19	Botão "Test Drive" . . . . .	36
2.20	Botão "Bio" e botão "Manual" . . . . .	37
2.21	Cronometro . . . . .	37
2.22	Velocímetro . . . . .	37
2.23	Interface da condução biológico . . . . .	38
2.24	Interface da condução manual . . . . .	38
3.1	Retas de regressão . . . . .	44
3.2	Velocidade . . . . .	44
3.3	Algoritmo que calcula a velocidade ao robô . . . . .	46
3.4	Percurso . . . . .	47
3.5	Percurso efetuado no sentido dos ponteiros do relógio . . . . .	49
3.6	Percurso efetuado no sentido contrário dos ponteiros do relógio . . . . .	49
3.7	Zona de inicio da experiência . . . . .	50
3.8	Exemplo de um percurso gravado dum ficheiro . . . . .	51
3.9	Tempo . . . . .	53
3.10	Colisões . . . . .	54
3.11	Paragens . . . . .	56
3.12	Suavidade de trajetória - pequeno raio de curvatura . . . . .	57
3.13	Suavidade de trajetória - grande raio de curvatura . . . . .	59
A.1	Tabela das velocidades . . . . .	68

## LISTA DE TABELAS

1.1	Comparação entre robôs . . . . .	5
1.2	Porcentagem de vendas no último trimestre de cada ano[11] . . . . .	6
3.1	Resultados obtidos da experiência para determinar a velocidade . . . . .	41
3.2	Equação $Y = bX^a$ . . . . .	42
3.3	Valores seguindo uma reta exponencial . . . . .	42
3.4	Equação $Y = mX + b$ . . . . .	43
3.5	Valores seguindo uma reta linear . . . . .	43
3.6	Método experimental . . . . .	50



## LISTINGS

2.1	Algoritmo para determinar as 9 zonas . . . . .	28
2.2	Algoritmo que atribui a velocidade ao robô . . . . .	33
3.1	Algoritmo que calcula a velocidade ao robô . . . . .	45



## INTRODUÇÃO

### 1.1 Motivação

A interação homem-máquina é uma área interdisciplinar que relaciona a ciência da computação, psicologia, sociologia, semiótica, artes, design, ergonomia e linguística. A interação do homem com a máquina é feita através de uma interface. Muitas vezes essa interação torna-se bastante artificial por estarem completamente afastadas da realidade humana.

Um grande desafio é aproximar estas duas realidades, o homem da máquina, de modo a tornar essa interação o mais natural possível. Para isso, uma abordagem interessante é implementar leis biológicas para melhorar a interação entre o homem e a máquina[20]. Estas leis tem como objetivo atribuir às máquinas comportamentos humanos. Deste modo, a interação torna-se mais intuitiva para o homem, pois a máquina se irá comportar tal como este se comportaria no seu lugar.

Hoje, vive-se na era dos smartphones e uma boa interação entre estas máquinas e o homem é cada vez mais importante. A interação é feita através do toque nos seus ecrãs. Para uma melhor utilização destes dispositivos, a interface terá que ser cada vez mais intuitiva para o homem, ou seja, que a realização de um determinado movimento possa reproduzir no mundo virtual a mesma realidade que se teria no mundo real.

Neste momento, estes dispositivos já se encontram num estado de evolução bastante avançado, razão pela qual podem ser o meio de interação entre o homem e outras máquinas, por exemplo, como controlos remotos de robôs móveis. Para esse efeito, o homem terá que utilizar a interface do smartphone para se comunicar com o robô móvel. Será conveniente que esta seja o mais intuitiva possível, de modo que o utilizador ao realizar certo movimento, o robô vá reagir de forma concordante com o desejo que o utilizador espera desse movimento. Visando facilitar a tarefa ao utilizador, poderão ser introduzidas leis biológicas no comportamento do robô. Este é um grande desafio, o de aproximar o comportamento da máquina do comportamento humano.

Outro grande desafio, é utilizar smartphones para controlar robôs à distância e sem contacto visual direto com o veículo. Para tal, os robôs terão que possuir sensores, neste caso uma câmara, para informar o utilizador da sua localização e do ambiente que o rodeia. A exibição dessa imagem reproduz-se no ecrã de um smartphone, em tempo real.

## 1.2 Estado de Arte

### 1.2.1 Robôs móveis

#### 1.2.1.1 Lego Mindstorms NXT

O Lego Mindstorms NXT é um robô da LEGO programável[10]. As características das peças da LEGO permitem a possibilidade de facilmente montar o veículo idealizado, pois possui um enorme número de componentes que tornam possível essa sua grande versatilidade. O início da sua comercialização ocorreu em 2006. Possui um processador 32-bit ARM7 microcontrolador com 256 KBytes de memória flash e 64KBytes de memória RAM. Inclui ainda um processador de 8-bit Atmel AVR ATmega48 controlador com 4 KBytes de memoria flash e 512 Bytes de memoria RAM[15]. As especificações técnicas encontram-se mais detalhadas na figura 1.1.

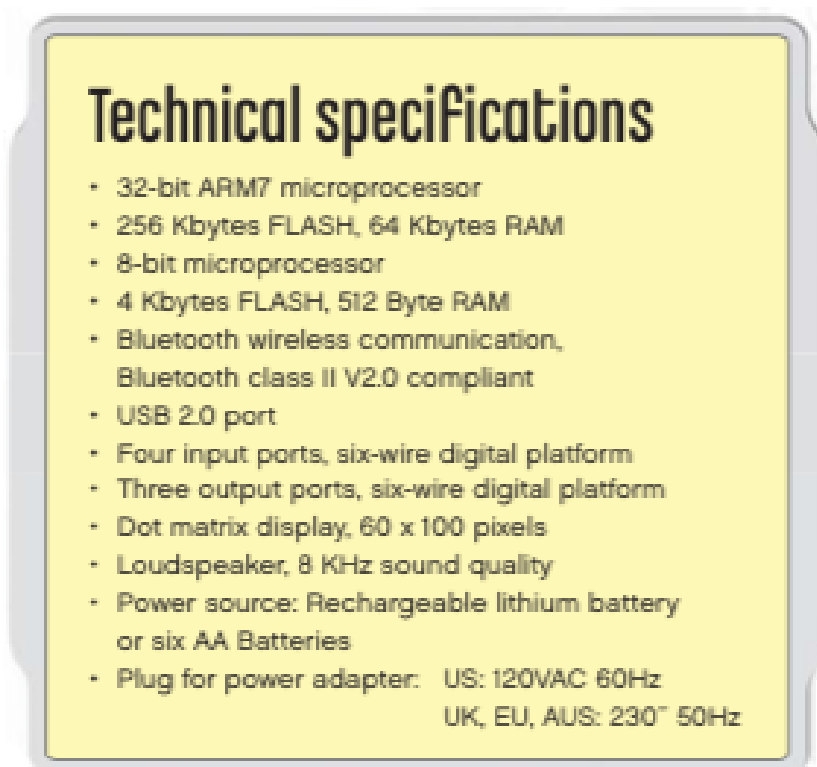


Figura 1.1: Especificações do Lego Mindstorms NXT 2.0 [10]

O NXT brick possui três portas de saída onde se podem ligar motores e lâmpadas, e quatro portas de entrada onde podem ser ligados sensores.

Este atua sobre o meio através dos motores e luzes, como se pode observar na figura 1.2, e recebe informações do mesmo através dos sensores que, na versão básica, podem ser de 4 tipos: sensor de toque, sensor de som, sensor de luz e sensor de ultra-som, representados na figura 1.3. Existe ainda a possibilidade adicionar sensores mais avançados, como por exemplo: uma bússola, um giroscópio ou um acelerómetro.



Figura 1.2: Atuadores: lâmpadas e motores



Figura 1.3: Sensores: sensor de toque, sensor de som, sensor de luz e sensor de ultra-som (da esquerda para a direita)

As duas vias de comunicação possíveis são: USB 2.0 e Bluetooth. Podem ser enviados ficheiros com instruções recorrendo ao USB 2.0. Através do Bluetooth é possível enviar comandos para atuar sobre os motores ou lâmpadas e receber dados dos sensores, em tempo real. Desta forma, torna-se possível controlar um robô, por exemplo, através de um smartphone, tal como está representado na figura 1.4.

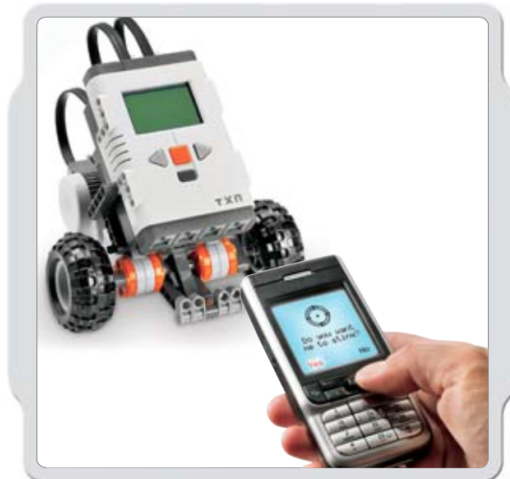


Figura 1.4: Controle de um robô através de um smartphone

Este robô poderá ser programado, utilizando a sua linguagem de programação oficial, NXT-G, ou opcionalmente, o LabVIEW para LEGO MINDSTORMS. Existe ainda uma variedade de línguas não-oficiais, tais como: NXC, NBC, leJOS NXJ e RobotC[17].

#### 1.2.1.2 MyDFRobotShop Rover

O MyDFRobotShop Rover é um robô móvel versátil baseado no popular Arduino Uno USB Microcontroller Rev 3[6, 16]. Move-se utilizando lagartas, pois tem a forma de um tanque. O DFRobotShop Rover PCB possui um standard Arduino Uno (montagem em superfície ATmega328), com um motor L293B (conectado aos pinos 5-8), regulador de tensão e área de prototipagem, contribuindo simultaneamente para a estrutura mecânica do robô. A linguagem de programação utilizada é C/C++[4]. A comunicação a este robô pode ser efetuada através de Bluetooth ou Wireless. Na figura 1.5 está representado este robô.

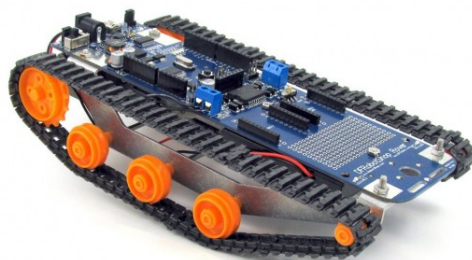


Figura 1.5: MyDFRobotShop Rover

### 1.2.1.3 Finch Programmable Mobile Robot Platform

O Finch Programmable Mobile Robot Platform surgiu na Carnegie Mellon's CREATE lab, que resultou de um estudo que se prolongou por quatro anos[7]. Este robô é muito usado em ambientes universitários. São projetados para permitir que os alunos criem programas interativos. O Finch é atualmente programável em Java, Python, C ++, Visual Basic, Scala, e Processing, sendo possível a utilização de outras linguagens. O robô possui acelerômetros, motores e buzina[7]. Na figura 1.6 pode se observar este robô.



Figura 1.6: Finch Programmable Mobile Robot Platform[7]

### 1.2.1.4 Comparação entre robôs

A tabela 1.1 revela que o Lego Mindstorms NXT é o robô mais versátil, com linguagens de programação mais simples e embora não tendo wireless como o MyDFRobotShop Rover, permite a tele-operação através de Bluetooth.

Tabela 1.1: Comparação entre robôs

Robô	Versatilidade	Programação	Conectividade
Lego Mindstorms NXT	Elevada	NXT-G, LabVIEW NXC, NBC, leJOS NXJ e RobotC	Bluetooth USB
MyDFRobotShop Rover	Moderada	C/C++	Bluetooth USB Wireless
Finch Programmable Mobile Robot Platform	Baixa	Java, Python, C ++, Visual Basic, Scala, e Processing	USB

### 1.2.2 Sistema operativo móvel

Até ao ultimo trimestre de 2014, o mercado de dispositivos móveis cresceu 28,2% ano após ano[11]. De acordo com Worldwide Quarterly Mobile Phone Tracker, foram expedidas 377,5 milhões de unidades[11].

Os principais sistemas operativos móveis são: Android, iOS, Windows Phone e BlackBerry OS. O sistema operativo Android é o líder de mercado com uma quota de 76,6% no ultimo trimestre de 2014. IOS é o segundo mais expedido, com uma quota de 19,7%. Este pertence à Apple, e no último trimestre de 2014 foram vendidas 74,5 milhões de unidades, tendo um crescimento ano após ano de 46%. No ano de 2014 foram vendidas cerca de 1,3 mil milhões de unidades, tendo ocorrido um crescimento de 27,7% em relação ao ano de 2013 onde as vendas rondaram os mil milhões de unidades[11]. Na figura 1.7 está representada a quota de mercado de cada sistema operativo.

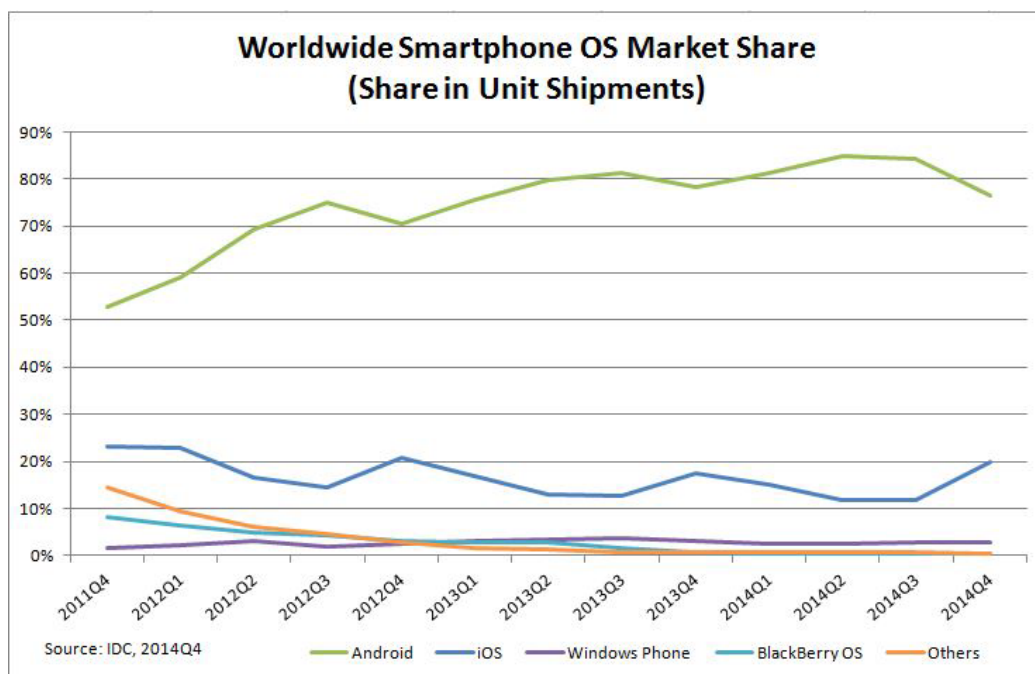


Figura 1.7: Quota de mercado de sistemas operativos móveis no mundo[11]

Tabela 1.2: Percentagem de vendas no último trimestre de cada ano[11]

Período	Android	iOS	Windows Phone	BlackBerry	Outras
2014	76.6%	19.7%	2.8%	0.4%	0.5%
2013	78.2%	17.5%	3.0%	0.6%	0.8%
2012	70.4%	20.9%	2.6%	3.2%	2.9%
2011	52.8%	23.0%	1.5%	8.1%	14.6%

De acordo com [11], em termos de quota de mercado, tem-se constatado que:

**Android** — A venda de dispositivos com este sistema operativo no último ano vendeu mais de mil milhões de unidades, ultrapassado assim a venda total de dispositivos do ano de 2013. A Samsung, uma das marcas de dispositivos móveis que utilizam o Android, é a principal responsável por este número de vendas tão avultado. Mas os principais responsáveis por este crescimento são: Huawei, Lenovo (incluindo Motorola), LG Electronics, Xiaomi, e ZTE.

**iOS** — No ano de 2014 houve uma ligeira diminuição de vendas de dispositivos com este sistema operativo. Contudo, o seu crescimento foi quase no mesmo ritmo do mercado, no geral. O grande responsável por esse crescimento foi o iPhone 6 e iPhone 6 Plus, pois tiveram uma grande procura. Embora a procura tenha sido muito grande, foi sujeito a muitas críticas, sobretudo devido aos ecrãs destes dispositivos serem muito grande.

**Windows Phone** — Com um crescimento de apenas 4,2%, este foi o que apresentou um menor crescimento ano após ano, muito abaixo do crescimento do mercado. A quota de mercado deve-se à venda de dispositivos da gama Lumia, antiga gama da Nokia, que foi adquirida pela Microsoft no início de 2014. Os parceiros HTC e Samsung também foram importantes no fecho de mercado em alta, através do lançamento do Windows mobile 10.

**Blackberry OS** — Este foi único sistema operativo, dos principais do mercado, que teve um declínio nas suas vendas ano após ano, caiu cerca de -69,8% em relação ao ano de 2013. O ano de 2014 foi um ano de reestruturação do mesmo, pois até ao final do ano foram apresentadas melhorias na plataforma e novas funcionalidades, o BlackBerry Passport e BlackBerry Classic. A perspectiva para 2015 é que sejam vendidas cerca de 10 milhões de unidades, o que marcaria um aumento de 72% em relação as vendas de 2014, que foram na ordem de 5,8 milhões.

A figura 1.8 mostra a comparação entre os diferentes sistemas operativos.

	 iOS 7	 Android 4.2	 Windows Phone 8	 BlackBerry 10
Control Centre	Yes, iOS 7 brings a new Control Centre to Apple devices, giving users quick access to toggles and other commonly-used features	Yes, Android 4.2 does have a central Control Centre, but it varies as per the phone manufacturer	No, Windows Phone 8 does not have provision for a Control Centre, and users need to go to the Settings menu for this	Just like Windows Phone 8, BlackBerry 10 also does not have a Control Centre and users will have to go to the system access settings every time
Notification Centre	Yes, Apple devices continue to have the notification centre, but it is now divided into three tabs, Today, All and Missed	Yes, Android 4.2 boasts of detailed notifications, including toggles, music controls etc	Microsoft's Windows Phone 8 mobile OS does not have a central notification centre, but users can access controls from the home screen via Live Tiles	To see all notifications and messages, BB 10 OS has a feature called BlackBerry Hub
Multitasking preview	iPhones, iPads and iPod touch will now get a flat design for multitasking preview with the new iOS 7	Users can switch between apps in Android 4.2 via the Recents list	Windows Phone 8 allows users to move between active apps via the multitasking preview feature	The BlackBerry 10 homescreen shows users all currently operating apps via the Active Frames grid
Peer-to-peer sharing	Apple has introduced AirDrop with iOS 7 that enables people to share contacts, songs, videos etc over Bluetooth and Wi-Fi	Android Beam is the NFC-based feature that allows people to share data wirelessly	In WP8, users can share data with others via the NFC-based Tap+Share feature	NFC-based sharing is also present in BlackBerry 10 operating system
Unified browser bar	Yes, a unified browser bar is present in the Safari browser	Yes, the default Android 4.2 web browser features a central browser bar for all tabs	Yes, Windows Phone 8 also has a unified browser bar	Yes, a unified browser bar is also present in the BlackBerry 10 default web browser
Photo grouping	Apple has introduced photo grouping features called Moments, Collections and Years with iOS 7	In Android 4.2, images are grouped by albums, dates and other tags	Photos are segregated as per albums and date in Windows Phone 8	The BB10 operating system categorizes photos by albums and recently clicked
Shared photo stream	Apple's new mobile operating system allows people to share Photo Streams with other users	Only Samsung Galaxy S4 has the provision to share such photo streams	Users can only share images one by one in Windows Phone 8 OS	Similar to WP8, BlackBerry 10 also requires users to share images one by one
Automatic app updates	Apple devices will now update apps automatically in the background	While apps do not update automatically in Android 4.2, users can select the option while downloading them	Yes, Windows Phone 8 does have an Update All option for apps	Users can only update apps one by one in BlackBerry 10
Password-protected reset	Yes, iOS 7 brings this feature, stolen devices cannot be used until the owners' Apple ID and password are provided	No, Android 4.2 does have provision to protect devices when stolen	No, there is no feature to protect stolen phones running on WP8	No, BlackBerry 10 OS does not allow devices to be bricked when stolen
In-car integration	Apple introduced this with the latest software update, so that they can access calls, messages, maps etc hands-free	Driving Mode in Android 4.2 allows in-car integration for smartphones and tablets	Nokia Drive is available on Windows Phone 8 for in-dash integration	No default feature to make phones compatible with cars, a few third-party apps can perform this function

Figura 1.8: Comparação entre os sistemas operativos[5]

## 1.2.3 Desenvolvimento de Aplicações Android

### 1.2.3.1 Android

Atualmente, a maioria dos dispositivos móveis em mais de 190 países possui sistema Android[3]. Desde o início da sua comercialização tem-se verificado um crescimento acentuado até aos dias de hoje. Este é um sistema baseado em Linux "open source", o que permitiu um rápido progresso. Este tornou-se um dos favoritos para consumidores e programadores. Através da plataforma Google Play, é possível também observar um enorme crescimento, com um grande aumento de aplicações, assim como os *downloads* destas.

Este sistema operativo tem como base o kernel do Linux, projetado para dispositivos móveis[1]. Este é "open source", o que levou a que muitos fabricantes adotassem este sistema operativo para as suas máquinas. A arquitetura do sistema do Android é representada abaixo na Figura 1.9.



Figura 1.9: Arquitetura do sistema Android

### 1.2.3.2 Eclipse e Android SDK

Eclipse é uma plataforma para desenvolvimento de software "open source" que inicialmente foi criada pela IBM. Posteriormente foi criada a fundação Eclipse sem fins lucrativos, que é detentora da plataforma [12, 13]. Esta é uma plataforma de desenvolvimento de software livre e extensível, baseada em Java. Consiste numa estrutura e num conjunto de serviços para desenvolvimento de aplicações de componentes de plug-in. Um exemplo de aplicativo baseado em Eclipse é o IBM® Rational® Software Architect, que compõe a base da família de ferramentas de desenvolvimento Java da IBM.

O Android SDK é um pacote que fornece as bibliotecas de API e todas as ferramentas necessárias para desenvolver, testar e fazer *debug* de aplicações para Android [2]. Para mais facilmente instalar estas ferramentas, existe um plugin designado de ADT, aumentando os recursos fornecidos no eclipse. Este plugin permite criar novos projetos ou construir uma interface de um modo mais prático, fazer debug num dispositivo físico e exportar a aplicação criada através de um pacote APK.

### 1.2.4 Visão móvel vs Visão estática

Um dos aspetos mais importantes a ter em conta na tele-operação de robôs é o modo de representação e perceção do espaço, o que se torna um problema devido à grande separação sensorial entre a entidade que controla a ação (ser humano) e a entidade de execução (o robô) [21]. Neste aspeto, para facilitar o desempenho do operador, as informações de feedback sobre o ambiente e ação realizada pelo robô têm que ser fornecidas de uma forma natural, de modo a aproximar as duas entidades a nível sensorial.

A perceção sensorial neste aspeto diz respeito ao campo visual que o operador tem do robô. Resultam daqui, problemas que podem ser de dois tipos de natureza: temporal e espacial.

Na perspetiva temporal, estudos têm demonstrado que um atraso de até 300 milissegundos cria grandes dificuldades para controlar o sistema [21]. Mesmo sem atraso, a visão é indireta o que implica muitas vezes uma falta de perceção da paralaxe do movimento e do fluxo ótico. Na perspetiva espacial, envolve dificuldade para avaliar distâncias e profundidades. De modo a diminuir as dificuldades de perceção espacial, a antecipação da visualização do cenário através de uma visão móvel parece ser uma boa solução comportamental. Durante um deslocamento, o facto da visão acompanhar a trajetória do robô faz com que o operador possua uma maior facilidade em antecipar a trajetória futura. Acompanhar a trajetória do robô consiste, em que nas trajetórias curvas, a orientação da visão esteja direcionada para o interior da curva, de modo a antecipar da melhor forma a amplitude de cada curva. Este é um comportamento natural que o ser humano executa automaticamente quando faz uma curva a pé [9] ou quando conduz [14] (ver figura 1.10). A figura 1.11 exemplifica melhor o modo de comportamento de uma visão móvel.

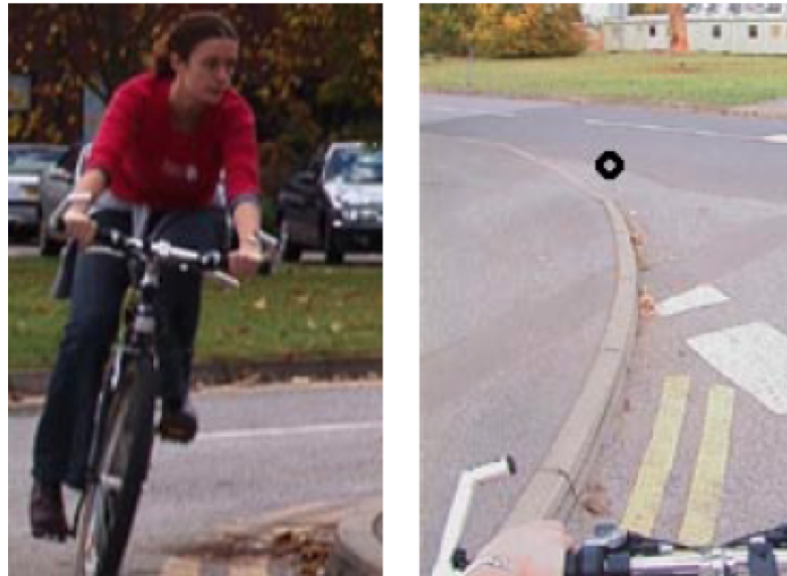


Figura 1.10: Painel esquerdo - um ciclista se aproximando de uma curva. A cabeça do ciclista está direcionada para o interior da curva. Painel direito - a vista da bicicleta com o ponto de fixação, marcado com um disco preto.

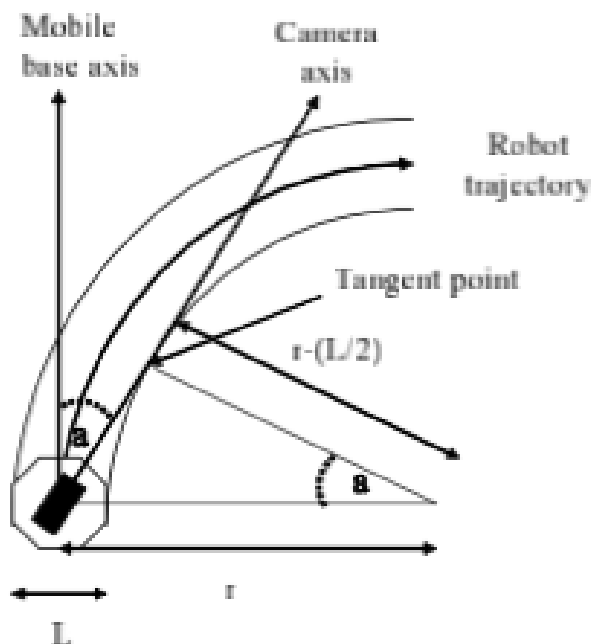


Figura 1.11: Exemplificação do modo de funcionamento de uma câmara móvel [21]

Assim, uma visão móvel torna a percepção visual do operador mais natural do que uma visão fixa. Estudos realizados provaram que, em ambientes virtuais [18] e com robôs [21], que utilizam de uma visão móvel (figura 1.12, à direita), as trajetórias são mais suaves do que quando a visão é fixa (figura 1.12, à esquerda). Como se pode observar na figura 1.12, as trajetórias obtidas com uma câmara móvel são mais naturais.

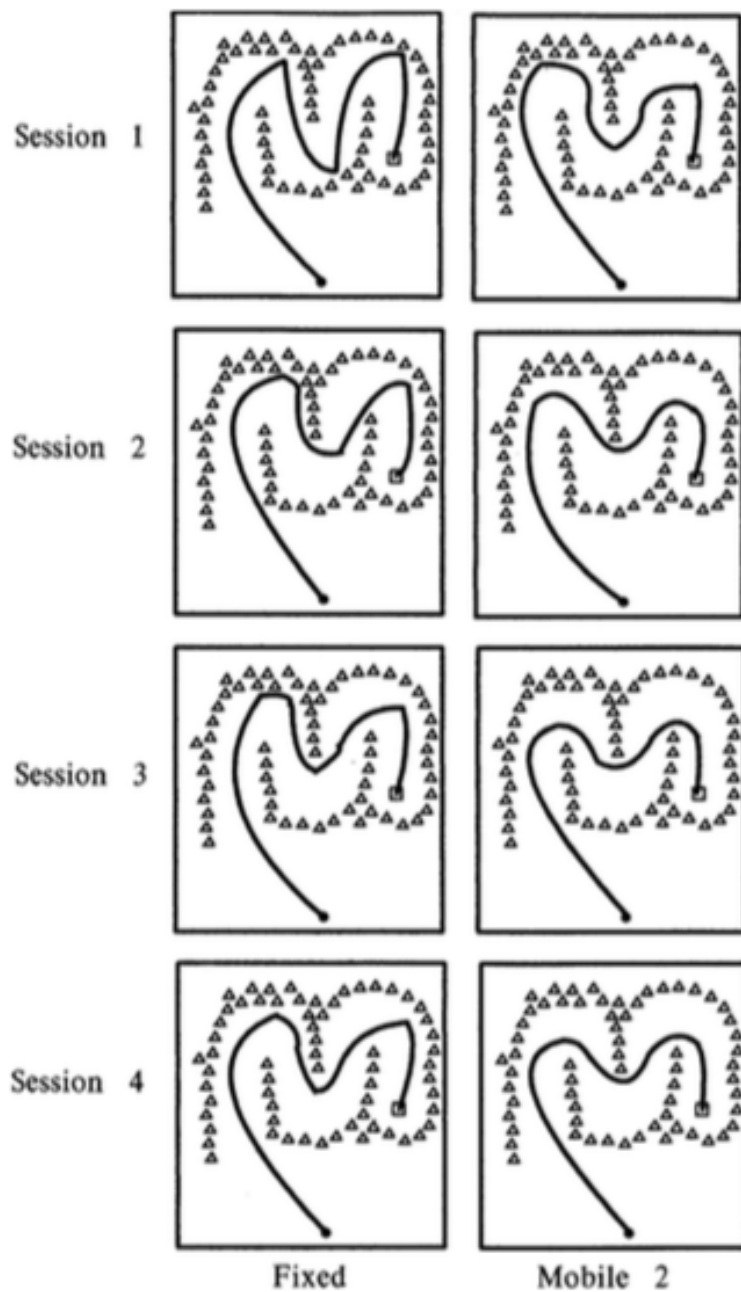


Figura 1.12: Comparação entre as condições de visão móvel e fixa, ao longo de 4 sessões[18]

## 1.2.5 Leis biológicas usadas na interação homem-máquina

### 1.2.5.1 Lei de Fitts

A lei de Fitts é um modelo descritivo do movimento humano, usado principalmente em ergonomia e interação homem-máquina[8]. Este modelo prevê o tempo necessário para se mover de um ponto inicial até que consiga atingir o seu alvo, em função da distância e do tamanho deste. Esta lei usa-se para planejar o ato de apontar, tanto no mundo real como no mundo informático.

O modelo Matemático da lei de Fitts tem sido formulado de várias formas diferentes. A forma mais comum de se apresentar é a de Shannon para movimentos sobre uma única dimensão:

$$T = a + b \log_2 \left( \frac{D}{W} + 1 \right)$$

em que:

- T é o tempo médio necessário até atingir o alvo.
- a e b são constantes empíricas, e podem ser determinadas aproximando os dados medidos com uma linha reta.
- D é a distância desde o ponto inicial até o centro do alvo.
- W é a largura do alvo medido ao longo do eixo de movimento.

É possível observar, através desta equação, que existe uma relação velocidade/precisão, que está relacionada com o ato de fazer pontaria, e que implica a necessidade de utilizar mais tempo para atingir alvos mais pequenos ou mais distantes. Através da figura 1.13 é possível perceber melhor esta lei.

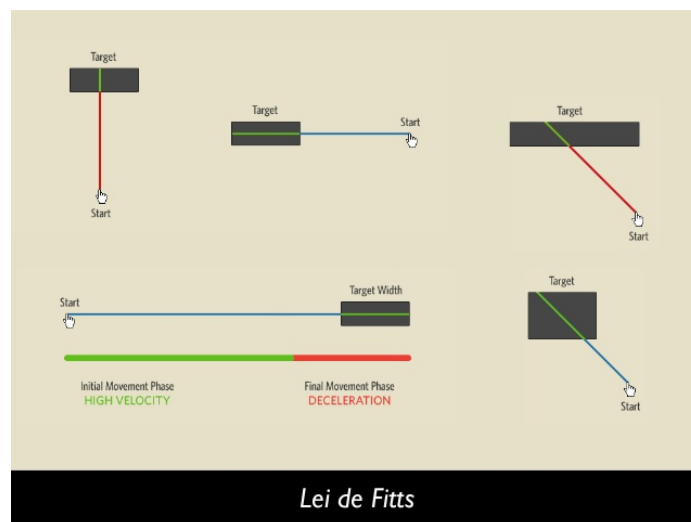


Figura 1.13: Lei de Fitts - embora a distância entre o bordo do alvo e o ponto de partida do movimento sejam os mesmos nas duas situações, o tempo para atingir o alvo será mais curto à direita do que à esquerda, porque o diâmetro do alvo é maior na configuração azul.

### 1.2.5.2 Lei de potência $\frac{2}{3}$

No início dos anos oitenta, a correlação que existe entre a cinemática da escrita à mão e a trajetória de movimento foi formalizada em termos de uma equação, conhecida como lei de potencia 2/3[19]. Esta lei liga curvatura de uma trajetória com a velocidade angular.

$$v_{\theta}(t) = kc^{2/3}(t)$$

A mesma expressão também pode ser apresentada em termos de uma relação entre a velocidade curvilínea e o raio de curvatura da trajetória.

$$v(t) = kr^{1/3}(t)$$

A lei foi ligeiramente modificada e adaptada ao longo dos anos para cobrir ainda maior conjunto de movimentos. A mais recente formulação [19] foi a seguinte:

$$v(t) = k(t) \left( \frac{r(t)}{1 + \alpha r(t)} \right)^{\beta}$$

Onde  $k(t)$  é uma função positiva do tempo,  $\alpha$  é uma constante positiva e o  $\beta$ , constante, é aproximadamente igual a 1/3.

Apresenta-se, de seguida, um exemplo onde a lei de potencia 2/3 foi usada para a avaliação de uma IHM com robôs moveis[22]. Os resultados apresentados na figura 1.14 comprovam que esta lei biológica é apenas preservada quando a câmara implementa uma antecipação visual sobre a deslocação do robô (gráfico de direita).

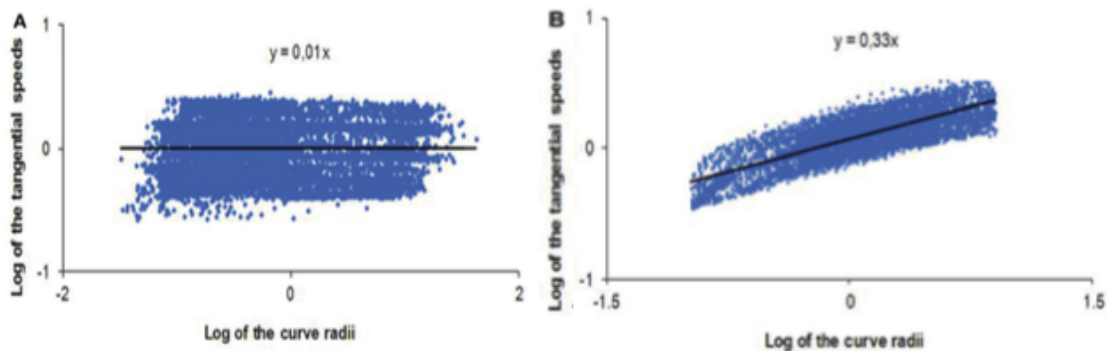


Figura 1.14: Representações logarítmicas e padronizadas dos testes: no (A) a "condição controle," e (B) o modelo de "human-like" de antecipação[22]

### 1.3 Proposta

Nesta tese, propõe-se implementar um sistema de assistência ao controlo remoto (ou tele-operação), sem contacto visual direto, de um robô NXT, com recurso a um dispositivo móvel baseado em Android.

O trabalho irá incluir três fases:

A primeira será dedicada à construção do robô. Sobretudo, procurou-se conceber um equipamento que se possa adequar ao que dele é pretendido e que cumpra as funções para as quais será concebido.

A segunda incluirá a programação de uma aplicação em android capaz de controlar o robô. Esta aplicação terá que ser o mais intuitiva possível de maneira a ser mais um contributo para facilitar a condução (interação homem-máquina). Para controlar o robô, exclusivamente, através de um dispositivo móvel baseado em Android, a aplicação terá que conter a visão do robô e um painel tátil que permitirá o controlo do robô.

Na aplicação serão desenvolvidos dois algoritmos de condução: um algoritmo de controlo autónomo da velocidade de deslocação do robô em função da sua trajetória, segundo a “Lei de Potência 2/3” [23] e outro que terá de ser capaz de retratar o modo de condução manual tradicional, onde o utilizador terá que controlar também a velocidade. De modo a facilitar a condução do robô terá que ser utilizada uma visão móvel. Esta acompanhará de um modo proporcional a curvatura da trajetória do robô. Estudos demonstraram que este sistema permite que a condução se faça de um modo mais natural[18].

Na última fase deverá montar-se um cenário experimental, onde se irá utilizar todo o sistema que foi implementado, a fim de observar eventuais vantagens do modo de condução com controlo automático da velocidade de deslocação do robô em função da sua trajetória em oposição ao controlo manual tradicional de um robô móvel.

Para esse efeito, serão convidados a participar 20 pessoas, que irão testar os modos de condução num percurso.

Terão que se desenvolver os protocolos para comunicar remotamente com o NXT e com uma IPcâmara incorporada no robô, utilizando um sistema Android que servirá ao mesmo tempo como controlo e terminal de visualização do tele-operador.

O principal objetivo será, investigar se haverá alguma vantagem em utilizar um algoritmo de controlo autónomo da velocidade de deslocação do robô em função da sua trajetória, segundo a “Lei de Potência 2/3” para a condução do robô, em oposição ao modo de condução manual tradicional.



## IMPLEMENTAÇÃO

### 2.1 Introdução

Esta dissertação de mestrado consiste em implementar um sistema de assistência ao controlo remoto de um robô NXT, com recurso a um dispositivo móvel baseado em Android. O robô NXT foi concebido para ser controlado sem que o utilizador tenha contacto visual com o mesmo. Esta situação está descrita no estado de arte como uma condição de tele-operação. Nele, foi instalada uma câmara a partir da qual são enviadas imagens, em tempo real, para o dispositivo móvel baseado em Android. Deste modo, a "visão" do robô é dada a conhecer ao utilizador. Porém, esta não é uma visão estática atendendo a que se procurou que o movimento da câmara estivesse sincronizado com o movimento do robô.

O controlo do robô é feito a partir do dispositivo móvel baseado em Android. No fundo do ecrã deste dispositivo, para além de nele serem exibidas as imagens enviadas pela câmara, encontra-se disponível um painel tátil que foi criado com a finalidade de, através dele, o utilizador poder controlar o robô.

O trabalho foi dividido em três fases:

A primeira foi dedicada à construção do robô. Sobretudo, procurou-se conceber um equipamento que se adequasse ao que dele era pretendido e que cumprisse as funções para as quais era concebido.

A segunda foi o desenvolvimento de uma aplicação em Android para que o robô fosse controlado, com recurso a um dispositivo móvel baseado em Android. A aplicação foi desenvolvida através da plataforma Android SDK utilizando o programa Eclipse. Nesta aplicação, foram desenvolvidos dois algoritmos de condução, um deles consiste num modo de condução manual no qual o utilizador será capaz de controlar a velocidade e trajetória do robô independentemente uma da outra. O outro, consiste num algoritmo de controlo autónomo da velocidade de deslocação do robô em função da sua trajetória, segundo a

"Lei de Potência 2/3"[23]. Este modo de controlo será designado no resto do documento como "modo biológico".

Na terceira parte foi montado um cenário experimental a fim de testar a validade do modelo implementado comparando então os dois modos de condução desenvolvidos na aplicação Android.

Para que esta experiência se pudesse realizar com sucesso, houve necessidade de se desenvolver os protocolos para que o dispositivo móvel baseado em Android comunicasse remotamente com o NXT, recorrendo à tecnologia de comunicação sem fios Bluetooth. Além desta comunicação, foi, igualmente, necessário desenvolver também os protocolos que permitissem que o dispositivo móvel baseado em Android comunicasse remotamente com uma IP câmara incorporada no NXT, a "visão" do robô. Para este efeito, recorreu-se à tecnologia de comunicação sem fios Wi-Fi.

## 2.2 Arquitetura do sistema

O sistema é composto por três elementos: Um dispositivo móvel baseado em Android, um robo NXT e uma câmara IP. Estes três elementos deverão permitir efetuar percursos, quer no modo de condução manual, quer no modo de condução biológica. O percurso deverá ser efetuado não tendo o utilizador contacto visual com o robô, devendo para esse efeito ser utilizada a câmara IP.

Os três elementos encontram-se ligados de dois modos distintos. O dispositivo móvel baseado em Android liga-se ao robô através da tecnologia Bluetooth e à câmara IP através da tecnologia Wi-Fi. A ligação entre o robô e a câmara IP é feita através de um suporte, de modo a que esta fique incorporada no robô.

O dispositivo móvel baseado em Android é o controlo do sistema, sendo através deste que o utilizador irá enviar indicações ao robô e receber a imagem da sua "visão". Para que este se torne no controlador foi desenvolvida uma aplicação em Android. Através da aplicação, irão ser enviadas para o robô as instruções que o utilizador indicar através da interface de utilizador. A aplicação receberá os dados enviados, sob a forma de imagem, pela IP câmara, e irá exibir essa imagem recorrendo à referida interface.

Através da interface é possível controlar a trajetória do robô, o sentido e o modo como a condução é efetuada, calibrar a direção da IP câmara, conectar e desconectar o dispositivo móvel baseado em Android do robô e por ultimo, possui a funcionalidade de iniciar um teste no percurso.

Na aplicação são exibidas a velocidade do robô e o tempo de percurso, para que o utilizador tenha noção das mesmas.

A ligação à IP câmara ocorre no momento em que a aplicação é iniciada, não podendo o utilizador controlar quando esta é conectada e desconectada, como acontece com o robô.

Na figura 2.1, está exemplificado o modo como as comunicações são feitas e as tecnologias utilizadas.

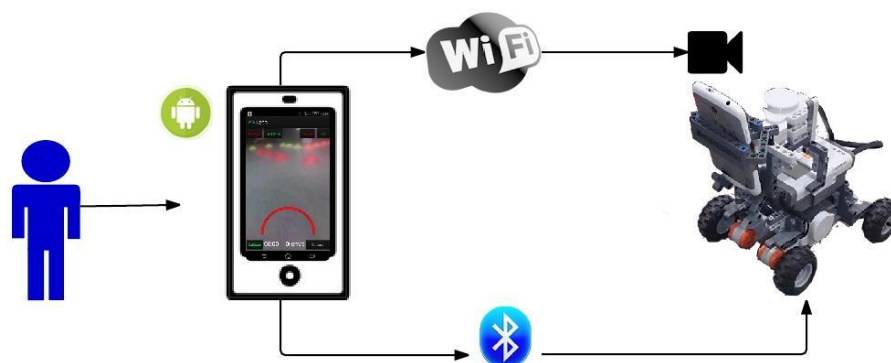


Figura 2.1: Modos de interação entre elementos

A aplicação, ao ser iniciada, conecta-se instantaneamente com a IP câmara, através de WI-FI. De seguida, é a exibição da interface do utilizador. Depois desta estar visível, é pedido ao utilizador para conectar o controlador ao robô. Assim que estes estiverem conectados, através do protocolo Bluetooth, a aplicação fica operacional para conduzir o robô. Nesta altura, estão visíveis várias funcionalidades para que seja possível a elaboração da experiência.

Na figura 2.2, é explicado através de um diagrama o modo de funcionamento da aplicação.

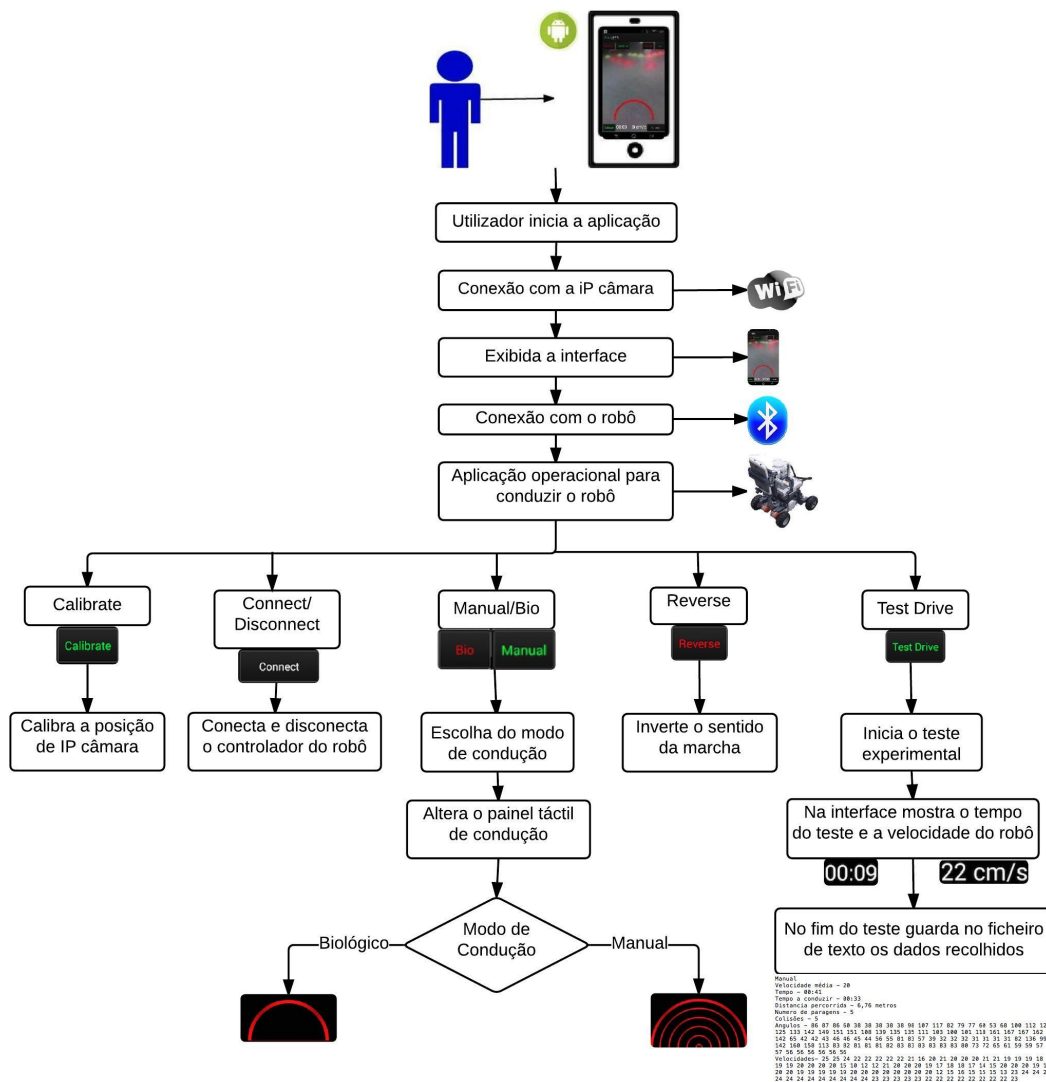


Figura 2.2: Arquitetura do sistema

## 2.3 Construção do robô

A primeira parte do trabalho foi dedicada a construção de um robô que se adequasse ao que era pretendido.

Para tal, chegou-se ao entendimento de que a decisão da escolha do robô deveria assentar num quadriciclo. Devendo este, ser equipado com um suporte onde se pudesse colocar uma IP câmara, que seria a componente de "visão" do robô.

Um dos aspetos de maior relevância que terá de ser considerado relaciona-se com o modo como o veículo iria curvar durante o seu movimento. Nesse sentido, foram construídos dois robôs: um com as rodas em modo diferencial, que permite atribuir maior velocidade a uma roda do que a outra de modo a que curve para o lado da roda com menor velocidade, tendo este robô tração nas rodas dianteiras.

Deste modo foi utilizado um motor para cada roda dianteira e um motor para controlar a posição da IP câmara consoante a curvatura do robô, procurando-se deste modo a sincronização dos movimentos da câmara e do veículo.

Na figura 2.3 e 2.4, encontra-se a imagem real do robô e um diagrama do seu funcionamento, respetivamente.

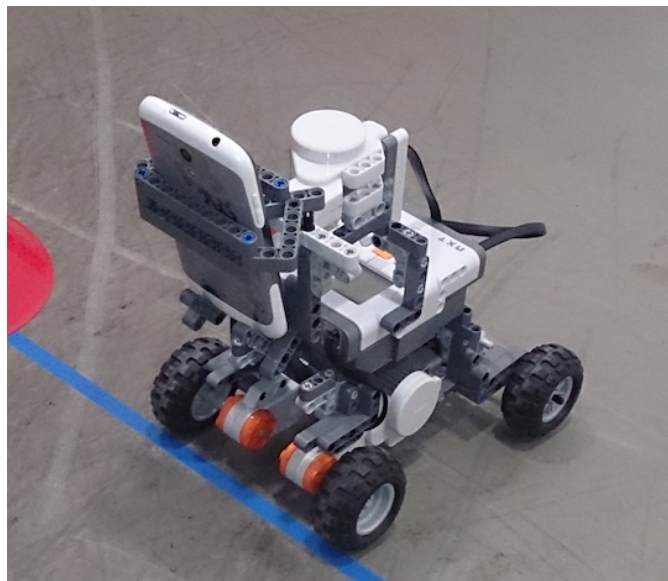


Figura 2.3: Robô com as rodas em modo diferencial

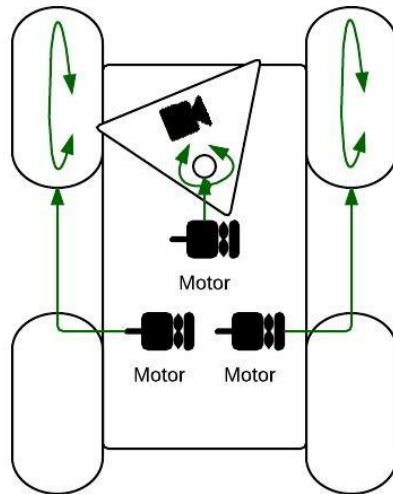


Figura 2.4: Esquemático do robô com as rodas em modo diferencial. As setas com apenas um sentido indicam as ligações dos motores as rodas e ao eixo da câmara. As setas com dois sentidos mostram a rotação dos mesmos. A diferença deste robô para o outro é o facto de que para este curvar terá que imprimir mais velocidade a uma roda do que a outra.

O outro robô que foi construído, possuía um eixo para que as rodas dianteiras virassem e era deste modo que este curvaria, sendo a tração deste robô nas rodas traseiras.

Neste veículo, foi utilizado um motor para as rodas traseiras que permite que este se mova, outro para as rodas dianteiras que possibilite que o robô curve e conseqüentemente mude de direção. Tal como o veículo anterior, possui um motor para controlar a posição da IP câmara consoante a curvatura do robô.

Na figura 2.5 e 2.6, encontra-se a imagem real do robô e um diagrama do seu funcionamento, respetivamente.



Figura 2.5: Robô com as rodas em modo direcional.

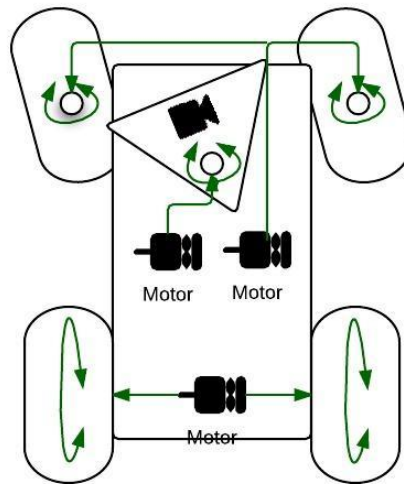


Figura 2.6: Esquemático do robô com as rodas em modo direcional. As setas com apenas um sentido indicam as ligações dos motores as rodas e ao eixo da câmara. As setas com dois sentidos mostram a rotação dos mesmos. A diferença deste robô para o outro é o facto de que para este curvar é utilizado um eixo que faz virar as rodas dianteiras.

A construção destes dois robôs visou, sobretudo, procurar conceber um veículo com uma boa mobilidade. O problema esteve sempre centrado no modo mais eficiente como este poderia curvar. Nas soluções apresentadas podem ser identificadas vantagens e inconvenientes. A grande desvantagem do primeiro robô em relação ao segundo é a maior dificuldade em calcular o ângulo de curvatura e a velocidade.

Embora o primeiro oferecesse mais dificuldade de cálculos foi este, o robô escolhido. Com efeito, após a realização de alguns testes chegou-se à conclusão que tinha vantagens bastante relevantes em relação ao segundo. A velocidade máxima que o primeiro robô consegue atingir é pelos menos o dobro da do segundo, pois o primeiro é puxado por dois motores enquanto que o segundo é apenas por um. Acresce o facto, de o primeiro ser mais leve que o segundo.

Uma das limitações do NXT assenta na impossibilidade de suportar mais de três motores. Atendendo a que no segundo robô estão instalados dois motores, um para virar as rodas dianteiras e outro para controlar a câmara IP, apenas poderá ser utilizado um terceiro motor para fornecer tração ao veículo, o que lhe vai conferir um fraco desempenho na sua mobilidade. Outra desvantagem do segundo em relação ao primeiro assenta na limitação do ângulo de curvatura que as rodas conseguiam atingir, ou seja o primeiro conseguia fazer curvas muito mais apertadas.

Merece ainda referência outra limitação deste equipamento, aquela que se relaciona com a precisão e calibração das rodas dianteiras do segundo robô que facilmente perdem a calibração, isto devido a mais uma limitação do NXT, a pouca precisão na rotação dos motores. Embora através de código se tivesse tentado minimizar essa perda de calibração. Para a IP câmara colocou-se o mesmo problema que acrescentava mais um

ponto desfavorável relativamente a este robô. Não fazia sentido utilizar este robô.

Face ao exposto e pesadas todas estas desvantagens do segundo robô, foi decidido utilizar o primeiro robô em detrimento do segundo.

## 2.4 Comunicação com o robô NXT

A comunicação entre o robô e o dispositivo móvel baseado em Android é feita através da tecnologia Bluetooth.

Esta comunicação tem como objetivo controlar o robô, através do dispositivo móvel. Para esse efeito, são enviadas instruções deste dispositivo para o robô, que tem como finalidade a ativação dos motores.

Foi possível esta comunicação utilizando as seguintes bibliotecas:

1. *android.bluetooth.BluetoothAdapter*;
2. *android.bluetooth.BluetoothDevice*;
3. *android.bluetooth.BluetoothSocket*;

De maneira a aceder à tecnologia Bluetooth do dispositivo móvel, foi necessário introduzir as seguintes permissões:

1. *android.permission.BLUETOOTH*;
2. *android.permission.BLUETOOTH\_ADMIN*;

As instruções são enviadas em notação hexadecimal para o NXT, este interpreta-as e só depois faz atuar os motores, consoante o que foi indicado pelo controlador. Foram desenvolvidas funções para controlar os motores, cujos parâmetros de entrada são os valores das velocidades para as respetivos motores de cada roda.

Os valores de entrada podem variar entre -128 e 128. Os valores negativos significam que o motor roda no sentido inverso ao dos valores positivos. A comunicação entre o controlador e o robô pode ter três estados. Quando se inicia a aplicação, o estado da comunicação é considerado desconectado, este é indicado pela cor do painel táctil de condução através da sua cor vermelha.

Para que o controlador se conecte com o robô é necessário premir o botão "Connect". Assim que este for premido, irá abrir uma janela onde vai procurar dispositivos com o Bluetooth ligado. Seguidamente, irá apresentar uma lista da qual terá que ser escolhido o robô. Assim que este for escolhido, o estado de conexão é alterado imediatamente para o estado considerado "a conectar", onde o controlador irá tentar conectar-se com o robô. Este estado é indicado pela cor do painel táctil de condução através da sua cor amarela. Quando o controlador se conecta com o robô, o estado passa a ser "conectado" e a cor do painel táctil de condução passa a ser verde, de modo a indicar que os dispositivos estão conectados.

Para regressar para o estado de comunicação considerado desconectado, basta o utilizador premir o botão “Disconnect”, terminando assim a ligação entre os dispositivos.

## 2.5 Transmissão de vídeo

O robô NXT possui uma câmara IP para que possa enviar a sua “visão” para o dispositivo móvel baseado em Android. Neste ponto, será tratado o modo como é efetuado o envio da visão da câmara IP para o dispositivo móvel.

Foi possível esta comunicação utilizando as seguintes bibliotecas:

1. *org.apache.http.HttpResponse;*
2. *org.apache.http.client.ClientProtocolException;*
3. *org.apache.http.client.methods.HttpGet;*
4. *org.apache.http.impl.client.DefaultHttpClient;*

De maneira a aceder ao endereço Url da câmara IP do dispositivo móvel, foi necessário introduzir a permissão *android.permission.INTERNET*.

Uma câmara IP, é um tipo de câmara de vídeo digital com a capacidade de enviar e receber dados através de uma rede de computadores e da Internet. Rede essa que foi criada através da tecnologia WI-FI.

Embora a maioria das câmaras que executam essa função sejam webcams, neste caso foi utilizado um smartphone. O smartphone utilizado como câmara é baseada em Android, o que facilita para o tornar uma câmara IP.

Através da “Play Store”, loja online de aplicações para android, foi descarregada uma aplicação, “IP Webcam”, de modo a tornar o simples smartphone numa IP câmara. A funcionalidade dessa aplicação consiste em gravar o vídeo através da câmara do smartphone e difundi-lo através da rede em tempo real.

A rede é criada pelo smartphone que é, igualmente, utilizado para correr a aplicação desenvolvida para controlar o robô. Este equipamento é utilizado então como “hotspot”, ou seja, o cérebro da rede.

O smartphone, ao ser usado como visão do robô, encontra-se ligado à rede. Ao lhe ser atribuído um endereço de IP, é através desse endereço de IP e de um “port” que irá difundir o vídeo. Neste caso o endereço de IP que lhe é atribuído é “192.168.43.4” e o “port” é “8000”.

O vídeo é então exibido através da página “http://192.168.43.4:8000”. Na aplicação de controlo do robô que foi criada, abre-se a página, sendo o vídeo exibido no fundo da aplicação.

O endereço Url utilizado foi "http://192.168.43.4:8000/videofeed". Como a "Sony" é o fabricante do smartphone utilizado o campo que teve que ser acrescentado a seguir ao "port" foi "videofeed", a fim de poder ir à página ilustrada na figura 2.7 e recolher apenas o vídeo.

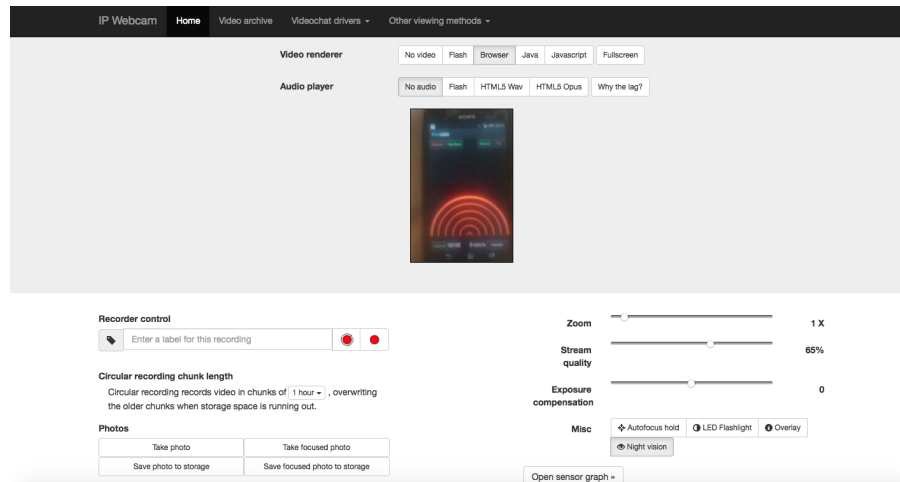


Figura 2.7: Página onde se recolhe a imagem da IP câmara.

Deste modo, a transmissão do vídeo no smartphone fica disponível para controlar o robô. Este vídeo tinha que ser exibido. Como o ecrã do smartphone utilizado tem apenas 5 polegadas, foi decidido utiliza-lo na totalidade para exibir o vídeo, tendo-se assim melhorado a percepção do cenário. O vídeo teve que ser transmitido com uma resolução de 240x160, pois foi a resolução que tinha melhor relação qualidade de imagem/atraso do vídeo.

Ficando o vídeo sem atraso perceptível em relação ao tempo real e com a qualidade imagem, que será exibida no ponto 2.9 que se refere à interface, onde se mostrou ser suficiente para se conseguir conduzir o robô através do smartphone de controlo, sem que haja contacto visual com o veículo.

## 2.6 Orientação automática da IP câmara

A orientação automática da IP câmara equivale à parte inteligente do robô em relação à sua visão.

Essa parte inteligente consiste em orientar a "visão" de modo a que esta acompanhe o movimento do robô, virando com o robô consoante a curvatura deste.

Optou-se por não utilizar uma visão estática, de modo a proporcionar ao utilizador uma condução mais natural. Com esta visão móvel, o tempo, as colisões e o numero de paragens serão menores comparando com uma visão estática.

A visão vira de um modo proporcional em relação à curvatura do robô, curvatura essa que é indicada pelo controlador. Ou seja, se o utilizador indicar ao robô para virar com um determinado ângulo, também está a informar que a visão deverá virar consoante esse

ângulo. Para mover a câmara é utilizado um motor do NXT. Este motor NXT roda sobre um eixo no sentido dos ponteiros do relógio ou no sentido oposto.

Na figura 2.8 está demonstrado o comportamento da IP câmara.

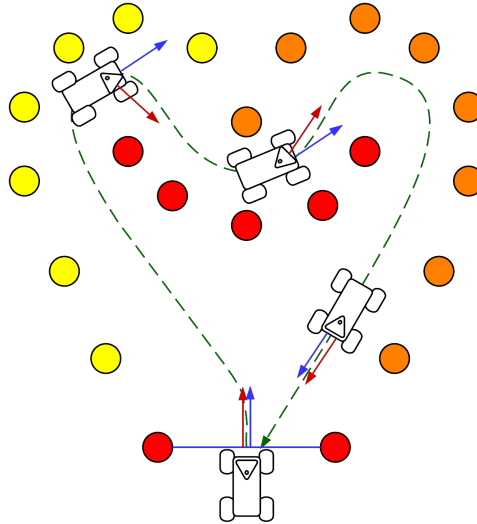


Figura 2.8: Comportamento da IP câmara consoante a curvatura. As setas vermelhas representam a direção da visão do robô, enquanto que as setas azuis representam a direção do robô a cada instante.

Ao Observar a figura 2.8, verifica-se o modo como a IP câmara acompanha a trajetória do robô.

Uma das dificuldades sentidas nesta fase do trabalho, relacionou-se com a falta de precisão deste motor, pois tinha uma margem de erro muito grande. Inicialmente a visão virava para todos os ângulos, ou seja iria estar constantemente a ler a posição do toque no controlador e se esta fosse diferente da anterior o controlador iria ordenar ao motor que rodasse, um determinado numero de milissegundos, para um lado ou para o outro consoante a posição anterior. Como a imprecisão se encontrava no motor, e para baixar a margem de erro, este teria que rodar o menor numero de vezes possível, mas acompanhando na mesma a curvatura do robô.

Para resolver este problema de imprecisão, foi desenvolvido um algoritmo em que a visão iria ter apenas nove posições. Ou seja, no painel táctil de condução, os ângulos variam entre zero e cento e oitenta graus, sendo as nove posições definidas uniformemente por esse espectro de ângulos. Assim, o painel táctil de condução está dividido em nove áreas e a visão só se ira mover se a posição do toque for numa área diferente da anterior, evitando assim as pequenas oscilações que fariam com que a posição da visão perdesse a calibração muito facilmente.

O algoritmo listing 2.1 serve para dividir o espectro de ângulos em nove áreas, atribuindo a cada área o seu ângulo central. Embora a primeira e última área sejam do mesmo tamanho das outras, na realidade estas duas áreas vão ser apenas metade das restantes, pois o espectro de ângulo apenas varia entre 0 e  $\pi$ .

Listing 2.1: Algoritmo para determinar as 9 zonas

```

1  for (int i = 0; i <= 8; i++) {
2      if (angle > (2*i-1)*Math.PI/16 && angle <= (2*i+1)*Math.PI/16) {
3          angle = (float) (2*i*Math.PI/16);
4          break;
5      }
6  }

```

Conforme ilustrada figura 2.9, as áreas variam entre os seguintes ângulos e o ângulos atribuídos a cada área são os seguintes:

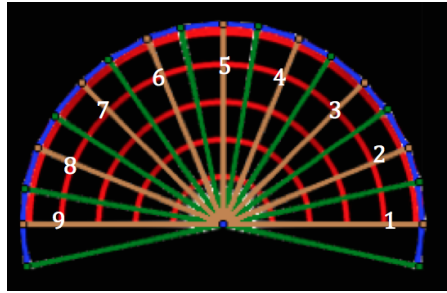


Figura 2.9: As nove áreas e o seus ângulo representativos

1.  $-\frac{1}{16}\pi < x < \frac{1}{16}\pi \rightarrow 0;$
2.  $\frac{1}{16}\pi < x < \frac{3}{16}\pi \rightarrow \frac{1}{8}\pi;$
3.  $\frac{3}{16}\pi < x < \frac{5}{16}\pi \rightarrow \frac{1}{4}\pi;$
4.  $\frac{5}{16}\pi < x < \frac{7}{16}\pi \rightarrow \frac{3}{8}\pi;$
5.  $\frac{7}{16}\pi < x < \frac{9}{16}\pi \rightarrow \frac{1}{2}\pi;$
6.  $\frac{9}{16}\pi < x < \frac{11}{16}\pi \rightarrow \frac{5}{8}\pi;$
7.  $\frac{11}{16}\pi < x < \frac{13}{16}\pi \rightarrow \frac{6}{8}\pi;$
8.  $\frac{13}{16}\pi < x < \frac{15}{16}\pi \rightarrow \frac{7}{8}\pi;$
9.  $\frac{15}{16}\pi < x < \frac{17}{16}\pi \rightarrow \pi;$

## 2.7 Condução em modo manual

A condução em modo manual consiste em dar ao utilizador o poder de ser ele a controlar tanto a direção como a velocidade do robô, ou seja este é controlado sem qualquer tipo de automatismo.

Neste modo, o robô é controlado através do painel táctil representado na figura 2.10. Este possui a forma de um semicírculo com várias semicircunferências concêntricas no seu interior.

As semicircunferências indicam os diferentes níveis de velocidade do robô. Quanto maior for o raio da semicircunferência, maior será a velocidade do robô, controlando-se deste modo a velocidade do robô.

O controlador irá estar constantemente a ler a posição do toque no painel táctil e, simultaneamente, identificará o ângulo que este ponto tem em relação ao centro do círculo, reconhecendo, deste modo, em qual das semicircunferências em que este se encontra. Ao calcular a distancia entre o centro do círculo e o ponto e através desse valor irá atribui uma velocidade ao robô.

Os ângulos das semicircunferências variam entre  $0^\circ$  e o  $180^\circ$ . O robô encontra-se a curvar para a direita quando os ângulos variam entre 0 graus e 90 graus, sendo o ângulo de curvatura do robô inverso ao ângulo do painel táctil. Assim, se o ângulo do painel táctil é  $0^\circ$  o ângulo de curvatura do robô será  $90^\circ$ . Sempre que o ângulo do painel táctil se aproxima de  $90^\circ$ , o ângulo de curvatura tende para  $0^\circ$ , significando que o robô se move em frente sem qualquer curvatura.

Por sua vez dos  $90^\circ$  aos  $180^\circ$  o robô está a curvar para a esquerda, sendo o ângulo de curvatura do robô, o ângulo do painel táctil retirando-se  $90^\circ$ . Se o ângulo do painel táctil é  $180^\circ$ , o ângulo de curvatura do robô será  $90^\circ$ . À medida que o ângulo do painel táctil se aproxima de  $90^\circ$ , o ângulo de curvatura tende para  $0^\circ$ , significando mais uma vez, que o robô se move em frente sem qualquer curvatura.

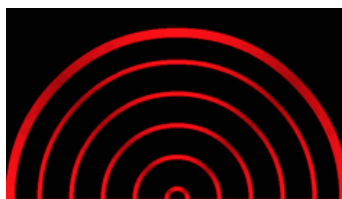


Figura 2.10: Interface da condução manual

O diagrama da figura 2.11 ilustra o funcionamento do modo manual.

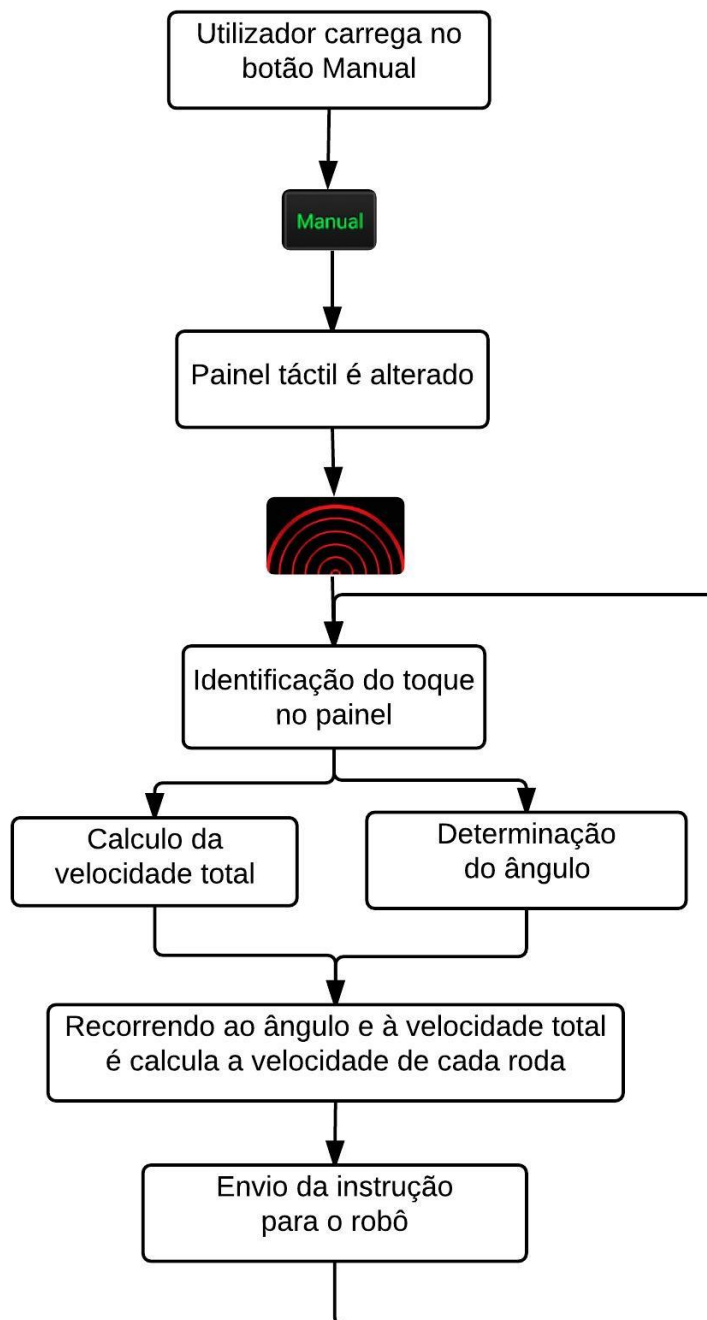


Figura 2.11: Funcionamento do modo manual

## 2.8 Condução em modo biológico

A condução em modo biológico consiste em ajudar o utilizador a controlar o robô com maior facilidade, pois neste modo o utilizador apenas tem que se preocupar com a direção que quer dar ao robô, através do painel táctil representado na figura 2.12.

A velocidade do robô, neste caso, é atribuída pelo controlador automaticamente, consoante a curvatura que o utilizador quiser dar ao robô.

O cálculo automático da velocidade do robô exprime-se em função do seu ângulo de curvatura, segundo a “Lei de Potência 2/3”[23]. Ou seja, se a curvatura da trajetória do robô for zero, deslocando-se em frente, mover-se-á à velocidade máxima de 30 cm/s. À medida que o ângulo de curvatura aumenta, sendo para a esquerda ou para a direita, a velocidade diminui com uma proporção de um terço. Assim, quando o ângulo de curvatura é 90°, que é a curvatura máxima, a velocidade do robô é um terço da velocidade máxima, 10 cm/s. A um aumento do ângulo de curvatura corresponderá sempre uma redução proporcional da velocidade, o que torna estas grandezas inversamente proporcionais.

Neste modo, o robô é controlado através do painel táctil anterior, sendo que este possui a forma de uma semicircunferência.

Neste caso, como o utilizador não tem o poder de controlar a velocidade, a semicircunferência representa exclusivamente a direção.

Neste modo, o controlador irá também estar constantemente a ler a posição do toque no painel táctil, mas agora apenas irá retirar o ângulo que este ponto tem em relação ao centro da semicircunferência. O ângulo retirado serve para calcular não só a curvatura do robô como também a sua velocidade.

Os ângulos da semicircunferência variam entre 0° e o 180°, tal como no modo anterior, ângulos estes que dos 0° aos 90°, significa que o robô está a curvar para a direita, sendo o ângulo de curvatura do robô inverso ao ângulo do painel táctil, tal como no modo anterior. O mesmo acontece com a velocidade, pois esta depende do ângulo de curvatura. Se o ângulo do painel táctil for 0°, a velocidade será um terço da velocidade quando o ângulo do painel táctil for 90°.

Por sua vez, e tal como no modo anterior, dos 90° aos 180° significa que o robô está a curvar para a esquerda, sendo o ângulo de curvatura do robô, o ângulo do painel táctil menos 90°. A velocidade neste caso também é máxima nos 90° e será um terço nos 180°.



Figura 2.12: Interface da condução biológico

A figura 2.13 ilustra o diagrama que descreve o funcionamento do modo biológico.

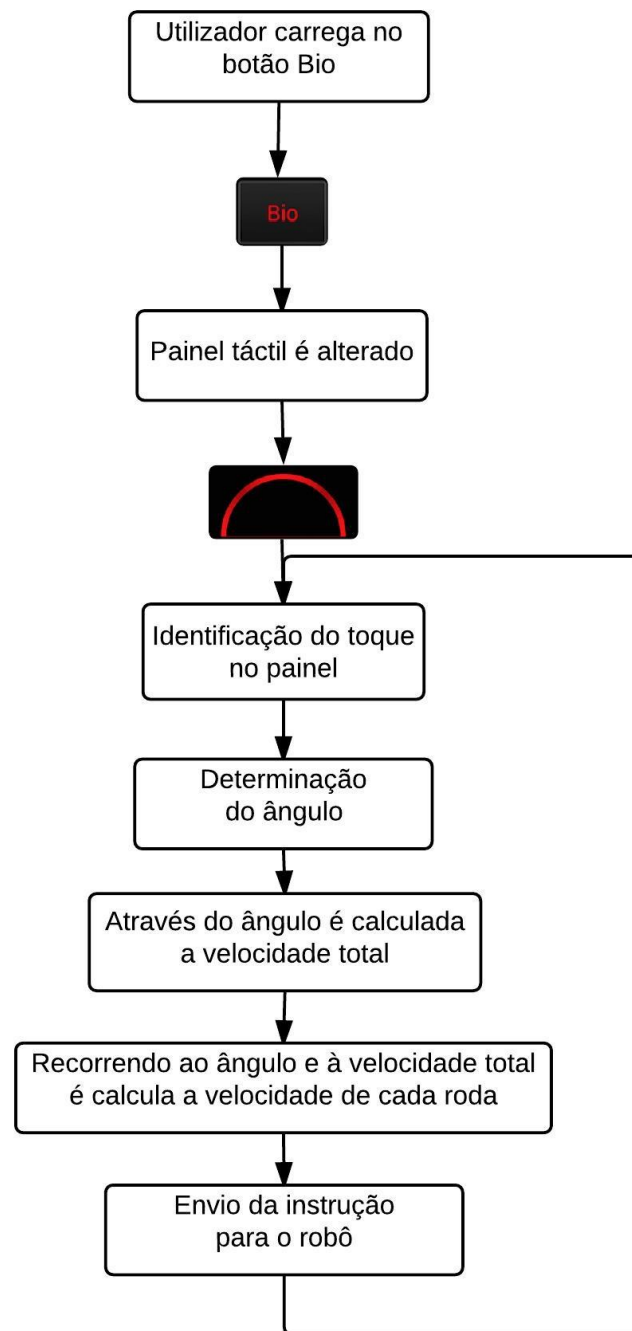


Figura 2.13: Funcionamento do modo biológico

Para que fosse possível estes dois modos de condução, teve que ser criado um algoritmo que atribui-se a velocidade a cada um deles. Este foi desenvolvido da seguinte forma, representada no listing 2.2:

Listing 2.2: Algoritmo que atribui a velocidade ao robô

```

1  if (Manual==1) {
2    power=(float)Math.sqrt (nx * nx + ny * ny);
3  }
4  else{
5    if(angle<=Math.PI/2.0f && angle>=0)
6      power=(float) (2.4f* ((4/ (3* (Math.PI))) *angle+(1.0f/3)));
7    else{
8      if(angle<=Math.PI && angle>Math.PI/2.0f)
9        power=(float) (2.4f* ((4/ (3* (Math.PI))) *-(angle- (Math.PI)) +(1.0f/3)));
10     else
11       power=0;
12   }
13 }

```

A figura 2.14 explica o algoritmo do listing 2.2 através de um diagrama.

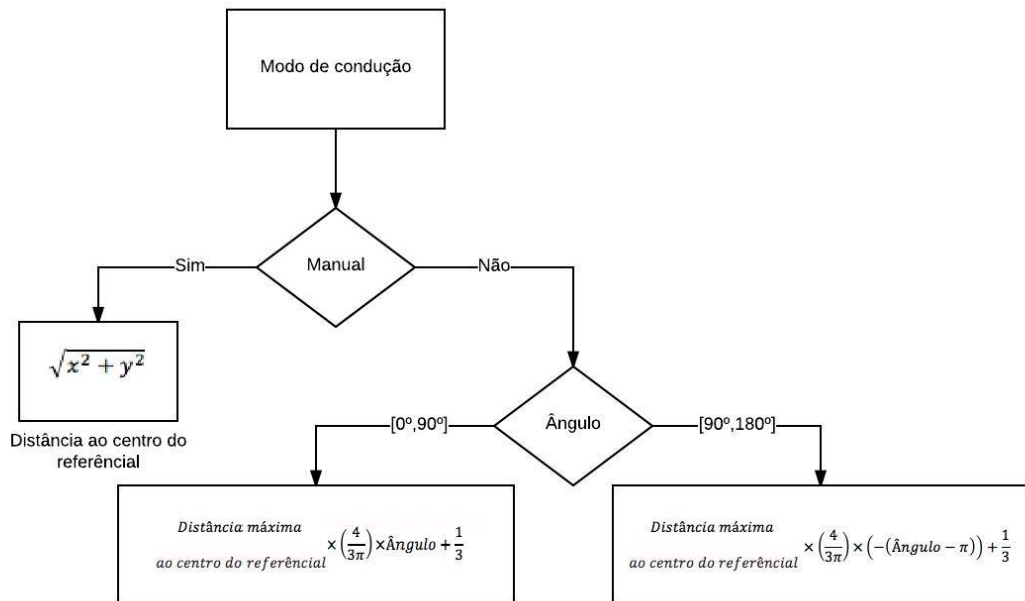


Figura 2.14: Diagrama do algoritmo que atribui a velocidade ao robô

## 2.9 Interface

Conforme referido antes, foi desenvolvida uma aplicação em Android com o intuito de controlar o robô. A aplicação possui uma interface gráfica. Esta interface, conforme ilustrado na figura 2.15, possui funcionalidades que vão ser descritas nos próximos pontos.

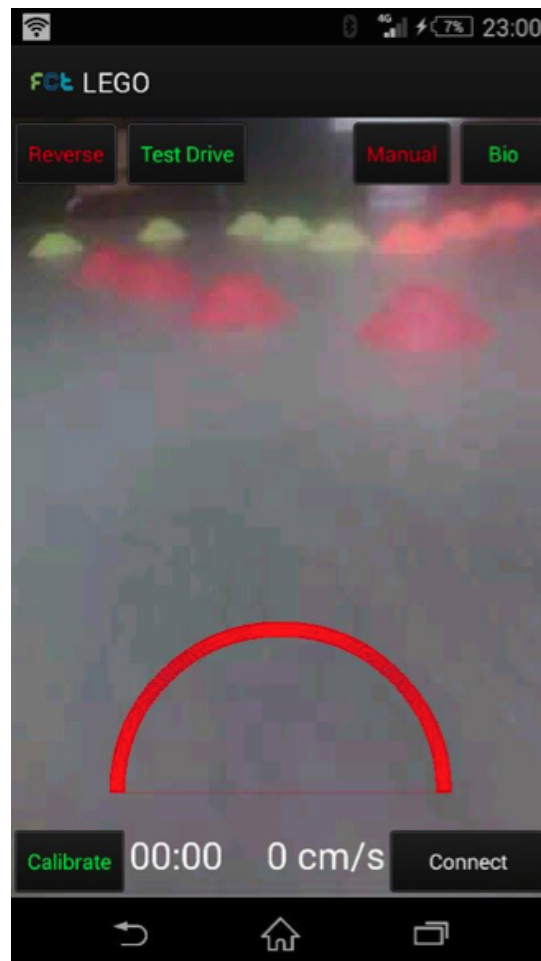


Figura 2.15: Interface

### 2.9.1 Fundo

Conforme se observa na figura 2.15, no fundo da aplicação, é exibida uma imagem transmitida pela IP Câmera.

Esta imagem representa a “visão” do robô e para que este possa ser controlado à distancia e sem contacto visual, a imagem é exibida em tempo real, pois o utilizador deve ter plena noção da posição onde se encontra o robô no preciso momento em que está a controla-lo.

### 2.9.2 Botão "Calibrate"

O botão "Calibrate", representado na figura 2.16, encontra-se no canto inferior esquerdo, como se pode observar na figura 2.15, e serve para calibrar a direção da IP câmara.

As letras deste botão podem ter duas cores, o verde significa que este pode ser premido, desempenhando a função para que foi concebido, e o vermelho alertando que a calibração está em curso, logo o botão não poderá ser premido. Ou seja, a cor vermelha surge assim que o botão é premido e é iniciada a calibração, para deste modo avisar o utilizador que a calibração está em curso. Assim que a calibração terminar, as letras do botão "Calibrate" voltam a mudar para verde, e aí este pode voltar a ser premido.

De salientar que mesmo que utilizador tente premir o botão com as letras a vermelho este não funcionará.

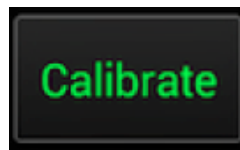


Figure 2.16: Botão "Calibrate"

### 2.9.3 Botão "Connect/Disconnect"

O botão "Connect/Disconnect", representado na figura 2.17, encontra-se no canto inferior direito, como se pode observar na figura 2.15, e serve para iniciar a conexão ao robô NXT.

Este botão pode ter dois estados, "Connect" que significa que ao ser premido este botão irá abrir uma janela para pesquisar dispositivos através de "Bluetooth", de modo a que se possa então ligar ao robô.

"Disconnect" é o outro estado, este ao ser premido irá fechar a ligação entre o dispositivo móvel e o robô.

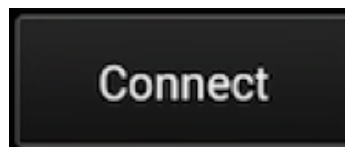


Figure 2.17: Botão "Connect/Disconnect"

### 2.9.4 Botão “Reverse”

O botão “Reverse”, representado na figura 2.18, encontra-se no canto superior esquerdo, como se pode observar na figura 2.15, e serve para inverter a marcha do robô.

As letras deste botão podem ter também duas cores, embora com uma lógica diferente do botão “Calibrate”. Neste caso, as cores indicam o estado da marcha do robô. O verde significa que a marcha a ré está ativa e o vermelho que está desativada.

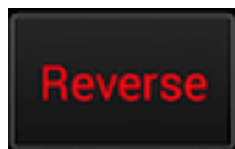


Figure 2.18: Botão “Reverse”

### 2.9.5 Botão “Test Drive”

O botão “Test Drive”, representado na figura 2.19, encontra-se do lado direito do botão “Reverse”, como se pode observar na figura 2.15. Serve para iniciar o teste experimental e guardar num ficheiro, os dados resultantes deste.

Este botão ao ser premido irá mostrar uma mensagem no centro do ecrã, “READY”, para avisar o utilizador de que o teste irá ter início. De seguida, irá iniciar uma contagem decrescente, “3, 2, 1, GO”. Assim que surge a mensagem, “GO”, o cronómetro inicia a contagem do tempo. O utilizador pode iniciar o teste e de imediato começam a ser guardados os dados.

No final do teste, este botão deverá ser novamente premido, mostrando a mensagem “FINISH” informando que o teste está terminado e os dados guardados no ficheiro.

Este botão é também controlado por cores. O verde significa que um teste pode ser iniciado e o vermelho significa que o botão ao ser premido irá terminar o teste em curso.

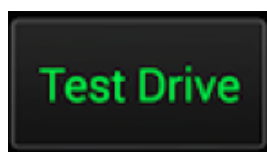


Figure 2.19: Botão “Test Drive”

### 2.9.6 Botão “Bio” e botão “Manual”

O botão “Bio”, representado na figura 2.20, encontra-se no canto superior direito, como se pode observar na figura 2.15, serve para alterar o modo de condução, modo este que possui uma velocidade assistida, deixando apenas para o utilizar o controlo da direção do robô.

Este botão está interligado com o botão “Manual” que se encontra ao seu lado esquerdo. O botão “Manual” por sua vez serve para alterar a condução para o modo manual, onde o utilizador tem que controlar, tanto a direção como a velocidade.

São estes dois botões controlados também pelas cores verde e vermelho. Quando as letras de um dos botões se encontram a verde, as do outro estão a vermelho. Se as letras de uma das mensagens se apresentam com a cor verde, significa que esse modo se encontra ativo. Para mudar de modo basta carregar no botão com as letras a vermelho e automaticamente as cores das letras dos botões são invertidas e o modo de condução alterado.



Figure 2.20: Botão “Bio” e botão “Manual”

### 2.9.7 Cronometro

O cronometro, representado na figura 2.21, encontra-se ao lado direito do botão “Calibrate”. O mostrador serve para informar o utilizador do tempo que já gastou no seu teste. A sua contagem é iniciada assim que surge a mensagem “GO” e terminada ao aparecer a mensagem “FINISH”, em seguida, o tempo é guardado no ficheiro.

The image shows a digital timer display. It consists of a black rectangular box with the text '00:09' in a white, sans-serif font. The digits are evenly spaced and centered within the box.

Figure 2.21: Cronometro

### 2.9.8 Velocímetro

O velocímetro, representado na figura 2.22, encontra-se ao lado esquerdo do botão “Connect/ Disconnect”, este mostrador serve para dar a conhecer ao utilizador a velocidade a que o robô se move. A velocidade do robô pode variar entre 0 cm/s e 30 cm/s.

The image shows a digital speedometer display. It consists of a black rectangular box with the text '22 cm/s' in a white, sans-serif font. The text is centered within the box.

Figure 2.22: Velocímetro

### 2.9.9 Painel tátil de condução

O Painel tátil de condução encontra-se por cima dos mostradores de tempo e velocidade. É através deste painel que o utilizador vai controlar o robô, com toque ou por arrastamento. Este painel lê constantemente a posição em que está a ser efetuado o toque por parte do utilizador. Possui duas imagens consoante o modo de condução. No modo de condução com a velocidade assistida, onde o utilizador não pode controlar esta grandeza, a imagem será um arco (figura 2.23).



Figure 2.23: Interface da condução biológico

O arco servirá exclusivamente para controlar a direção do robô. Por exemplo, com o toque no topo do arco o robô desloca-se em frente. À medida que se arrasta o dedo para a esquerda ou para a direita, o robô irá consequentemente se direcionar para cada um desses lados, respetivamente. Quanto mais o toque se afastar do topo do arco, mais o robô irá curvar.

No modo manual, onde o utilizador tem a tarefa de controlar não só a direção mas também a velocidade, a imagem no painel irá alterar-se para vários arcos (figura 2.24).



Figure 2.24: Interface da condução manual

Neste caso, como o utilizador é solicitado a controlar não só a direção como também a velocidade, a imagem é alterada de modo a atender a essa necessidade. O toque no painel irá fazer com que o robô em termos de direção se comporte da mesma maneira que no modo anterior. A grande diferença neste modo é que o toque no arco maior irá fazer com que o robô se desloque à velocidade máxima de 30cm/s. À medida que se arrasta o dedo para os arcos menores, a velocidade diminuirá, ou seja, quanto mais pequeno for o arco menor será a velocidade do robô.

O painel serve ainda para indicar se o robô está conectado com o dispositivo móvel baseado em Android conforme referido antes. Com essa finalidade, o painel poderá ter três cores: verde, que indica que estão conectados; amarelo, que indica que se estão a conectar; e vermelho, que indica que não estão conectados.

## EXPERIMENTAÇÃO

### 3.1 Introdução da experiência

Após a conclusão da implementação do sistema de assistência ao controlo remoto de um robô NXT, e estando este completamente operacional, chegou a altura de elaborar uma experiência para então perceber se de facto a condução biológica tem alguma vantagem em relação à condução manual.

A experiência passa pelos utilizadores conduzirem o robô num percurso através de um dispositivo móvel baseado em Android, sem qualquer contacto visual direto tanto com o robô como com o percurso. O robô é conduzido apenas através da imagem enviada pela IP câmara (situação de teleoperação), que é exibida no fundo da interface da aplicação desenvolvida.

O objetivo de cada utilizador, seja qual for o modo de condução, é concluir o percurso no menor tempo possível e também com o menor número de colisões. Seriam estes os principais fatores que iriam ditar se havia de facto vantagem em utilizar o modo de condução biológico em detrimento do modo de condução manual.

Para elaborar a experiência, houve necessidade de ser criado um percurso que se adequasse às dimensões do robô, com algumas curvas para salientar os eventuais benefícios do modo de condução biológico, pois é nas curvas que se poderá notar alguma diferença entre os desempenhos dos modos de condução biológico e manual. Seria relevante saber se o facto do modo biológico reduzir automaticamente a velocidade consoante a curvatura se traduz numa vantagem, ou não. Em todo o caso, o percurso deveria ser equilibrado de forma que fosse relativamente acessível para que os utilizadores o conseguissem concluir com sucesso, pois os mesmos foram escolhidos aleatoriamente, não tendo estes, em princípio, qualquer tipo de experiência em atividades do género.

O objetivo desta experiência é verificar se existe alguma vantagem, em termos de tempo

gasto pelos utilizadores para concluir o percurso, numero de colisões, numero de vezes que o utilizador imobiliza o robô e ainda a suavidade das trajetórias, no modo biológico em relação ao modo manual. Pelo facto do modo biológico regular automaticamente a velocidade em função da trajetória do robô, colocamos a hipótese de que o desempenho do utilizador deverá ser melhor nesta condição do que no modo manual.

A experiência anteriormente referida é considerada a experiência principal, pois é esta que irá determinar se há ou não alguma vantagem do modo biológico em relação ao modo manual. No entanto, antes ainda da implementação, foi efetuada uma experiência auxiliar, que serviu para determinar a velocidade do robô. Seguindo a ordem cronológica dos factos, será então esta experiência auxiliar a primeira a ser abordada, por ter ocorrido mesmo antes da implementação.

## 3.2 Calculo da velocidade do robô

Numa fase inicial do trabalho foi pensado qual a melhor maneira de abordar o problema proposto, foi então nesta fase que surgiu uma questão que iria dificultar a interação com o robô NXT.

Esta questão prendia-se com o facto dos valores enviados para os motores do robô NXT, quando em movimento, não resultarem num comportamento linear, ou seja, de inicio notou-se que à medida que se enviavam valores maiores para o robô, a velocidade dos seus motores aumentava, não de uma forma linear, mas sim de uma forma exponencial.

Conforme referido antes, a velocidade no modo de condução biológico é atribuída pelo controlador automaticamente consoante a curvatura que o utilizador escolhe dar ao robô. A velocidade máxima, quando o robô se move em frente, vai diminuindo numa proporção de um terço à medida que a curvatura aumenta. Portanto é importante ter uma noção exata da velocidade à que o robô se está a mover, para que se possa fazer os cálculos de modo a que esta condição se verifique.

Para calcular a velocidade foram então realizados a seguintes passos:

1. Foi marcado no chão um trajeto em linha reta com um comprimento de 2 metros.
2. O robô foi colocado numa das extremidades da reta.
3. Por fim, foram efetuados 5 testes em 5 velocidades diferentes, 88,78,68,58,48, valores estes que correspondem aos valores que são enviados para o robô, onde foi cronometrado o tempo que o robô demorava a percorrer dois metros nas diferentes velocidades.

Os resultados obtidos são apresentados na tabela 3.1:

Tabela 3.1: Resultados obtidos da experiência para determinar a velocidade

<b>Coefficiente</b>	<b>Tempo (ms)</b>	<b>Velocidade ( cm / s)</b>
88	6590	30,35
78	8190	24,42
68	9090	22,00
58	10720	18,66
48	14370	13,92

A partir resultados obtido foram construídas duas tabelas (3.3 e 3.5) para verificar que a relação entre os valores enviado para o robô e o tempo que este demorou a percorrer dois metros lhe atribuíam era exponencial e não linear.

A tabela 3.3 foi criada a partir da equação  $Y = bX^a$  com os parâmetros da tabela 3.2:

Tabela 3.2: Equação  $Y = bX^a$

Valor	Variável
1545291,135	b
-1,214281743	a

Tabela 3.3: Valores seguindo uma reta exponencial

Coefficiente	Tempo (ms)	Velocidade (cm/s)	Coefficiente	Tempo (ms)	Velocidade (cm/s)
2	666001	0,30	62	10293	19,43
4	287038	0,70	64	9904	20,19
6	175435	1,14	66	9541	20,96
8	123710	1,62	68	9201	21,74
10	94347	2,12	70	8883	22,52
12	75610	2,65	72	8584	23,30
14	62703	3,19	74	8303	24,09
16	53317	3,75	76	8038	24,88
18	46212	4,33	78	7789	25,68
20	40662	4,92	80	7553	26,48
22	36219	5,52	82	7330	27,29
24	32587	6,14	84	7119	28,10
26	29569	6,76	86	6918	28,91
28	27024	7,40	88	6728	29,73
30	24852	8,05	90	6547	30,55
32	22979	8,70	92	6374	31,38
34	21348	9,37	94	6210	32,21
36	19917	10,04	96	6053	33,04
38	18651	10,72	98	5903	33,88
40	17525	11,41	100	5760	34,72
42	16517	12,11	102	5623	35,57
44	15610	12,81	104	5492	36,41
46	14790	13,52	106	5367	37,27
48	14045	14,24	108	5246	38,12
50	13365	14,96	110	5131	38,98
52	12744	15,69	112	5020	39,84
54	12173	16,43	114	4913	40,71
56	11647	17,17	116	4810	41,58
58	11161	17,92	118	4712	42,45
60	10711	18,67	120	4616	43,32

Para um comportamento linear dos motores, utilizou-se a equação  $Y = mX + b$ , cujos parâmetros se encontram na tabela 3.4. Os resultados obtidos estão na tabela 3.5.

Tabela 3.4: Equação  $Y = mX + b$ 

Valor	Variável
-180,9	m
22093,2	b

Tabela 3.5: Valores seguindo uma reta linear

Coefficiente	Tempo (ms)	Velocidade (cm/s)	Coefficiente	Tempo (ms)	Velocidade (cm/s)
2	21731,4	9,20	62	10877,4	18,39
4	21369,6	9,36	64	10515,6	19,02
6	21007,8	9,52	66	10153,8	19,70
8	20646	9,69	68	9792	20,42
10	20284,2	9,86	70	9430,2	21,21
12	19922,4	10,04	72	9068,4	22,05
14	19560,6	10,22	74	8706,6	22,97
16	19198,8	10,42	76	8344,8	23,97
18	18837	10,62	78	7983	25,05
20	18475,2	10,83	80	7621,2	26,24
22	18113,4	11,04	82	7259,4	27,55
24	17751,6	11,27	84	6897,6	29,00
26	17389,8	11,50	86	6535,8	30,60
28	17028	11,75	88	6174	32,39
30	16666,2	12,00	90	5812,2	34,41
32	16304,4	12,27	92	5450,4	36,69
34	15942,6	12,55	94	5088,6	39,30
36	15580,8	12,84	96	4726,8	42,31
38	15219	13,14	98	4365	45,82
40	14857,2	13,46	100	4003,2	49,96
42	14495,4	13,80	102	3641,4	54,92
44	14133,6	14,15	104	3279,6	60,98
46	13771,8	14,52	106	2917,8	68,54
48	13410	14,91	108	2556	78,25
50	13048,2	15,33	110	2194,2	91,15
52	12686,4	15,76	112	1832,4	109,15
54	12324,6	16,23	114	1470,6	136,00
56	11962,8	16,72	116	1108,8	180,38
58	11601	17,24	118	747	267,74
60	11239,2	17,79	120	385,2	519,21

Das tabelas resultou uma descrição do comportamento dos motores que se ilustra na figura 3.1:

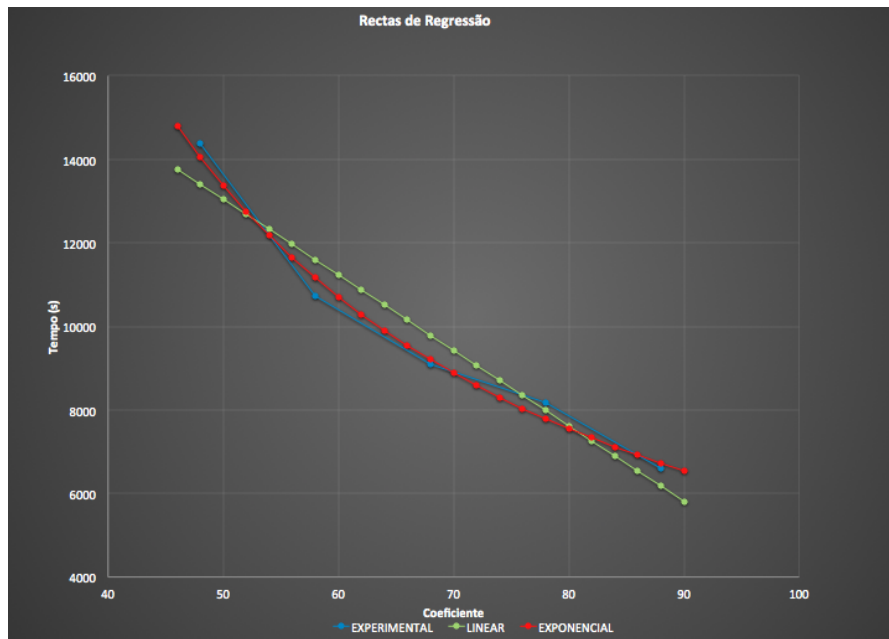


Figura 3.1: Retas de regressão

Como se pode observar no gráfico desta figura, a reta exponencial é a que melhor se aproxima da reta composta pelos dados experimentais. Foram utilizados os dados obtidos através da reta exponencial para calcular a velocidade.

Desses dados resultou o seguinte gráfico ilustrado na figura 3.2:

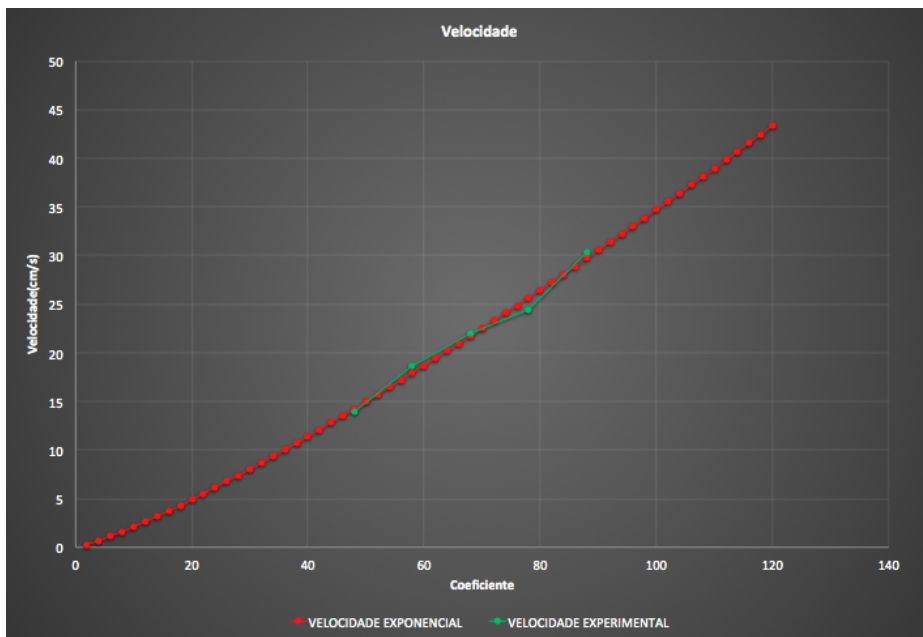


Figura 3.2: Velocidade

A velocidade, nas tabelas e gráficos, diz respeito à media da velocidade dos dois motores, que neste caso particular, têm a mesma velocidade, significando que o robô se está a mover em frente.

Como o robô tem as rodas em modo diferencial, para que possa curvar, a velocidade de uma roda tem que ser superior à da outra. Para que este curve para a esquerda a velocidade da roda da direita tem que ser superior à da esquerda, e vice versa para virar para a direita.

Foi então criada uma tabela que mostra a velocidade total do robô enquanto este está a curvar. Através desta foi decidido qual a melhor gama de valores a utilizar. Esta tabela pode-se encontrar no anexo (figura A.1).

Assim, foi possível encontrar uma equação que associasse uma velocidade ao valor a ser enviado para o robô.

O valor a ser enviado ao robô depende da velocidade e da direção que se lhe quer dar. A primeira coisa a ter em conta é a velocidade. Depois consoante a velocidade, existe uma gama de combinações de valores possível para que o robô circule aquela velocidade. É neste ponto onde se tem que ter em conta o ângulo de curvatura que se quer dar ao robô. Com esse ângulo, recorrendo ao algoritmo desenvolvido, vai-se procurar na gama de valores daquela velocidade qual o valor a atribuir à roda, do lado para o qual se quer curvar, de modo a que o veiculo tenha a curvatura deseja. De seguida, após se ter o valor a atribuir a uma roda, o valor da outra irá ser retirado por meio de uma equação recorrendo à velocidade desejada e ao valor conhecido.

O algoritmo foi implementado conforme ilustrado no listing 3.1:

Listing 3.1: Algoritmo que calcula a velocidade ao robô

```

1 A=36.25f*power-27.0f;
2 B=0.37069f*A+8.75862f; //velocidade do robo
3 // DIREITA-FRENTE
4 if (angle > 0 && angle <= Math.PI / 2f) {
5     r = (float) (60.0f * angle / Math.PI)+A;           //roda direita
6     l = (float) (2.72141947f+B-0.18565f*r)/0.18564962f; //roda esquerda
7 // ESQUERDA-FRENTE
8 } else if (angle > Math.PI/ 2f && angle <= Math.PI) {
9     l = (float) (60.0f * (Math.PI - angle) / Math.PI)+A; //roda esquerda
10    r = (float) (2.72141947f+B-0.18565f*l)/0.18564962f; //roda direita
11 }

```

Este algoritmo está também ilustrado no fluxograma da figura 3.3.

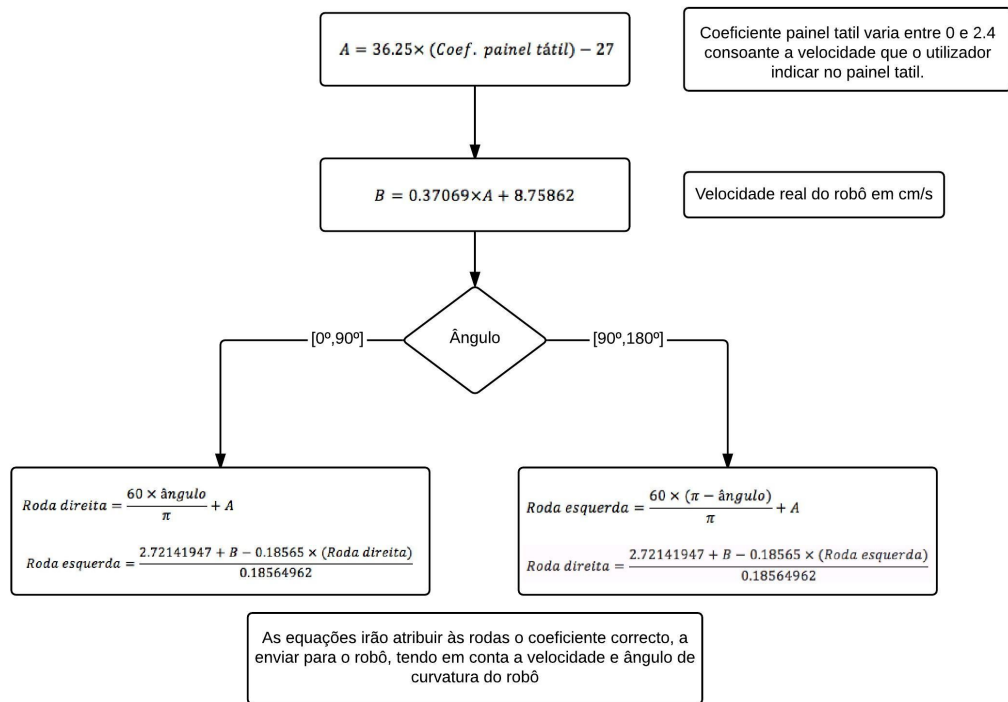


Figura 3.3: Algoritmo que calcula a velocidade ao robô

### 3.3 Material/Método

Neste ponto irá ser explicado o modo como foi elaborada a experiência principal.

Para que esta experiência fosse possível, foi necessário três elementos essenciais: o procedimento, o percurso e os participantes.

Sobre o percurso, irá ser explicado o modo como este foi desenhado, os materiais que foram utilizados e o porquê de assim ser.

No que diz respeito aos participantes, irá ser abordado o modo como estes foram elucidados do que se tratava a experiência, como a iam executar, como foram distribuídos, a ordem pela qual efetuaram os testes e como se recolheram os dados da experiência.

No procedimento explicar-se-á o modo como a experiência foi feita e as suas etapas.

#### 3.3.1 Percurso

O percurso escolhido para a elaboração da experiência foi desenhado de maneira a que se adequasse as dimensões do robô, tendo uma distância de aproximadamente 7 metros. Este foi desenhado de modo a que se pudesse observar os benefícios, se os houver, do modo de condução biológico. Para tal teria que ter algumas curvas, pois é nas curvas que se poderá notar alguma diferença entre os desempenhos dos modos de condução biológico e manual. Assim foi escolhido o percurso ilustrado na figura 3.4:

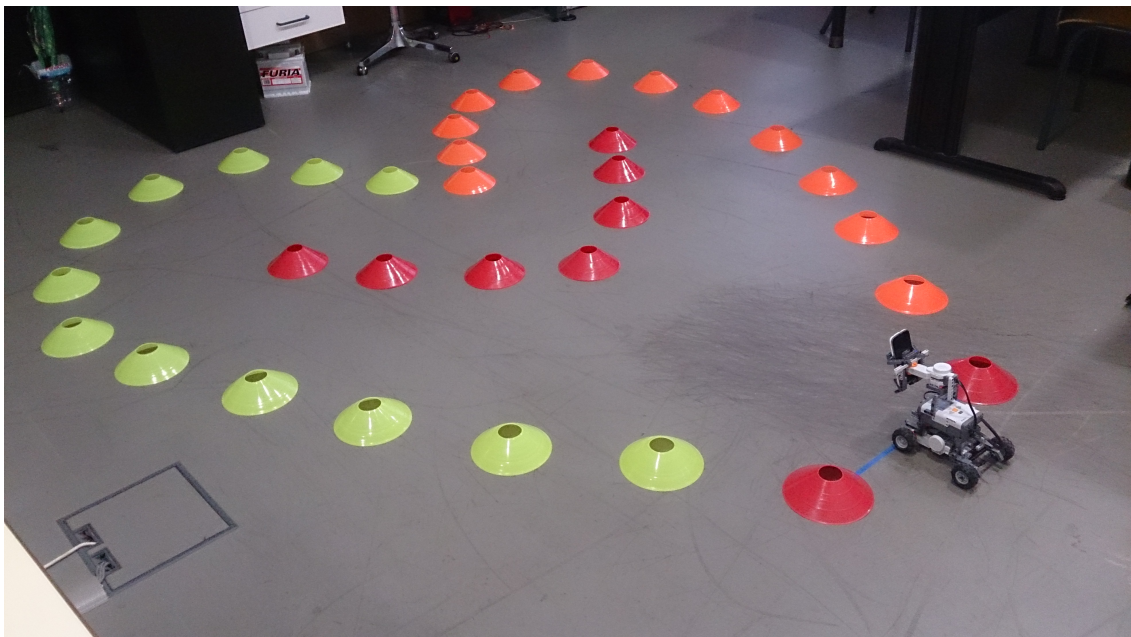


Figura 3.4: Percurso

Como se pode observar nesta figura, o percurso é composto inicialmente por uma reta, no seu final, tem-se uma curva apertada de aproximadamente  $150^\circ$ , de seguida uma contracurva de  $90^\circ$ , depois, novamente outra contracurva apertada de aproximadamente  $150^\circ$ , terminando numa reta. O percurso tem apenas três curvas, de modo a que a tarefa

para o utilizador não se torne demasiado complexa, mas ao mesmo tempo estas curvas são bastante acentuadas, de modo a que se acentua as esperadas vantagens do modo de condução biológico em relação ao modo de condução manual.

É possível observar que o percurso é composto por pinos de diversas cores. Estes servem para delimitar o percurso e é neles que eventualmente ocorrem as colisões. Os dois pinos vermelhos junto ao robô representam o início e o fim do percurso, para salientar melhor este facto foi colocada uma linha azul a uni-los. Do lado esquerdo, foram colocados pinos amarelos e do lado direito pinos laranja. Foram colocados deste modo, pois o percurso não tem um sentido fixo, sendo suposto os utilizadores executado nos dois sentidos. Assim, através das cores, permite-se que os utilizadores possam ter uma melhor noção do percursos através da imagem exibida na IP câmara. Os pinos vermelhos, que se encontram entre os pinos amarelos e laranjas, foram colocados naquela posição para delimitar as três curvas, foi utilizada uma cor diferente para uma melhor compreensão do percurso por parte dos utilizadores.

### **3.3.2 Participantes**

Para participar nesta experiência foram convidados vinte indivíduos. Estes tinham idades compreendidas entre os 22 e 27 anos. Eram todos estudantes universitários do sexo masculino, não possuindo estes, qualquer tipo de incapacidade motora.

Em termos de visão, todos eles viam corretamente, embora alguns, através de meios que permitissem corrigir alguma incapacidade visual.

Os participantes foram informados da finalidade do estudo e verbalmente deram o seu consentimento .

### **3.3.3 Procedimento**

No início da experiência foi explicado, aos participantes, que teriam como objetivo efetuar o percurso no menor tempo e com o mínimo de colisões.

Teriam que o fazer oito vezes, quatro utilizado o modo manual e outras quatro o modo biológico, dessas quatro de cada modo, duas teriam que ser no sentido dos ponteiros do relógio e as outras duas no sentido oposto, como é exemplificado nas figuras 3.5 e 3.6.

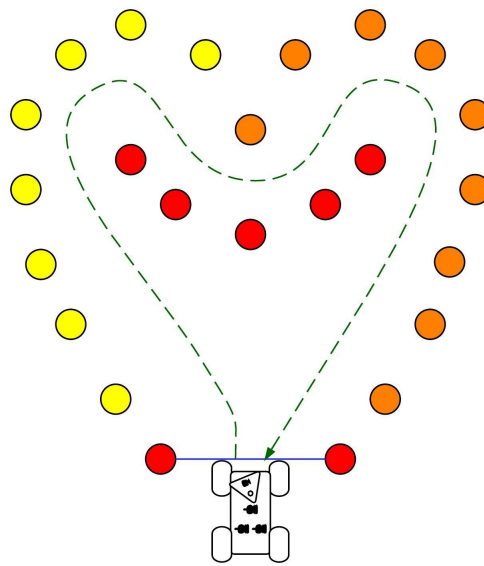


Figura 3.5: Percurso efetuado no sentido dos ponteiros do relógio

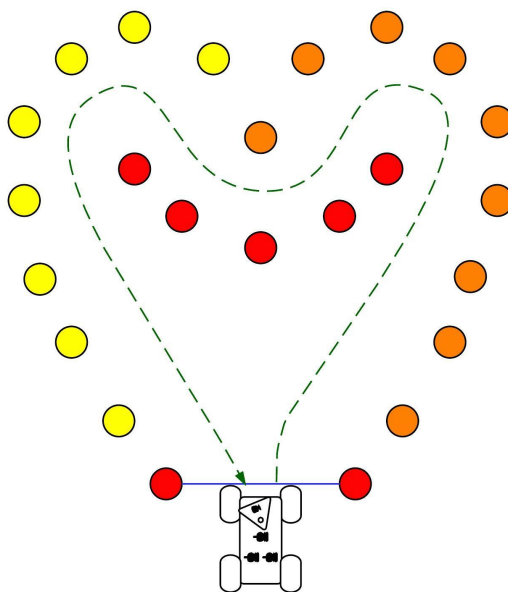


Figura 3.6: Percurso efetuado no sentido contrário dos ponteiros do relógio

Os indivíduos foram divididos em quatro grupos, onde dois deles iniciariam a experiência no modo de condução manual e os outros dois no modo biológico. Os dois grupos do mesmo modo de condução diferenciavam-se pelo facto de uns começarem no sentido dos ponteiros do relógio e os outros no sentido oposto.

As experiências são efetuadas pela ordem dos grupo, para que o individuo seguinte após ter observado a experiência anterior não tenha vantagem. Esta foi a maneira encontrada para combater o efeito de aprendizagem.

A ordem dos percursos e o indivíduos foram distribuídos pelos grupos conforme discriminado na tabela 3.6:

Tabela 3.6: Método experimental

	Grupo 1	Grupo 2	Grupo 3	Grupo 4
Ordem dos Indivíduos	Indivíduo 1	Indivíduo 2	Indivíduo 3	Indivíduo 4
	Indivíduo 5	Indivíduo 6	Indivíduo 7	Indivíduo 8
	Indivíduo 9	Indivíduo 10	Indivíduo 11	Indivíduo 12
	Indivíduo 13	Indivíduo 14	Indivíduo 15	Indivíduo 16
	Indivíduo 17	Indivíduo 18	Indivíduo 19	Indivíduo 20
Ordem dos percursos	Manual ↗	Manual ↖	Biológico ↗	Biológico ↖
	Manual ↖	Manual ↗	Biológico ↖	Biológico ↗
	Manual ↗	Manual ↖	Biológico ↗	Biológico ↖
	Manual ↖	Manual ↗	Biológico ↖	Biológico ↗
	Biológico ↗	Biológico ↖	Manual ↗	Manual ↖
	Biológico ↖	Biológico ↗	Manual ↖	Manual ↗
	Biológico ↗	Biológico ↖	Manual ↗	Manual ↖
	Biológico ↖	Biológico ↗	Manual ↖	Manual ↗

No início de cada experiência, foi mostrado a cada indivíduo o ambiente com o qual iria interagir, o percurso e o robô. De seguida, este foi conduzido a uma cadeira, de onde não tinha contacto visual direto com o ambiente. Depois deste se encontrar numa posição confortável para executar a tarefa, o supervisor da experiência entregou-lhe o smartphone, com a aplicação já iniciada. O supervisor informou-o que teria que conduzir o robô apenas através da aplicação, onde teria a visão do robô e o controlo do mesmo. Após todas estas explicações de como iria efetuar a tarefa, foi lhe dito que teria que efetuar uma volta de teste antes de cada modo de condução que iria utilizar, para ter uma noção de como o robô se iria comportar consoante as suas indicações. Portanto teria que realizar uma volta de teste antes de iniciar a experiência e outra quando mudasse de modo de condução. Assim que a primeira volta de teste terminou, o individuo foi informado que iria iniciar a experiência. O robô foi colocado na zona de partida, como se pode ver na figura 3.7.

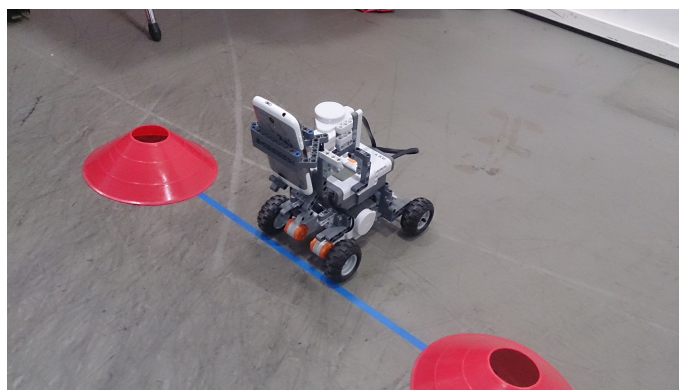


Figura 3.7: Zona de inicio da experiência



### 3.4 Resultados

Neste ponto serão apresentados os resultados da experiência. São estes que vão indicar se, de facto, existe alguma vantagem em utilizar o modo de condução biológico em vez do manual.

Depois dos dados todos recolhidos, estes foram organizados e sujeitos a testes estatísticos, utilizando a ferramenta SPSS.

Os dados analisados foram interpretados conforme se descreve nas próximas secções:

#### 3.4.1 Tempo

##### 3.4.1.1 Efeito de sessão

Duas maneiras de análise de medidas repetidas da variância (ANOVA):

$$F(3, 17) = 3.25; p < 0.05$$

Conforme se observa, verifica-se efeito de aprendizagem. A análise de pares mostra uma diferença significativa entre a sessão 1 e sessão 4 ( $p < 0,03$ ). O tempo para completar o percurso diminuiu significativamente da sessão 1 para a sessão 4.

$$F(3, 17) = 1.68; \text{Não significativo}$$

Não há efeito da interação entre as condições (Manual vs Biológico) e as sessões (1, 2, 3 e 4).

##### 3.4.1.2 Efeito da condição (Manual vs Biológico)

Uma maneira de análise de medidas repetidas da variância (ANOVA):

$$\text{Global: } F(1, 19) = 15.16; p < 0.01$$

Amostras emparelhadas T-teste:

$$\text{Sessão 1: } F(19) = 3.19; p < 0.01$$

$$\text{Sessão 2: } F(19) = 2.11; p < 0.05$$

$$\text{Sessão 3: } F(19) = 2.33; p < 0.04$$

$$\text{Sessão 4: } F(19) = 1.40; \text{Não significativo}$$

No que diz respeito ao tempo despendido pelos utilizadores para concluir cada percurso, chega-se à conclusão que de uma maneira geral o modo de condução biológico tem uma vantagem significativa em relação ao modo de condução manual.

Em particular conclui-se que apenas a sessão 4 não tem uma diferença significativa, contudo através da figura 3.9 se possa observar que embora a diferença não seja significativa não deixa de ter vantagem. Verificou-se ainda um efeito de aprendizagem.

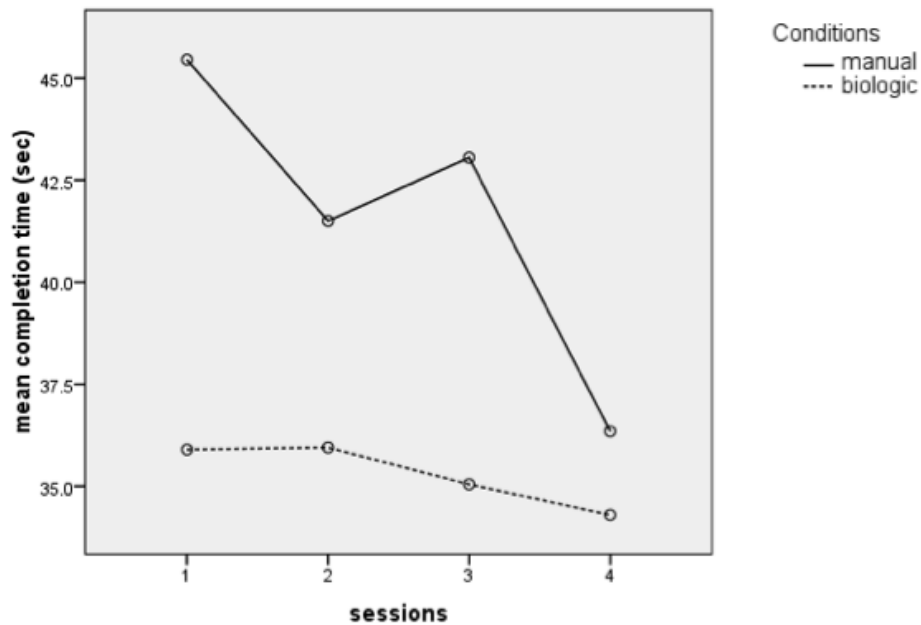


Figura 3.9: Tempo

### 3.4.2 Colisões

#### 3.4.2.1 Efeito de sessão

Estabeleceu-se duas formas de análise de medidas repetidas da variância (ANOVA):

$$F(3, 17) = 4.09; p < 0.03$$

Verifica-se também um efeito de aprendizagem. A análise de pares mostra uma diferença significativa entre a sessão 1 e sessão 4 ( $p < 0,02$ ). As colisões diminuem significativamente da sessão 1 para a sessão 4.

$$F(3, 17) = 2.08; \text{ Não significativo}$$

Não há efeito da interação entre as condições (Manual vs Biológico) e as sessões (1, 2, 3 e 4).

### 3.4.2.2 Efeito da condição (Manual vs Biológico)

Foi utilizada uma forma de análise de medidas repetidas da variância (ANOVA):

$$\text{Global: } F(1, 19) = 7.75; p < 0.02$$

Amostras emparelhadas T-teste:

$$\text{Sessão 1: } F(19) = 3.59; p < 0.01$$

$$\text{Sessão 2: } F(19) = -0.39; \text{ Não significativo}$$

$$\text{Sessão 3: } F(19) = 1.07; \text{ Não significativo}$$

$$\text{Sessão 4: } F(19) = 2.50; p < 0.03$$

Quanto às colisões, os resultados obtidos mostram que também de uma maneira geral o modo de condução biológico é mais vantajosa que o modo de condução manual.

No caso particular, pode-se verificar que na sessão 2 e 3 a diferença entre eles não é significativa. Contudo, através da figura 3.10 pode-se observar duas situações distintas entre estas duas sessões não significativas. A sessão 3, embora a diferença não seja significativa, não deixa de ter vantagem. Na sessão 2 já não se verifica tal situação, a diferença não só não é significativa como também demonstra que não existe qualquer vantagem em usar o modo biológico. Verificou-se também neste aspeto um efeito de aprendizagem.

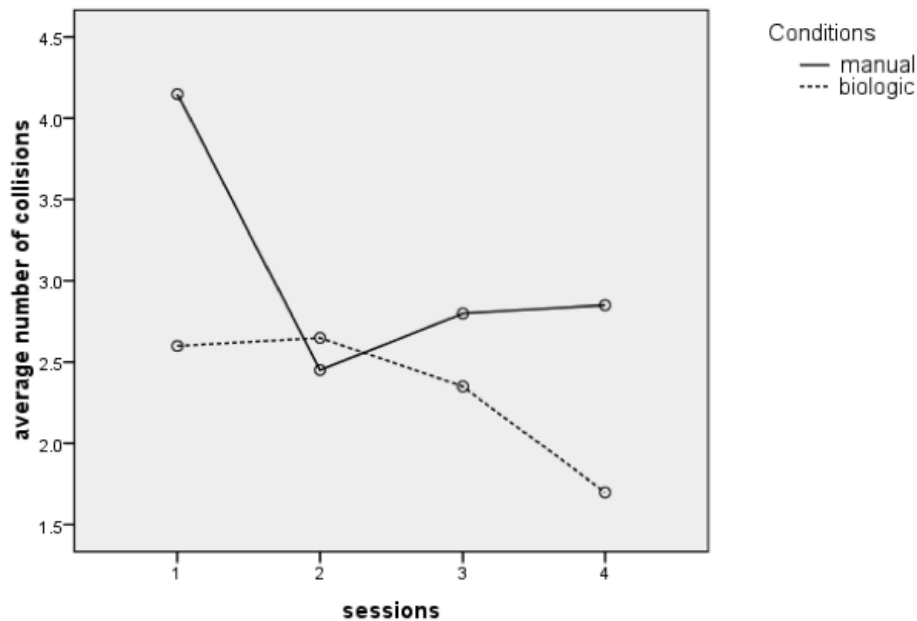


Figura 3.10: Colisões

### 3.4.3 Paragens

#### 3.4.3.1 Efeito de sessão

Foram utilizadas duas formas de análise de medidas repetidas da variância (ANOVA):

$$F(3, 17) = 12.53; p < 0.01$$

Verifica-se efeito de aprendizagem. A análise de pares mostra uma diferença significativa entre a sessão 1 e sessão 4 ( $p < 0,01$ ). As paragens diminuíram significativamente da sessão 1 para a sessão 4.

$$F(3, 17) = 0.18; \text{Não significativo}$$

Não há efeito da interação entre as condições (Manual vs Biológico) e as sessões (1, 2, 3 e 4).

#### 3.4.3.2 Efeito da condição (Manual vs Biológico)

Uma maneira de análise de medidas repetidas da variância (ANOVA):

$$\text{Global: } F(1, 19) = 1.72; \text{Não significativo}$$

Nas paragens que o utilizador recorreu para efetuar cada percurso, os resultados obtidos dizem que de uma forma geral a vantagem do modo de condução biológico em relação ao modo de condução manual não é significativa.

No entanto com recurso à figura 3.11, pode-se confirmar que existe uma ligeira vantagem em usar o modo de condução biológico. Mas dado o facto de essa vantagem ser tão ligeira, não se pode considerar que o modo de condução biológico é mais vantajoso em termos de paragens que o modo manual. Em termos de paragens verificou-se também um efeito de aprendizagem.

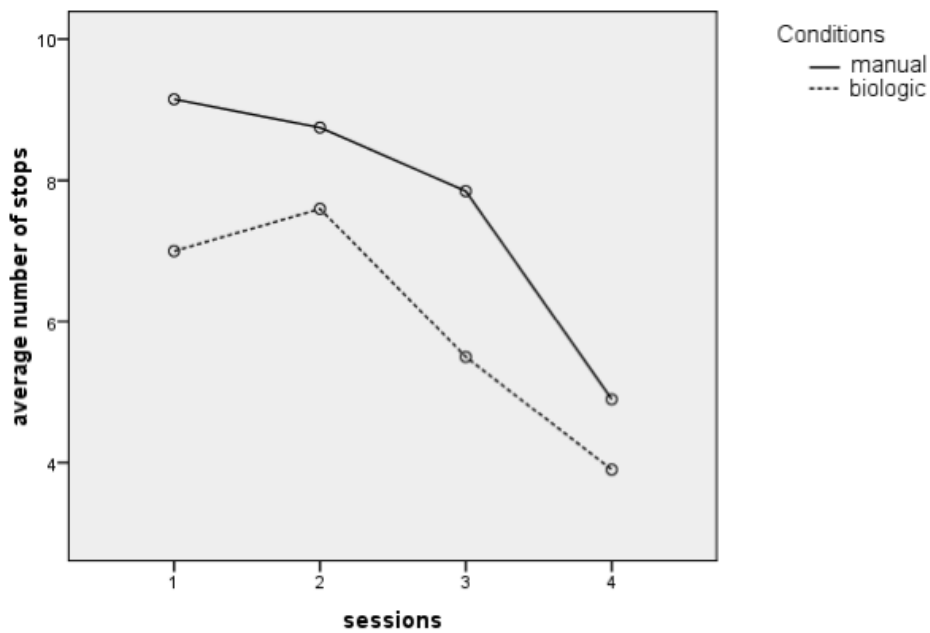


Figura 3.11: Paragens

### 3.4.4 Suavidade de trajetória - pequeno raio de curvatura

A suavidade de trajetórias foi analisada através de um cálculo com base na distribuição de frequência do raio de curvatura instantânea de cada trajetória. Para tal foi utilizada a seguinte fórmula:

$$r(m) = \frac{v(m/s)}{w(rad/s)}$$

onde  $r(m)$  corresponde ao raio de curvatura,  $v(m/s)$  é a velocidade instantânea, e  $w(rad/s)$  é a velocidade de rotação instantânea absoluta. O raio de curvatura é convertido em logaritmo decimal[18].

#### 3.4.4.1 Efeito de sessão

Foram utilizadas duas formas de análise de medidas repetidas da variância (ANOVA):

$$F(3, 17) = 0.79; \text{ Não significativo}$$

Não se verifica efeito de aprendizagem.

$$F(3, 17) = 0.57; \text{ Não significativo}$$

Não há efeito da interação entre as condições (Manual vs Biológico) e as sessões (1, 2, 3 e 4).

### 3.4.4.2 Efeito da condição (Manual vs Biológico)

Foi utilizada uma forma de análise de medidas repetidas da variância (ANOVA):

$$\text{Global: } F(1, 19) = 15.35; p < 0.01$$

Amostras emparelhadas T-teste:

$$\text{Sessão 1: } F(19) = 2.44; p < 0.03$$

$$\text{Sessão 2: } F(19) = 3.38; p < 0.01$$

$$\text{Sessão 3: } F(19) = 4.39; p < 0.01$$

$$\text{Sessão 4: } F(19) = 3.66; p < 0.01$$

Nos pequenos raios de curvatura, de uma maneira geral o modo de condução biológico tem uma vantagem significativa em relação ao modo de condução manual.

Em particular pode-se concluir também que em todas as sessões existe uma clara vantagem em utilizar o modo de condução biológico. Esta clara vantagem pode-se observar na figura 3.12. Neste aspeto não se pode dizer que exista um efeito de aprendizagem.

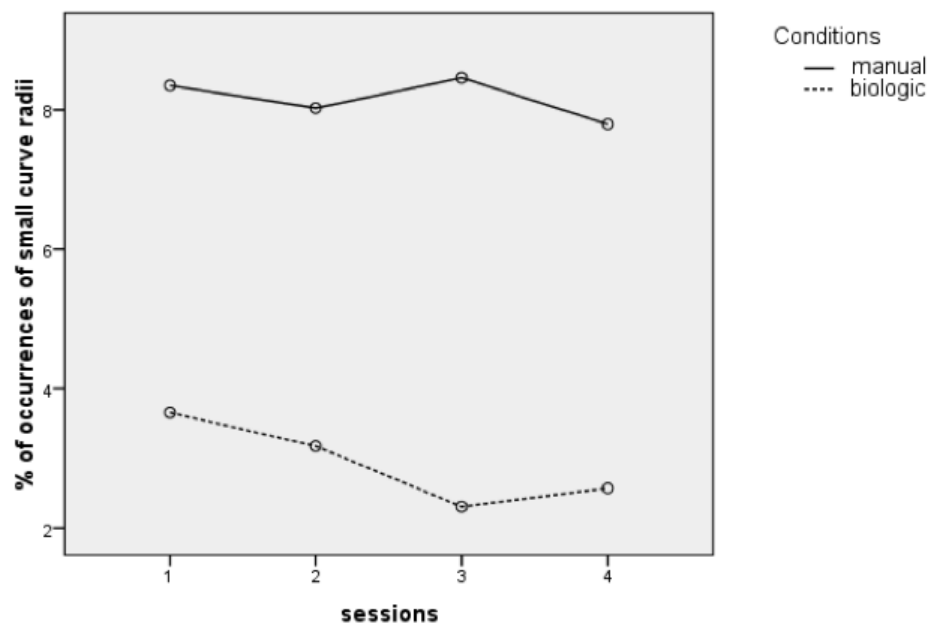


Figura 3.12: Suavidade de trajetória - pequeno raio de curvatura

### 3.4.5 Suavidade de trajetória - grande raio de curvatura

#### 3.4.5.1 Efeito de sessão

Foram utilizadas duas de análise de medidas repetidas da variância (ANOVA):

$$F(3, 17) = 0.13; \text{ Não significativo}$$

Não se verifica efeito de aprendizagem.

$$F(3, 17) = 1.84; \text{ Não significativo}$$

Não há efeito da interação entre as condições (Manual vs Biológico) e as sessões (1, 2, 3 e 4).

#### 3.4.5.2 Efeito da condição (Manual vs Biológico)

Foi utilizada uma forma de análise de medidas repetidas da variância (ANOVA):

$$\text{Global: } F(1, 19) = 16.02; p < 0.01$$

Amostras emparelhadas T-teste:

$$\text{Sessão 1: } F(19) = 2.59; p < 0.02$$

$$\text{Sessão 2: } F(19) = 3.38; p < 0.01$$

$$\text{Sessão 3: } F(19) = 4.37; p < 0.01$$

$$\text{Sessão 4: } F(19) = 3.52; p < 0.01$$

Nos grandes raios de curvatura as conclusões não são diferentes, pois pode-se também concluir que de uma maneira geral o modo de condução biológico tem uma vantagem significativa em relação ao modo de condução manual.

Nestes também se verifica que em todas as sessões existe uma clara vantagem em utilizar o modo de condução biológico. Neste caso esta vantagem é demonstrada na figura 3.13.

Neste aspeto, tal como no anterior, não se pode dizer que exista um efeito de aprendizagem.

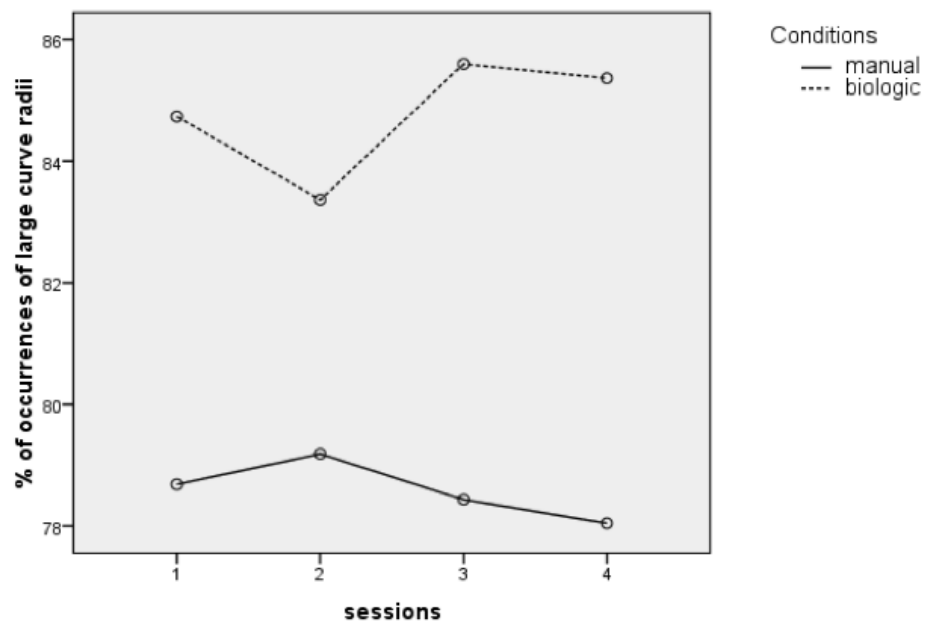


Figura 3.13: Suavidade de trajetória - grande raio de curvatura

### 3.5 Discussão

Este estudo procurou saber se existiria alguma vantagem em usar um modo de condução biológico. Foram comparados dois modos de condução: o modo de condução manual tradicional e um modo de condução biológico. Este modo consiste na aplicação de um algoritmo de controlo autónomo da velocidade de deslocação do robô em função da sua trajetória, segundo a "Lei de Potência 2/3"[23].

Nos resultados apresentados é possível concluir que de um modo geral o modo de condução biológico apresenta, para o utilizador, vantagens significativas em quase todos os aspetos estudados.

O menor tempo despendido pelos utilizadores na realização de cada percurso, o número inferior de colisões e a maior suavidade das trajetórias, apresentam-se como fatores determinantes para a escolha do modo de condução biológico em detrimento do modo de condução manual. Já o número de paragens ocorridas durante o percurso, contabilizado durante os dois modos de condução, não mostrou que haja uma vantagem significativa para o utilizador no modo de condução biológico.

Face a estes resultados, pode-se afirmar que o modo de condução biológico do robô parece permitir ao utilizador conduzir o veículo de forma mais rápido e com maior segurança. Tudo aponta para a confirmação da hipótese anteriormente colocada. Por outro lado, a maior suavidade de trajetórias observadas no modo de condução biológico confere maior naturalidade ao processo de condução.

O objetivo do estudo passava por facilitar, ao máximo, a condução do robô. No modo biológico, a preocupação do utilizador estava centrada apenas na manutenção da trajetória correta que pretendia dar ao robô, já que a velocidade era controlada de forma automática. A introdução de uma visão móvel permitia ao utilizador antecipar melhor a trajetória a seguir nas curvas.

No modo de condução manual o utilizador não só teria que controlar a direção, tal como no modo biológico, como também a velocidade. O controlo simultâneo da trajetória a seguir e da velocidade obrigavam a que o utilizador necessitasse de uma maior concentração e habilidade. Os resultados deste modo de condução, revelam ser mais difícil conduzir neste modo.

Para que a diferença nos resultados se atenuasse, os utilizadores necessitavam de estar treinados e habituados aos percursos, para poderem ter uma melhor noção da velocidade que teriam que imprimir ao robô consoante a sua trajetória. É aqui que a "Lei de Potência 2/3", utilizada para controlar a velocidade do robô segundo a sua trajetória, faz a diferença. Ela permite atribuir, aproximadamente, a melhor velocidade a que o robô deve ser conduzido, tendo em conta a sua trajetória. Provavelmente, o facto do percurso possuir curvas tão acentuadas também levou a que esta vantagem fosse tão clara. Assim, ainda que o utilizador revelasse maior habilidade a controlar o robô no modo manual, dificilmente iria conduzir o robô melhor que no modo biológico, dado que teria de ter uma noção precisa para a velocidade a adotar para o robô em cada momento do trajeto.

O modo de condução biológico poderia ser um bom sistema a ser implementada nos automóveis, ainda que a escala da dimensão do problema fosse maior, o que exigiria um aperfeiçoamento do processo, pois iria elevar os níveis de segurança e facilitar a condução do veículo. Desta forma, poderia ser reduzida a sinistralidade tendo em conta que um dos seus fatores de risco assenta na velocidade inadequada decidida pelo condutor na abordagem das trajetórias que pretende que o veículo siga.



## CONCLUSÃO E TRABALHO FUTURO

Nesta tese, pretendeu-se implementar um sistema de assistência ao controlo remoto (ou tele-operação), sem contacto visual direto de um robô NXT, com recurso a um dispositivo móvel baseado em Android.

A primeira fase, consistiu em desenvolver o melhor robô para efetuar a tarefa pretendida. Foram consumidas muitas horas em torno da montagem e desmontagem de peças LEGO na procura do robô mais eficiente. Após várias tentativas, chegou-se aos dois robôs das figuras 2.3 e 2.5. O primeiro foi um robô de fácil construção, enquanto que a construção segundo se revestiu de uma maior complexidade. Depois de testar os dois robôs, concluiu-se que o primeiro seria o que melhor se adequava ao que era pretendido, preenchendo todos os requisitos necessários para efetuar a tarefa da forma mais eficaz.

A segunda fase, que tinha como objetivo desenvolver uma aplicação em Android para controlar o robô nos dois diferentes modos de condução, foi a mais demorada. Embora já se tivesse algumas noções de programação em Android, em parte devido ao conhecimento da linguagem utilizada, Java, este era um desafio praticamente novo que levou a adquirir muitos conhecimentos.

Foi efetuada uma pesquisa para entender melhor a programação do Android, nomeadamente, focando-se inicialmente nos aspetos gráficos, na melhor maneira de construir uma interface que fosse o mais intuitiva possível para o utilizador.

A maior dificuldade nesta fase, foi estabelecer as conexões entre o controlador, o robô e a IP câmara, pois foi necessário pesquisar como o fazer e de que maneira.

Ainda neste etapa surgiu um novo problema. Como calcular a velocidade do robô e a partir desta, deduzir a fórmula que levaria o robô a comportar-se segundo a Lei de potência  $2/3$ .

Ultrapassadas estas adversidades, o objetivo nesta etapa foi alcançado com sucesso, pois o robô comporta-se tal como foi idealizado e a aplicação desenvolvida demonstrou ser muito intuitiva e simples para o utilizador.

A terceira e última fase, onde foi montado um cenário experimental para testar a hipótese, foi bastante mais simples do que se esperava. Havia o receito de não se conseguir encontrar 20 indivíduos que se disponibilizassem como utilizadores do sistema, mas acabou por se tornar fácil. Foi decisiva a colaboração de todos os colegas que prontamente se mostraram disponíveis para realizar a experiência. Fica aqui o agradecimento a todos os participantes, pois sem eles a realização da experiência poderia ficar comprometida.

Para realizar a experiência foi necessário desenhar um percurso. Foram consideradas várias hipóteses. No entanto o percurso mais adequado seria o da figura 3.4, pois este era curto, relativamente simples e com curvas acentuadas. O percurso escolhido era curto e simples de modo a que a experiência não fosse muito demorada. Este teria que possuir curvas acentuadas, dado que se admitia que poderia ser nestas que o modo de condução biológico iria obter uma maior vantagem em relação ao modo de condução manual. O que mais tarde, depois de os resultados experimentais serem analisados, se veio a verificar que de facto era nas curvas que o modo de condução biológico era mais vantajoso.

No que diz respeito ao objetivo principal do estudo desenvolvido, foi possível concluir que a condução de um robô por meio de tele-operação se torna mais fácil utilizando um algoritmo de controlo autónomo da velocidade de deslocação do robô em função da sua trajetória, segundo a "Lei de Potência 2/3". O modo de condução que recorre ao algoritmo permite ao utilizador conduzir o veículo de modo mais rápido e com maior segurança, em comparação com o modo de condução manual tradicional. A análise dos resultados também revelou uma maior suavidade das trajetórias observadas na condição biológica, revelando-se uma condução mais natural.

Entre os trabalhos a realizar no futuro, a implementação deste sistema, como um todo, possa ser alargada a outro tipo de robôs, designadamente, aqueles que podem ser usados na proteção civil ou para fins militares, conferindo-lhes maior eficiência na sua operação.

Seria, contudo interessante para a indústria automóvel desenvolver estudos que conduzissem à implementação deste modo de condução em veículos automóveis. Na verdade, o modo de condução biológico poderia facilitar a condução do veículo conferindo maior suavidade às trajetórias, proporcionando maior conforto aos seus ocupantes e acima de tudo, fornecendo maior segurança à condução. Desta forma, poderia ser reduzido um dos fatores de risco na sinistralidade, aquele que se prende com a escolha, por parte do condutor de uma velocidade inadequada á trajetória a seguir pelo veículo.

O trabalho termina com um sentimento de dever cumprido, pois as três fases planeadas foram concluídas com sucesso.

## BIBLIOGRAFIA

- [1] *Android Architecture*. URL: [http://elinux.org/Android\\_Architecture#Architecture\\_Diagram](http://elinux.org/Android_Architecture#Architecture_Diagram) (acedido em 23/03/2015).
- [2] *Android SDK*. URL: <http://developer.android.com/sdk/index.html> (acedido em 23/03/2015).
- [3] *Android, the world's most popular mobile platform*. URL: <http://developer.android.com/about/index.html> (acedido em 23/03/2015).
- [4] *Arduino Build Process*. URL: <http://arduino.cc/en/Hacking/BuildProcess#> (acedido em 23/03/2015).
- [5] *Comparação OS*. URL: <http://www.gadgetguide4u.com/wp-content/uploads/2013/06/iOS-7-vs-Android-4.2-JellyBean-vs-Windows-Phone-8-vs-BlackBerry-10.jpg> (acedido em 23/03/2015).
- [6] *DFRobotShop Rover*. URL: [http://www.myrobots.com/wiki/DFRobotShop\\_Rover#About](http://www.myrobots.com/wiki/DFRobotShop_Rover#About) (acedido em 23/03/2015).
- [7] *Finch Programmable Mobile Robot Platform*. URL: <http://www.robotshop.com/en/finch-programmable-mobile-robot-platform.html> (acedido em 23/03/2015).
- [8] P. M. Fitts. "The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement". Em: *General* 121.3 (1992), pp. 262–269.
- [9] R. Grasso, S. Glasauer, Y. Takei e A. Berthoz. "The predictive brain : anticipatory control of head direction for the steering of locomotion". Em: *NeuroReport* 7 (1996), pp. 1170–1174.
- [10] T. L. Group. *Mindstorms education*. 2006.
- [11] IDC. *Smartphone OS Market Share*. URL: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp#> (acedido em 23/03/2015).
- [12] *Installing the Eclipse Plugin*. URL: <https://developer.android.com/sdk/installing/installing-adt.html> (acedido em 23/03/2015).
- [13] *Introdução à Plataforma Eclipse*. URL: <https://www.ibm.com/developerworks/br/library/os-eclipse-platform/> (acedido em 23/03/2015).
- [14] M. F. Land e D. Lee. "Where we look when we steer ?" Em: *Nature* 369 (1994), pp. 339–340.

- [15] *Mindstorms*. URL: [https://web.archive.org/web/20090930194014/http://mindstorms.lego.com/Overview/The\\_NXT.aspx](https://web.archive.org/web/20090930194014/http://mindstorms.lego.com/Overview/The_NXT.aspx) (acedido em 23/03/2015).
- [16] *MyDFRobotShop Rover - Arduino Tracked Robot*. URL: <http://www.robotshop.com/en/mydfrobotshop-rover-myrobots-arduino.html#pageTop> (acedido em 23/03/2015).
- [17] *NXT Mindstorms*. URL: <http://roboticsaumair.jimdo.com/nxt-mindstorms/> (acedido em 23/03/2015).
- [18] P. Péruch e D. Mestre. "Between Desktop and Head Immersion: Functional Visual Field During Vehicle Control and Navigation in Virtual Environments". Em: *Presence* 8.1 (1999), pp. 54–64.
- [19] R. Plamondon e W. Guerfali. "The 2/3 power law: when and why?" Em: *Acta Psychol (Amst)* 100.1-2 (1998), pp. 85–96.
- [20] Y. Rybarczyk, G. S., H. P., C. E. e M. D. "The development of robot human-like behaviour for an efficient human-machine co-operation". Em: *AAATE* (2001).
- [21] Y. Rybarczyk, M. D., H. P. e C. E. "Implémentation de mécanismes d'anticipation visuo-motrice en téléopération." Em: *Le Travail Humain* (2004), pp. 209–233.
- [22] Y. P. Rybarczyk e D. Mestre. "Effect of temporal organization of the visuo-locomotor coupling on the predictive steering". Em: *frontiers in PSYCHOLOGY* (2012).
- [23] P. Viviani e R. Schneider. "A developmental study of the relationship between geometry and kinematics in drawing movements". Em: *J. Exp. Psychol. Hum. Percept. Perform.* 17 (1991), pp. 198–218.

APÉNDICE



ANEXO

The table displays a 120x120 matrix of numerical data. The diagonal elements, which represent the velocity at each point relative to itself, are highlighted in red and are consistently 0. The off-diagonal elements represent the velocity between different points, with values ranging from approximately 1 to 119. The matrix is symmetric, indicating that the velocity between point i and point j is the same as between point j and point i. The values increase as the distance between points increases, following a linear pattern along the diagonal.

Figura A.1: Tabela das velocidades



