Instituto Politécnico de Tomar

Escola Superior de Tecnologia de Tomar

Jorge Anécio Dias Rodrigues

Robotização de uma cadeira de rodas elétrica

Projeto

Orientado por:

Doutor Gabriel Pereira Pires – Instituto Politécnico de Tomar

Doutora Ana Cristina Lopes – Instituto Politécnico de Tomar

Projeto apresentado ao Instituto Politécnico de Tomar
para cumprimento dos requisitos necessários à obtenção do
grau de Mestre em Engenharia Eletrotécnica

DEDICATÓRIA

Dedico este projeto a todos os que me apoiaram neste percurso, particularmente à minha família, que se privou de muitos momentos em conjunto para que este objetivo fosse alcançado.

Dedico-o também a todos que têm um sonho a concretizar e não se deixam desvanecer com as "intempéries" do percurso.

RESUMO

Este projeto consiste na robotização de uma cadeira de rodas elétrica comercial Salsa R2 existente no laboratório VITA.IPT, dotando-a de um sistema de navegação assistida que lhe confere um nível de segurança anti-colisão. A robotização da cadeira de rodas foi realizada mantendo funcional o sistema original da cadeira de rodas Salsa R2. Foi concebida uma nova arquitetura de *hardware* e *software*, tendo sido integrados os seguintes módulos: 1) sistema de potência baseado no controlador Roboteq MDC2230; 2) *joystick* para interface do utilizador; 3) unidade de processamento/controlo de baixo nível baseado numa *Raspberry Pi* (RPi); 4) módulo sensorial de perceção baseado num laser Hokuyo UTM-30LX; 5) módulo de odometria; e 6) unidade de processamento de alto nível com plataforma ROS (*Robotic Operating System*) para implementação de algoritmos de navegação.

O sistema permite ao utilizador selecionar entre modo manual e modo assistido, através de um comutador acessível no braço da cadeira. No modo de navegação assistida, o módulo de navegação/perceção evita a colisão com possíveis obstáculos que se encontrem no percurso de circulação, filtrando comandos inapropriados do utilizador. O método implementado baseia-se na análise de ocupação de sectores predefinidos numa região frontal à cadeira de rodas, e tem em consideração o comando de direção e velocidade do utilizador e as definições físicas e dinâmicas da cadeira robótica.

O sistema foi testado e validado experimentalmente com sucesso em ambiente real (ambiente interior de laboratório e corredores, com obstáculos estáticos e dinâmicos).

Palavras-chave: Cadeira rodas robótica, ROS, Navegação assistida, Roboteq, *Raspberry Pi*

ABSTRACT

This project describes the modification of a commercial powered wheelchair Salsa R2, available in the laboratory VITA.IPT, into a robotic wheelchair, endowing it with an assisted navigation system which provides an anti-collision safety level. The modification performed on the original powered wheelchair Salsa R2 kept the original system functional. New hardware and *software* architectures were designed and the following modules were implemented: 1) Power system based on the MDC2230 Roboteq controller; 2) *joystick* as user interface; 3) low-level processing/control unit based on a Raspberry Pi (RPi); 4) sensory perception module based on a Hokuyo UTM-30LX laser; 5) odometry module; e 6) high level processing unit based on the ROS platform for the implementation of the navigation algorithm.

The system offers the user the possibility of switching between the navigation mode and the manual mode, through switch-button on the wheelchair arm. In the assisted navigation mode, the navigation/perception module avoids the collision with possible obstacles located on the navigation track, by filtering unsuitable commands from the user. The implemented method is based on the analysis of the occupation of a predefined grid in the frontal area of the wheelchair, and considers the velocity and direction commands given by the user and the physical and dynamic definitions of the wheelchair.

The system was tested and validated experimentally in real environment with success (indoor environments, namely VITA.IPT laboratory and halls, considering static and dynamic obstacles).

Keywords: Powered wheelchair, ROS, Assisted navigation, Roboteq, Raspberry Pi

AGRADECIMENTOS

A realização deste projeto de mestrado contou com importantes apoios e incentivos sem os quais não teria sido possível e aos quais estarei eternamente grato.

Aos meus orientadores, a professora Doutora Ana Lopes e o professor Doutor Gabriel Pires, pela liberdade e confiança que depositaram em mim, cedendo-me toda a autonomia necessário para a gestão do tempo e disponibilidade para evoluir neste trabalho.

Aos colegas e professores de curso pelo conhecimento e motivação que me transmitiram ao longo de todo o percurso académico. Ao Eng.º Pedro Neves pela disponibilidade e apoio às questões e dúvidas apresentadas no decorrer deste projeto.

Um agradecimento em especial para a minha família que sempre esteve a meu lado com palavras de motivação e encorajamento desde o início à conclusão desta jornada.

Não podia deixar de agradecer também os recursos e condições oferecidas pelo Lab. VITA.IPT – Vida Assistida por Ambientes Inteligentes.

Este projeto foi parcialmente financiado pelo Projeto QREN: ICT_2013_05_010_5334 ID:64763: Reforço das competências científicas e tecnológicas do Instituto Politécnico de Tomar (fonte de financiamento 415).

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

LISTA DE ACRÓNIMOS

BCI – *Brain Control Interface*

CR - Cadeira robótica

DWA - *Dynamic Window Approach*

GPIO - *General purpose input/outpu*t

HDMI - High-Definition Multimedia Interface

IP - Internet Protocol

LIDAR – *Light Detection And Ranging*

LRF - *Laser Rangefinder*

NAAC – Navegação Assistida Anti-Colisão

PC – *Personal Computer*

PID - Proporcional Integral Derivativo

RAM - *Random Access Memory*

ROS - *Robotic Operating System*

RPi - *Raspberry Pi*

SD - *Secure Digital Card*

SLAM - *Simultaneous Localization and Mapping*

SPI - *Serial Peripheral Interface*

SSH - *Secure Shell*

TTL - *Transistor–Transistor Logic*

UDP - *User Datagram Protocol*

USB - *Universal Serial Port*

VFF - *Virtual Force Field*

1. Introdução

1.1. Motivação e Contexto

A falta de mobilidade numa pessoa pode causar grandes dificuldades na realização das mais simples ações do nosso quotidiano. Estas limitações motoras tornam a pessoa muitas vezes dependente da intervenção de terceiros para ajudar na sua deslocação.

No caso de limitações motoras graves nos membros inferiores, as cadeiras de rodas são uma das formas mais utilizadas de auxílio na deslocação. Os modelos motorizados podem ser usados sem muito esforço, sendo particularmente relevantes para pessoas com limitações também nos membros superiores. Ainda assim, os indivíduos com limitações mais severas nos membros superiores podem não conseguir controlar um *joystick* com a destreza desejada, levando a ineficiência e insegurança na condução. A robotização das cadeiras de rodas pode ajudar a aumentar a segurança da sua utilização ou mesmo possibilitar uma navegação semi-autónoma ou autónoma através de métodos de perceção do ambiente envolvente e do planeamento automático de trajetórias. Existem utilizadores para os quais a robotização das cadeiras de rodas poderá contribuir para uma melhoria da sua independência e consequente aumento da sua qualidade de vida e melhor inclusão na sociedade.

1.2. Objetivos

Pretende-se com este projeto robotizar uma cadeira de rodas elétrica modelo Salsa R2 existente no laboratório VITA.IPT, preservando o seu funcionamento original, nomeadamente o seu sistema de tração e as interfaces de condução (*joystick* e comando de cabeça). A cadeira de rodas ficará deste modo com dois sistemas a funcionar em paralelo, o original e o implementado neste trabalho de projeto, sendo possível comutar entre eles de forma relativamente simples. A robotização da cadeira de rodas consistiu na implementação de um sistema de segurança integrando um módulo de localização baseado em odometria e um módulo de perceção baseado num sensor *Laser Rangefinder* (LRF). Estes módulos de

localização e perceção permitem a deteção e mapeamento dos obstáculos envolventes, sendo assim possível a implementação de funções de assistência à navegação da cadeira.

Desta forma os principais objetivos do projeto são:

- Análise dos componentes da cadeira de rodas Salsa R2 a fim de verificar possível integração dos mesmos no projeto de robotização da cadeira
- Desenvolvimento de uma base robótica constituída por interface de condução, controlador de tração e estimação de posição (odometria);
- Desenvolvimento de um sistema de perceção para implementação de um método de navegação reativa.

1.3. Contribuições e trabalho realizado

A principal contribuição desta dissertação é a adaptação de uma cadeira de rodas, dotando-a de um sistema de perceção que possibilita a sua navegação com um sistema reativo de anti-colisão.

A adaptação e robotização da cadeira de rodas Salsa R2 passou por várias fases de desenvolvimento. Após análise do sistema originalmente implementado na cadeira de rodas Salsa R2, verificou-se que o esforço necessário para o reaproveitamento de alguns módulos através de engenharia inversa não seria vantajoso devido à dificuldade em adquirir e processar os dados que circulavam entre os seus periféricos. Esta dificuldade deveu-se ao sistema ser fechado e não serem conhecidos os protocolos de comunicação. Desta forma foi necessário desenvolver e implementar uma solução de raiz para alcançar os objetivos traçados. Fizeram parte desta solução:

- A Implementação de um controlador num microcomputador RPi para a integração da interface de navegação e respetiva comunicação com outros periféricos;

- A integração de uma interface para navegação da cadeira baseada num *joystick*;

- A integração de *encoders* para implementação de controlo de velocidade e odometria do sistema;

- A integração de um sensor LRF para perceção local do ambiente;

- A integração de módulos na plataforma ROS (Robotic Operating System) para processamento da informação sensorial e de controlo recorrendo a algumas bibliotecas existentes;

- Implementação de um módulo de navegação reativa anti-colisão.

1.4. Organização da dissertação

O documento encontra-se dividido nos seguintes capítulos:

- **Introdução**

Este capítulo apresenta uma descrição sucinta do projeto desenvolvido, do seu enquadramento, objetivos e contribuições.

- **Estado da Arte e Fundamentos**

Apresentação do estado da arte respeitante às temáticas abordadas nesta tese, nomeadamente cadeiras de rodas robotizadas e sistemas baseados na plataforma ROS, assim como alguns fundamentos de sistemas de navegação.

- **Arquitetura de *Hardware* e *Software***

Neste capítulo são descritas as principais características e funcionalidades dos diversos componentes integrados no desenvolvimento da cadeira robotizada.

- **Localização e Navegação**

Descrição e desenvolvimento dos métodos aplicados na perceção e análise do meio ambiente circundante que serão utilizados no algoritmo de navegação reativa proposto.

- **Integração do sistema na plataforma ROS**

Descrição dos vários módulos integrados na plataforma ROS responsáveis pela perceção e navegação do sistema, sua interligação e representação de nós e tópicos associados.

- **Ensaios e resultados experimentais**

Descrição e apresentação dos ensaios experimentais realizados, permitindo assim documentar e validar os conceitos apresentados no desenvolvimento desta tese.

- **Conclusões e trabalho futuro**

Apresentação das conclusões obtidas no desenvolvimento da tese, assim como propostas de trabalhos a desenvolver para melhorar o desenvolvimento realizado.

2. Estado da Arte e Fundamentos

2.1. Cadeiras de Rodas Robotizadas

A evolução tecnológica a que temos vindo a assistir tem proporcionado a generalização de sensores e atuadores, tornando economicamente mais viável a sua utilização em aplicações de carácter de assistência na mobilidade a pessoas com limitações motoras ou cognitivas, podendo assim acrescentar grande valor na qualidade de vida dessas pessoas. No caso especifico das cadeiras de rodas, existe já uma grande oferta de cadeiras de rodas motorizadas, dotadas dos mais versáteis interfaces de navegação, específicos ou aconselhados para cada tipo de limitação do utilizador, conseguindo assim estender este tipo de ajuda a outro amplo leque de possíveis utentes.

A evolução da robótica tem permitido o desenvolvimento de vários projetos e aplicações para a assistência do ser humano, nomeadamente o desenvolvimento de sistemas de ajuda na navegação ou mesmo a navegação autónoma de cadeiras de rodas. Apresentamos alguns desses protótipos na secção 2.1.1. Estes projetos, apesar de muito evoluídos e testados experimentalmente carecem ainda de alguma robustez e encontram-se ainda em fases de prototipagem e experimentação laboratorial, não estando assim ainda disponíveis no mercado para comercialização particular. Em 2016 a *Singapore-MIT Alliance for Research and Technology* (SMART) implementou com sucesso uma cadeira de rodas inteligente no "*Singapore's Changi General Hospital*" [1], a qual consegue percorrer com sucesso os corredores do hospital. Esta cadeira realiza mapeamento através de três sensores lidar e possui um algoritmo para estimar a sua localização. Possui também seis rodas que lhe fornecem estabilidade e manobrabilidade para realizar navegação através de portas de normal dimensão. Recentemente implementou mais uma cadeira de rodas inteligente no Aeroporto Haneda, em Tóquio [2], tendo incorporado no seu sistema funções de paragem automática no caso de proximidade com obstáculos, navegação autónoma através da seleção de destinos e a possibilidade de deslocação em linha, evitando a esforço do pessoal do aeroporto para a recolha das cadeiras de rodas.

Tem-se verificado também alguns desenvolvimentos de cadeiras de rodas inteligentes baseados em módulos independentes para integração em cadeiras de rodas elétricas, provendo-as assim de funcionalidades suplementares para navegação assistida, como o caso do *Smart Wheelchair Component System* [3], tendo este protótipo sido já avaliado em quatro cadeiras de rodas de diferentes fabricantes e o *Hephatestus Smart Wheelchair* [4] foi desenvolvido a pensar na possibilidade de integração de vários componente por parte de fabricantes e clínicos para a conversão de cadeiras de rodas elétricas em cadeiras de rodas inteligentes, estando o protótipo montado numa cadeira *Everest and Jennings Lancer* 2000.

2.1.1. Projetos de cadeiras de rodas inteligentes

A robotização de cadeiras de rodas tem sido um tema proeminente nos últimos anos, onde se têm verificado vários estudos e desenvolvimento de protótipos, baseados nos mais diversos interfaces, sensores e técnicas de navegação.

Alguns desses projetos e suas principais características encontram-se na Tabela 1, sendo as suas características adequadas a um ou mais grupos de utilizadores atendendo às suas limitações físicas e/ou cognitivas.

Nome e Instituição	Sensores	Técnologia e metodos	Modo de utilização	Interface com utilizador
RobChair - instituto de sistemas e robótica universidade de Coimbra [33, 5]	LRF, odometria	Uso de Mapas de custos métrico, HECTOR SLAM, Planeamento híbrido baseado no A* e planeamento local com desvio de obstáculos, D-DWA (*Double-Dynamic Window Approach*).	Duas camadas de controlo cooperativo através de navegação com BCI.	BCI síncrono com deteção de estado de não controlo, *Joystick*, Teclado,
SHARIOTO - Katholieke Universiteit Leuven [6]	LIDAR (*Light Detection And Ranging*), Sensores Ultrasons, Giroscópio, Sensores infra-vermelhos	DWA (*Dynamic Window Approach*) para desvio de obstáculos.	Controlo partilhado com predição de intenção do utilizador baseada em redes bayesianas.	*Joystick*

LURCH - Politecnico di Milano [7]	Sonar; Sensores infravermelhos, LRF, Odometria	Planeamento SPIKE em ambiente conhecido.	Comportamento semi-autónomo fornece a solução para tarefas específicas, como entrar numa casa de banho. Em modo automático navega até o objetivo determinado pelo utilizador.	*Joystick*, Ecrã táctil, interface electromiográfico, BCI
Rolland - Universität Bremen[8]	Visão, Sonar, Sensores infravermelhos, Sensor de toque, Odometria	Navegação baseada em reconhecimento ótico de marcas.	Monitoriza e aprende as características do ambiente enquanto navega, para planeamento trajetórias, Aprende o desvio a obstáculos através de repetições.	*Joystick*
VAHM . University Paul Verlaine – Metz [9]	Sonar, Sensores Infravermelhos, Odometria .	Filtro de partículas para reconhecimento de caminhos mais frequentes através de mapa topológico existente.	Navegação autónoma através de mapas internos e navegação semi-autónoma no qual a VAHM desvia de obstáculos e segue paredes.	*Joystick*

Tabela 1- Cadeiras de rodas inteligentes

2.1.2. Interfaces de condução

A interface de condução é uma componente essencial da cadeira de rodas, pois, tal como o nome indica, faz a interface entre o utilizador e o sistema a controlar. Tem de estar adaptada às necessidades do utilizador, para que este consiga de forma efetiva fornecer comandos de condução. Apresentam-se de seguida as principais interfaces de condução usadas em cadeiras de rodas.

2.1.2.1. *Joystick*

O *Joystick* é o interface mais comum neste tipo de veículos robotizados (Figura 1 A.). Este interface não é dispendioso e é bastante fiável, podendo o utilizador selecionar diretamente a direção e quantificar a velocidade pretendida. Existem também *joysticks* com "*force feedback*" (Retorno de força) [10], podendo estes dificultar a aproximação a obstáculos através da aplicação de uma resistência ao movimento do manípulo do *joystick* na direção dos obstáculos, sendo esta força inversamente proporcional à distância a que se encontra o obstáculo. Para limitar algumas possíveis vibrações transmitidas ao interface pelo

utilizador, devido a limitações físicas ou cognitivas, pode ser aplicado um filtro Kalman aos sinais a serem transmitidos do interface [35].

2.1.2.2. *Joystick* de queixo, controlo de cabeça proporcional e botoneira

Algumas pessoas possuem limitações físicas que as impossibilita da utilização do *joystick* comum, pelo que existem outros tipos de atuadores, tais como os *joysticks* de queixo (Figura 1 B.), os atuadores de cabeça ou a botoneira (Figura 1 C.), que são concebidos para lhes fornecer um controlo efetivo sobre o equipamento a manusear.

Podemos descrever os *Joysticks* de queixo basicamente como adaptadores de interligação entre o queixo do utilizador e um *joystick*, sendo aplicável a utilizadores que possuam deficiência nos membros superiores que os impossibilitam de manipulação de um *joystick*. Para utilizadores que não possuam coordenação motora fina pode ser instalado como interface uma botoneira, que pela simples disposição de botões com funções específicas ou um simples botão com atuação codificada, isto é, com código Morse ou uma codificação mais simples, são selecionados os comandos a enviar à cadeira. No caso do controlador de cabeça proporcional, é através da proximidade ou contacto com sensores situados no encosto da zona posterior da cabeça e nas abas laterais do suporte, que são enviadas as ordens de acionamento, sendo também possível inverter o sentido de marcha através da atuação do sensor posterior.

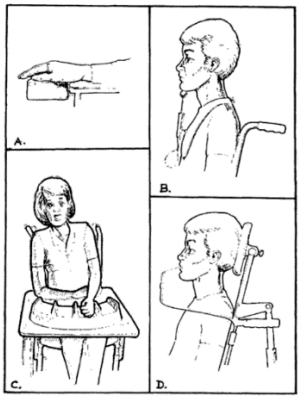

Figura 1- Dispositivos de controlo para cadeiras de rodas elétricas.A) Joystick de operação manual. B) Joystick adaptado de queixo. C) Botoneira. D)Tubo de sopro e sucção[11]

2.1.2.3. Tubo sopro e sucção

Para utilizadores que possuam incapacidade física de manipulação existem interfaces como o tubo de sopro e sucção (Figura 1 D.), que através de pressão ou depressão registada envia comandos para o movimento da cadeira. Estes equipamentos podem ser calibrados para cada utilizador específico através do ajuste de alguns parâmetros de pressão no controlador [12].

2.1.2.4. Ecrã táctil

Em certas situações pode não ser possível o uso de *Joystick* pelo utilizador, podendo existir um interface (gráfico por exemplo) através de um ecrã táctil com sectores de direção definida, permitindo a sua fácil seleção [13], ou até noutros casos, com mapas implementados, devendo o utilizador escolher destinos previamente selecionados, localizações em mapas estáticos ou mesmo dinâmicos, permitindo assim a seleção do destino pretendido através de um toque na zona da posição desejada [14].

2.1.2.5. Interfaces cérebro-computador

Este interface tem como potencial alvos pessoas com severas limitações motoras, sendo os comandos obtidos através da deteção de sinais elétricos no cérebro do utilizador. De momento este tipo de interfaces ainda se encontra em investigação, sendo normalmente a sua aplicação feita com algum tipo de controlo partilhado ou colaborativo para permitir uma navegação segura e com um nível aceitável de eficiência mesmo quando os comandos são esparsos e até pouco fiáveis [15, 33].

2.2. Perceção e planeamento de caminhos

A integração de perceção e localização nas cadeiras de rodas permite a estas aumentarem o seu nível de autonomia na navegação (entre semi-autónoma a completamente autónoma), o que pode representar grande relevância para utilizadores com níveis severos de incapacidade motora. A perceção pode ser realizada através de vários tipos de sensores, passivos ou ativos, que permitem a recolha de dados do ambiente envolvente, podendo-se assim tomar as respetivas ações necessárias ou desejadas. Os sensores usualmente mais utilizados são os LRF, LIDAR, sonar ou mesmo de visão, os quais permitem uma área de perceção abrangente com informação suficiente para tomada de decisões sobre ações de navegação a realizar. Esta perceção permite a aplicação de métodos de navegação local como o VFF (Virtual Force Field) [16] e o DWA (*Dynamic Window Approach*) [17]. Por serem dois métodos relacionados com a abordagem de navegação anti-colisão implementada neste trabalho, iremos descrevê-los de seguida.

O método VFF usa uma grelha bidimensional de tamanho fixo para representar o mundo, desta forma a mesma encontra-se centrada no robô, sendo atualizados os valores das suas células de acordo com o deslocamento da base robótica. Cada célula armazena um valor de certeza de ocupação C(i,j) por um objeto o qual é atualizado por cada varrimento do sensor, sendo o valor de certeza C(i,j) incrementado se a célula se mantiver ocupada no mesmo eixo acústico e na mesma distância a que se encontrava anteriormente.

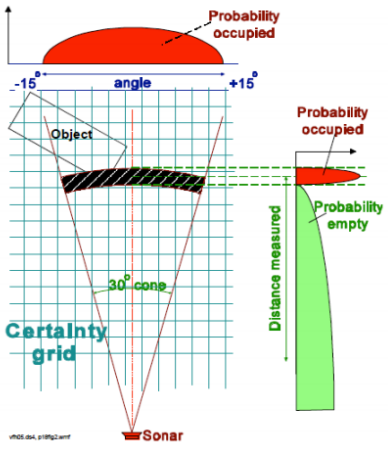

Figura 2- Projeção bidimensional de campo de visão e resposta probabilística considerando leituras obtidas por sonar [18]

Para cada célula, é calculada uma força repulsiva, F(i,j), sendo esta diretamente proporcional ao valor de certeza e inversamente proporcional à distância ao robô. O O somatório destas forças resultam numa força de repulsão, F_r, causada pelos obstáculos existentes. É também calculada uma força de atração, F_a, de acordo com a distância ao objetivo selecionado. As constante F_{cr} e F_{ct} são parâmetro de ponderação das forças de repulsão e atração.

$$F(i,j) = \frac{F_{cr}\, C(i,j)}{d^2(i,j)} \left(\frac{x_i - x_0}{d(i,j)}, \frac{y_i - y_0}{d(i,j)}\right) \quad (1)$$

Onde

C(i,j) – valor de certeza de ocupação da célula (i,j)

d(i,j) – distância entre a célula (i,j) e o robô

x_i, y_i – posição da célula (i,j)

x_0, y_0 – posição do robô

$$F_r = \sum_{i,j} F(i,j) \quad (2)$$

$$F_a = F_{ct} \left(\frac{x_t - x_0}{d}, \frac{y_t - y_0}{d}\right) \quad (3)$$

Através da soma destas duas forças obtemos a força resultante que indicará a direção a ser aplicada ao robô conforme exemplificado pela Figura 3

$$R = F_a + F_r \quad (4)$$

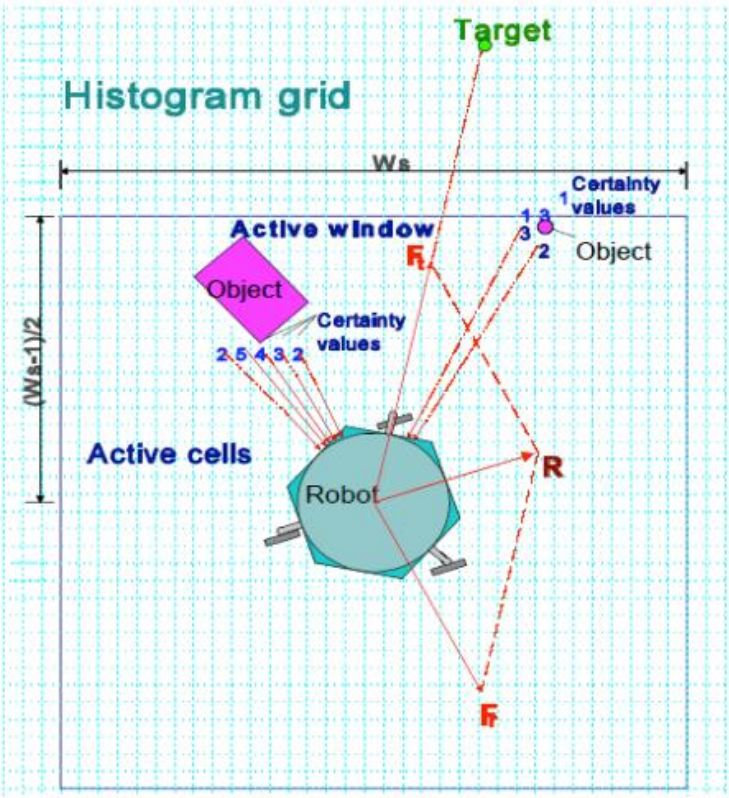

Figura 3- Exemplo de campos potenciais aplicados num robô [18]

O método DWA [17] obtém as velocidades de controlo a enviar para o robô diretamente através da análise das possíveis velocidades no espaço envolvente. O campo de dados para análise é reduzido ao verificarmos apenas as velocidades possíveis de acordo com a dinâmica do robô. No primeiro passo desta análise são apenas consideradas velocidades seguras para as possíveis trajetórias, isto é, velocidades que realizem um deslocamento onde não seja possível a ocorrência de colisão, sendo no passo seguinte verificada qual a velocidade que maximiza a função para o objetivo pretendido.

Pela aplicação de uma velocidade (v, ω) ao robô, são realizados pequenos deslocamentos que descrevem uma trajetória circular, sendo que cada trajetória circular pode apenas ser definida por um único vetor de velocidade (v, ω). No caso de existência de obstáculos perto do robô, irão ocorrer restrições às velocidades lineares e angulares, sendo a velocidade máxima admissível do robô calculada através da distância ao obstáculo nesta trajetória circular.

Assim, sendo o termo dist(v, ω) representativo da distância correspondente ao obstáculo mais perto quando aplicada a velocidade (v, ω), tendo como valores de desaceleração/travagem $\dot{v}_t$ e $\dot{\omega}_t$ e podemos calcular as velocidades admissiveis V_a, sabendo que apenas serão consideradas admissiveis se o robô se conseguir imobilizar antes de atingir o obstáculo:

$$V_a = \left\{ (v, \omega) \mid v \leq \sqrt{2.dist(v, \omega).\dot{v}_t} \ \wedge \ \omega \leq \sqrt{2.dist(v, \omega).\dot{\omega}_t} \right\} \qquad (5)$$

Tendo em conta os possíveis valores de aceleração dos motores, o espaço de análise pode ser reduzido apenas ao espaço atingível no próximo intervalo de tempo t. Sendo as acelerações $\dot{v}_{ac}$ e $\dot{\omega}_{ac}$ e sendo (v_{at}, ω_{at}) a velocidade atual podemos determinar as velocidades V_d que pertencem à janela dinâmica como

$$V_d = \{(v, \omega) | v \in [v_{at} - \dot{v}_{ac}.t, v_{at} + \dot{v}_{ac}.t] \wedge \omega \in [\omega_{at} - \dot{\omega}_{ac}.t, \omega_{at} + \dot{\omega}_{ac}.t]\} \qquad (6)$$

Na Figura 4 podemos verificar um exemplo da janela dinâmica calculada com os pressupostos anteriormente referidos.

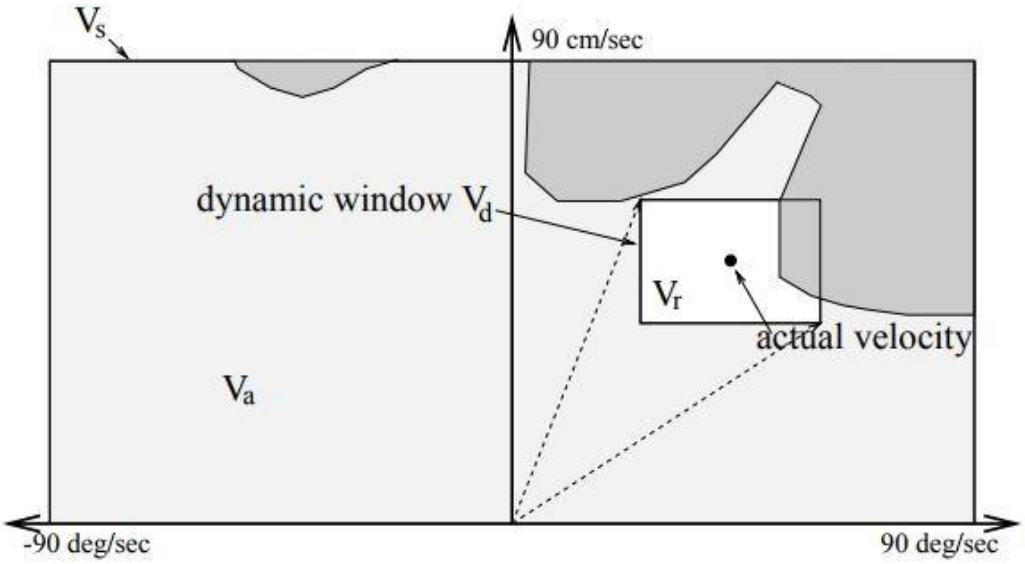

Figura 4- Exemplo de Janela Dinâmica [17]

Através das restrições definidas e sabendo que Vs é o conjunto de velocidades que permitem percorrer o espaço de percepção para atingir o objetivo pretendido, podemos realizar a intersecção dos vários resultados, obtendo assim um conjunto de velocidades resultantes Vr, as quais realizam um deslocamento permitido sem colisão (área branca representada na Figura 4), para posterior análise de maximização de acordo com a localização do objectivo a alcançar.

$$Vr = Vs \cap Va \cap Vd \quad (7)$$

Para maximizar a função do objetivo serão incorporados nas velocidade resultantes Vr os critérios de direção, distância e velocidade com as respetivas ponderações α, β e γ ajustadas de acordo com as especificidades do projeto da seguinte forma:

$$G(\boldsymbol{v},\boldsymbol{\omega}) = \sigma\big(\alpha\,.\,direção(\boldsymbol{v},\boldsymbol{\omega}) + \beta.distância(\boldsymbol{v},\boldsymbol{\omega}) + \gamma.velocidade(\boldsymbol{v},\boldsymbol{\omega})\big) \quad (8)$$

Sendo a função $direção(\boldsymbol{v},\boldsymbol{\omega})$ responsável pela medição da progressão do robô em direção ao objectivo a atingir, a função $distância(\boldsymbol{v},\boldsymbol{\omega})$ responsável por verificar a distância do arco da trajetória ao obstáculo mais perto, sendo que a menor distância ao objeto implica a maior intenção de se afastar do mesmo. A função $velocidade(\boldsymbol{v},\boldsymbol{\omega})$ representa a velocidade do robô. O termo σ é aplicado para suavizar a soma ponderada das três funções descritas

2.3. ROS – Robotic Operating Systems

O ROS [19] foi a camada *middleware* escolhida para a implementação do nosso projecto. A função desta camada é permitir o agrupamento de vários periféricos e funções sem a necessidade de desenvolvimento de *software* específico para os mesmos. A título de exemplo existem também os seguintes middleware: Player Project; RT – Middleware Projects; Urbi e Micro entre outros.

O ROS é um sistema operativo desenvolvido com base em Linux, especialmente dedicado a aplicações robóticas. Este é baseado numa plataforma de *OpenSource* disponibilizando várias interfaces e funções, amplamente ensaiadas e funcionais, que nos permitem uma fácil integração e interação entre os vários sensores, atuadores e manipulação de dados necessários, evitando assim todo o esforço de desenvolvimento de *software* já existente.

O ROS teve a sua primeira versão em 2010 com a distribuição Box Turtle, tendo lançado até á data 9 novas versões, estando identificadas as quatro últimas na Tabela 2, sendo a mais recente a ROS Kinetic Kame.

Distro	**Release date**	**Poster**	***Tuturtle*, turtle in tutorial**	**EOL date**
ROS Kinetic Kame (**Recommended**)	May 23rd, 2016			May, 2021
ROS Jade Turtle	May 23rd, 2015			May, 2017
ROS Indigo Igloo	July 22nd, 2014			April, 2019 (Trusty EOL)
ROS Hydro Medusa	September 4th, 2013			May, 2015

Tabela 2- Recentes distribuições de ROS

O ROS baseia-se no conceito da criação de pacotes de *software* dedicados a cada projeto ou aplicação. A partilha de informação entre os diversos participantes do projeto é feita através de nós que publicam e subscrevem a informação (mensagem) necessária organizadas por tópicos, conforme exemplificado na Figura 5, ficando esta acessível a outros participantes que necessitem ou queiram partilhar informação sobre os mesmos tópicos.

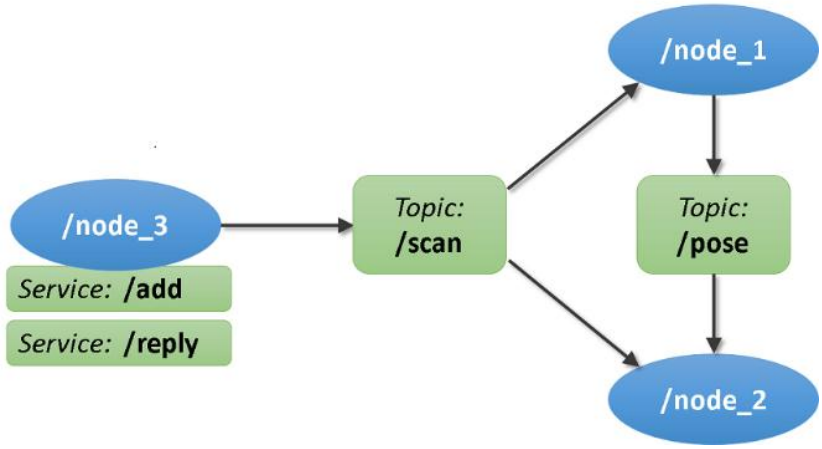

Figura 5- Estrutura de Nós e Tópicos - ROS

O desenvolvimento dos programas podem ser realizados em C++, Phyton ou Lisp. Devido à escalabilidade possível através do ROS, este é apropriado para o desenvolvimento de sistemas complexos e dimensão, sendo este *software* usado por grandes empresas como a DARPA e a Softbank Robotics.

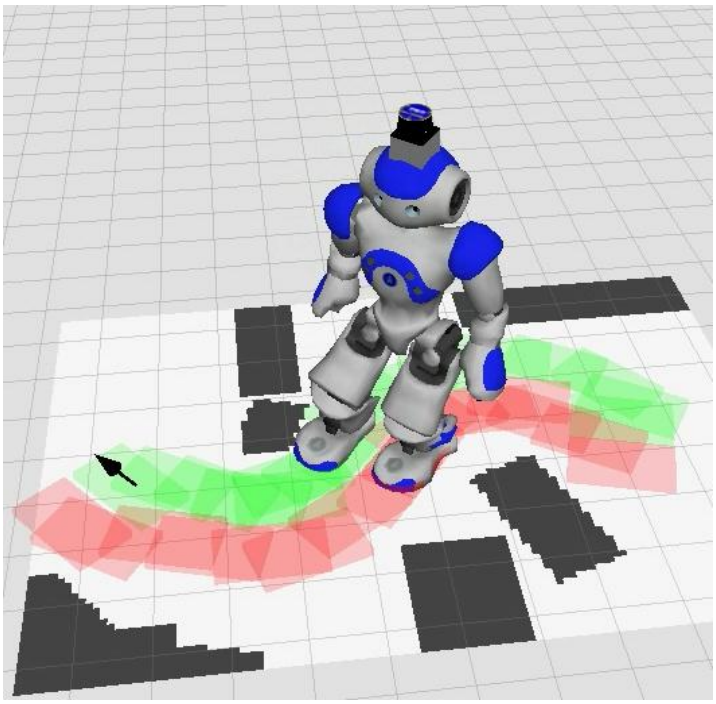

Figura 6- Softbanks Robotics - Varrimentos 3D e pegadas em rviz.[19]

Em 2010 a Softbanks Robotic utilizou o seu robô humanoide Osiris para permitir realizar o desenvolvimento da localização do robô em ambientes complexos Este robô tem 58cm e incorpora um LSR no topo da cabeça, tendo sido usado também para o desenvolvimento dos movimentos de subir escadas e a imitação de vários movimentos humanos exemplificados na Figura 6. Devido aos movimentos exercidos pelo corpo na sua deslocação torna-se bastante mais difícil a correta perceção da localização e mapeamento. Através da aplicação de novos métodos e de leituras dos dados recolhidos de vários sensores que o robô tem integrado, tais como do laser, da odometria, do IMU e dos dados propriocepção, foi possível obter a localização 6D do torso do Osiris, realizando assim o mapeamento do mesmo. Foi assim desenvolvida na Universidade de Freiburg uma plataforma de navegação para esta classe de robôs, a qual incluía uma biblioteca sobre os movimentos de caminhar.

Neste momento a Softbanks Banks possui o projeto Romeo, tratando-se de um robô humanoide de 1,40 m com perceção multissensorial permitindo a sua interação com o ambiente circundante, realizando interação física e cognitiva com humanos (Figura 7).

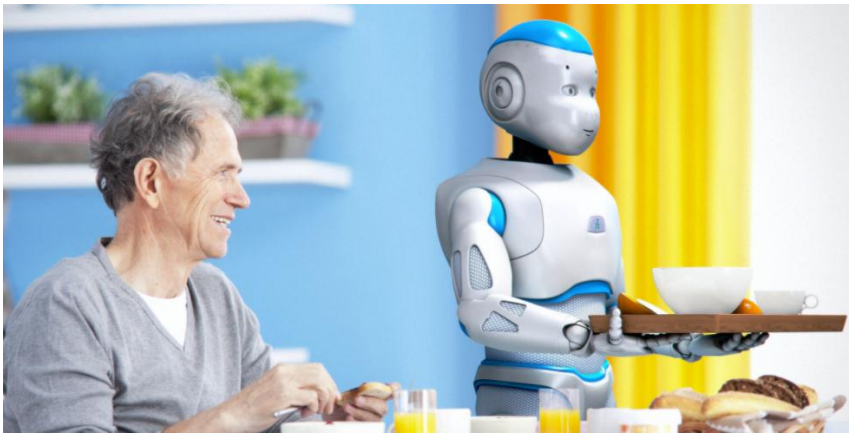

Figura 7- Romeo - Aldbaran e SoftBank Robotics [34]

Na última distribuição Kinetic Kame encontram-se já implementadas as primeiras versões de pacotes de *software* de piloto automático para drones desenvolvido em ROS nativo, os quais incluem pacotes de navegação, mapeamento, desvio de obstáculos e outros. Este pacote de *software* pode ser aplicado quer a drones para corridas, quer a drones para

transporte de carga, sendo que os diferentes blocos de *software* comunicam usando a arquitetura de publicação e subscrição de mensagens.

3. Arquitetura de *Hardware* e *Software*

Neste capítulo encontram-se descritos os principais elementos que compõe o sistema, estando os principais módulos identificados na Figura 8.

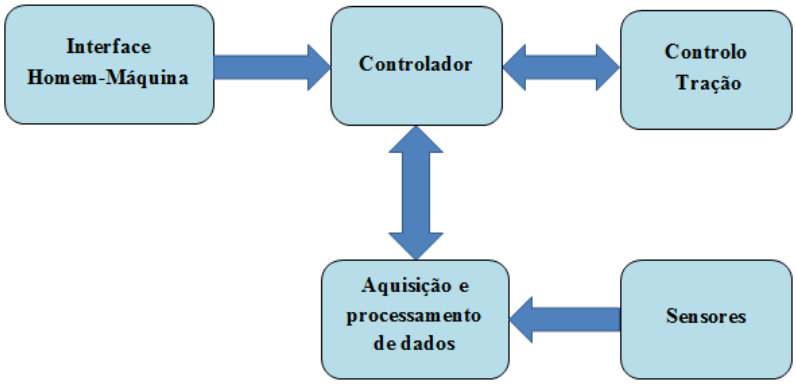

Figura 8- Arquitetura do sistema

3.1. Plataforma base – Salsa R2

Como plataforma base para o desenvolvimento deste projeto, foi utilizada uma cadeira de rodas elétrica modelo Salsa R2 (Figura 9), existente no laboratório VITA.IPT. Esta cadeira possui uma estrutura onde é possível incorporar vários tipos de assentos inclináveis, assim como configurar vários tipos de interfaces para o utilizador, conseguindo assim obter o melhor desempenho de acordo com as necessidades de cada utilizador.

Figura 9- Cadeira de rodas elétrica Salsa R2 [20]

Os vários módulos constituintes dos circuitos de comando e controlo da Salsa R2 encontram-se interligados de acordo com o esquema apresentado na Figura 10.

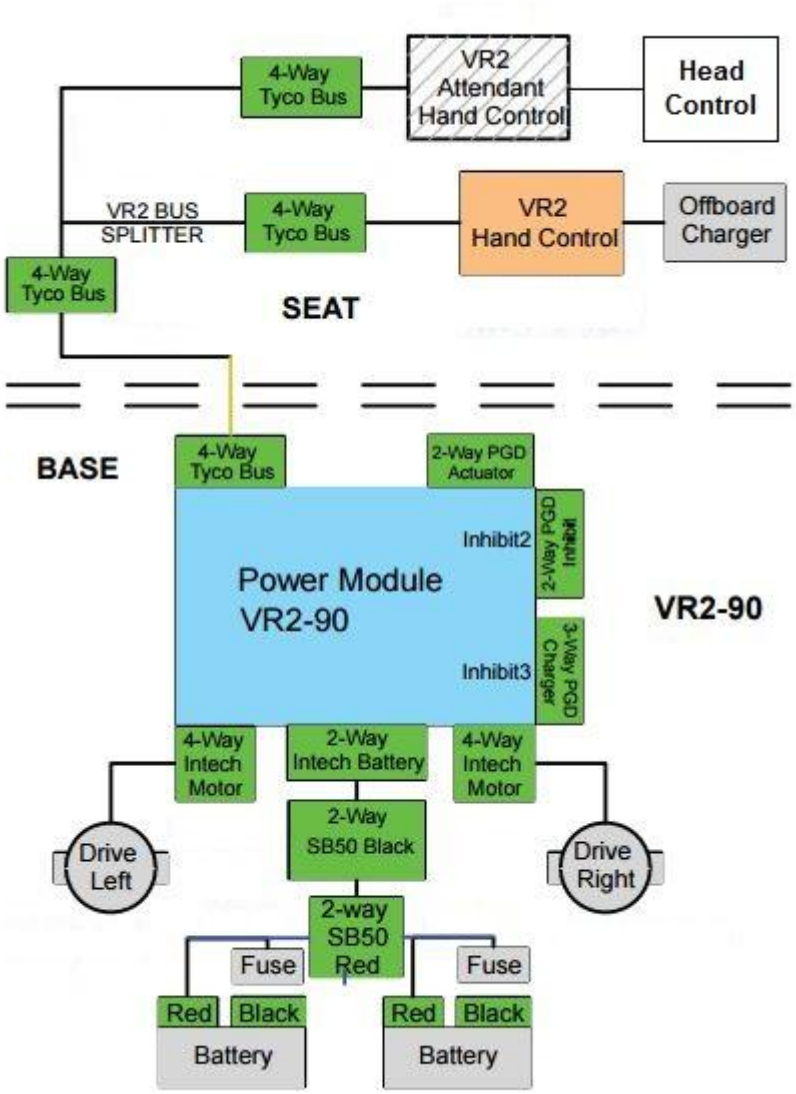

Figura 10- Diagrama principal de ligações VR2 and RNET[20]

Este sistema é constituído por duas baterias de 12 V ligadas em série para o fornecimento de 24 V ao módulo de potência VR2-90. Neste módulo estão incorporadas as ligações de potência aos motores e também um barramento Tyco que interliga vários dispositivos de controlo, como o *joystick* (VR2 *Hand Control*) e o interface de configuração do sistema (VR2 *Attendant Hand Control*). Encontra-se também neste sistema um controlador de cabeça, atuado por contacto/pressão de sensores localizados nas partes

laterais e traseira do mesmo. Integrado no sistema existe uma tomada para ligação de um carregador de baterias externo, permitindo assim o seu carregamento e a reposição da sua autonomia.

O barramento Tyco é composto por 1 par de linhas de alimentação de 24V e outro de comunicação, realizando assim a interligação de todos os seus módulos. Devido a este sistema ser um sistema comercial e não serem disponibilizadas pelo fabricante as especificações técnicas sobre o protocolo aplicado neste barramento, não foi possível a descodificação do mesmo para utilização dos módulos existentes (e.g., *joystick*) e integração dos mesmos no projeto. Pela análise destes módulos podemos constatar que os mesmos possuem processadores internos, conforme ilustrado na Figura 11, o que torna mais complexa a análise de dados existentes no barramento, devido ao vasto processamento interno realizado por cada equipamento, tendo inviabilizado o procedimento de engenharia inversa.

Figura 11- Microprocessadores do módulo de potência e do módulo de Joystick

3.2. Robotização da cadeira de rodas elétrica

Dados os constrangimentos referidos na secção anterior, optou-se por implementar uma arquitetura completamente nova, mantendo o sistema original montado na cadeira completamente funcional. A nova arquitetura de aquisição sensorial e comando da cadeira de rodas encontra-se na Figura 12.

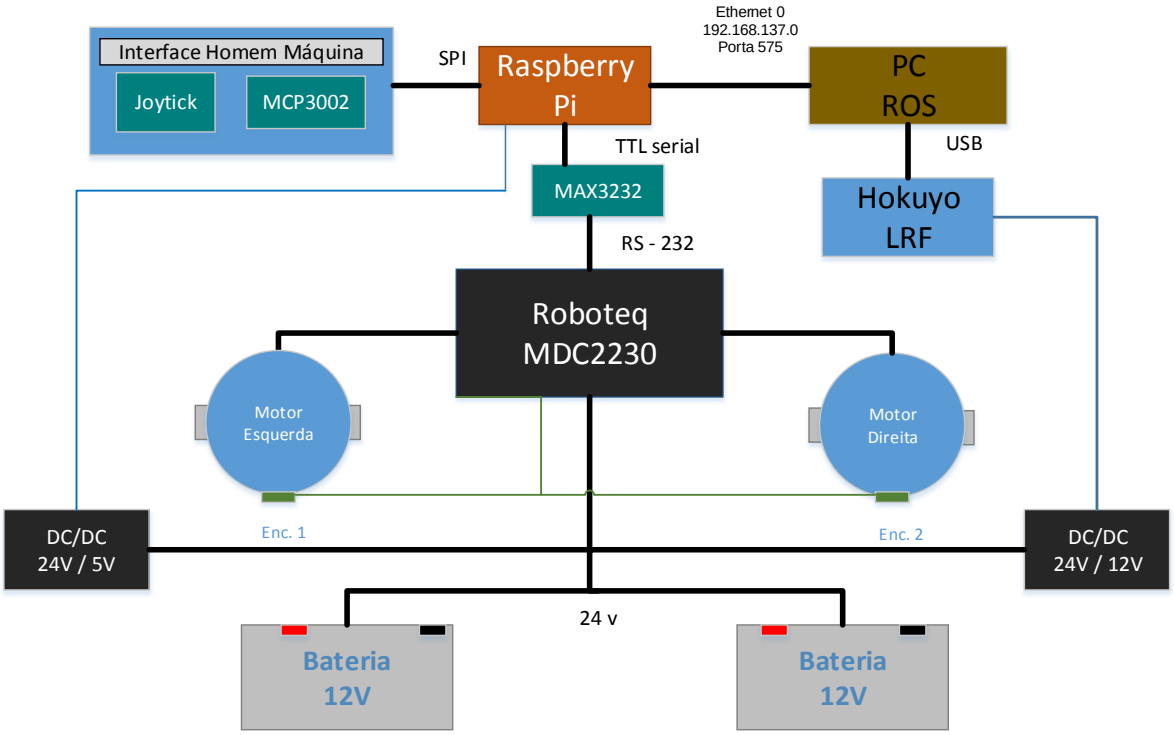

Figura 12- Estrutura de implementação da cadeira de rodas robótica

A alimentação do nosso sistema é assegurada pelas duas baterias da Cadeira Salsa R2 ligadas em série, fornecendo assim 24 V DC ao controlador de potência MDC2230, e aos dois conversores DC/DC de 5V e 12V que fornecem energia ao RPi e ao LRF, respetivamente,

O módulo Roboteq MDC2230 é o responsável pelo controlo de tração, comandando assim os dois motores da Sunrise Medical, onde foram mecanicamente incorporados os sensores taquimétricos. A alimentação e processamento dos sinais dos *encoders* também é responsabilidade do módulo Roboteq.

Para a interface de condução foi integrado um *joystick* resistivo e um conversor Analógico/Digital para ler a informação do *joystick* uma vez que o RPi não possui entradas analógicas para ligação direta a este dispositivo. O módulo RPi é o responsável pela receção de comandos do módulo de interface e envio dos comandos para o controlador MDC3220, sendo esta comunicação realizada através de uma ligação série. Para a realização da ligação série foi necessário introduzir um conversor RS232- TTL MAX3232 [21] para

compatibilizar os sinais entre ambos os equipamentos. O RPi recebe os comandos diretamente do *joystick*, transmitindo-os ao controlador de tração no modo livre ou ao módulo de navegação anti-colisão, através de uma ligação Ethernet que encapsula os pacotes UDP, que se encontra a correr no PC.

O módulo de navegação anti-colisão está implementado no sistema ROS, sendo, responsável pelo tratamento e validação das velocidades aplicar ao sistema. Esta análise é realizada através dos dados recebidos do sensor LRF Hokuyo diretamente ligado ao PC.

3.2.1. *Raspberry Pi* 1 Model B

O *Raspberry Pi* (RPi) é um microcomputador baseado num processador 700 MHz ARM1176JZF-S core o qual integra 512 Mb de RAM. Possui duas portas USB, 1 porta Ethernet e uma saída HDMI entre outras. Possui também um *slot* para cartão SD onde se encontra instalado o sistema operativo Debian baseado em Linux.

Figura 13- Raspberry Pi 1 Modelo B [22]

A alimentação do RPi é feita através da ficha micro-usb presente na placa, não estando esta ligação preparada para a transferência de dados. O RPi possui também um conjunto de pinos de entrada/saída de designação GPIO (*General purpose input/output*) de uso geral, cuja funcionalidade se encontra identificada na Figura 14. Alguns destes pinos têm associado funções específicas as quais algumas foram implementadas no nosso sistema conforme será descrito de seguida.

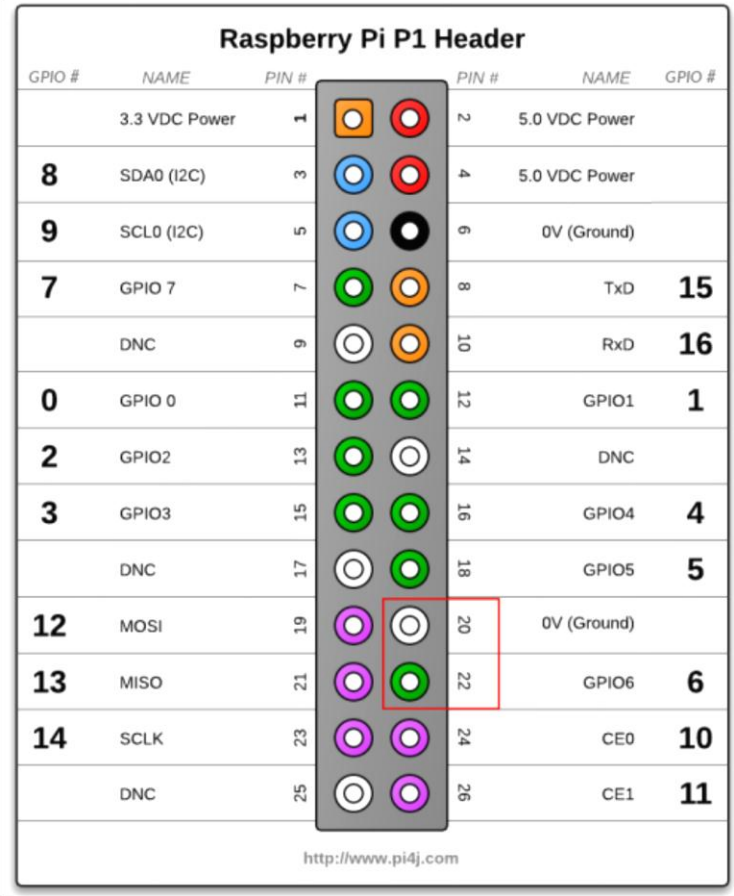

Figura 14- GPIO Pinout do RPi

A comunicação realizada para o módulo de potência é realizada através dos pinos de comunicação série GPIO 15 e 16 do RPi. Devido ao níveis de tensão na porta série do RPi serem do tipo TTL (Transistor-Transistor Logic), isto é, o valor digital 0 e 1 são representados por valores de tensão de 0V e +5V respetivamente, é necessário converter estes sinais para o standard RS232 de -15V e + 15V (0 e 1 lógico respectivamente). Assim foi necessária a implementação de um conversor TTL – RS232 realizado com o integrado MAX3232 para estabelecer a comunicação com o módulo de potência. Os parâmetros da comunicação como *baudrate*, número de bits e paridade são programados através de *software*.

Aos pinos GPIO 10, 12, 13 e 14 do RPi está associada a comunicação SPI cujas configurações de comunicação, tal como no caso anterior, são realizadas ao nível da programação do *software*.

Outra interface realizada pelo RPi é a comunicação com o computador que será responsável pelo processamento da odometria e assistência á navegação. Esta interface é realizada através da rede Ethernet, tendo sido parametrizado o IP do RPi definido de forma como estática (192.168.1.5) a fim de se ligar diretamente à rede do computador.

3.2.2. Joystick

Foi inicialmente proposto a utilização de um Pretorian Technologies n-ABLER *Joystick*, tendo-se verificado após a implementação que o mesmo possui uma baixa resolução nos valores de comando, não sendo assim aconselhável à nossa aplicação. Desta forma optamos pela integração de um Keyes-SJoys *Joystick* (Figura 15), mais acessível e económico, constituído apenas por dois potenciómetros e um interruptor de pressão. Podendo ser alimentado por 5 ou 3.3 V, tendo valores de saída entre Vcc e GND.

Figura 15- Joystick Keyes_SJoys

Devido ao RPi não possuir entradas analógicas, optámos por intercalar um integrado MCP3002 – Conversor A/D 10 bit de dois canais com ligação *Serial Peripheral Interface* (SPI). Na Figura 16 está exemplificada a troca de dados existentes no bus, sendo o D_{IN} os dados de seleção de canal enviados pelo RPi e D_{OUT} o sinal analógico codificado a 10 Bit. Através da comunicação SPI é possível interligar diretamente o integrado e as respetivos pinos de comunicação do RPi, tendo apenas sido desenvolvida uma nova biblioteca para correta leitura dos dados entre ambos os equipamentos.

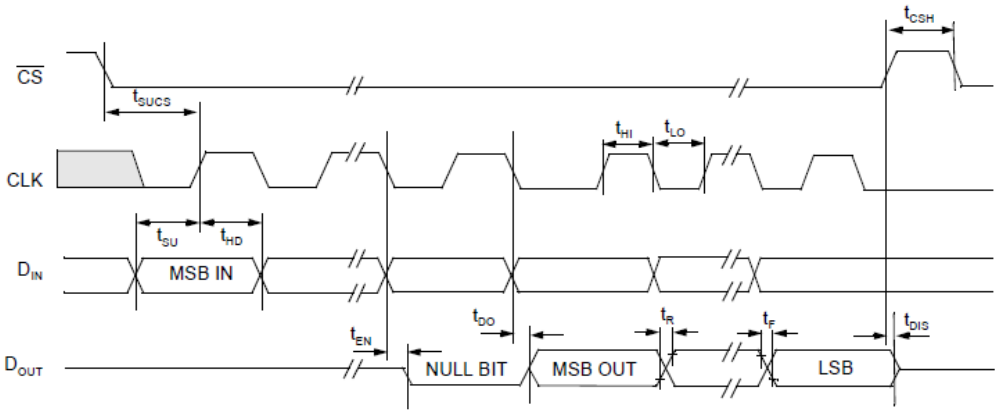

Figura 16- Sincronismo série MCP3002 [23]

Para evitar tendências e pequenos desvios provocados por características próprias do *joystick*, foi implementada, ao nível de programação, uma pequena calibração e mapeamento dos valores obtidos do conversor MCP3002, de 0 a 1024 para -1000 a 1000, aplicando uma zona morta entre os valores -20 e 20.

3.2.3. Roboteq MDC2230

O Roboteq MDC2230 (Figura 17), é um controlador de motor desenvolvido para receber comandos através de várias interfaces de comunicação: Série, CAN, Radio, *wireless*, etc. e ativar as saídas de potência para controlo de um ou dois motores de corrente contínua. No nosso projeto optámos por usar a comunicação série RS232 como canal de comunicação com o RPi.

Este controlador para disponibiliza várias funções implementadas no seu *hardware* que permitem o utilizador configurar facilmente o controlador [25]. Através da ligação USB ao computador e com o programa Roborun+ foram configuradas e programadas ações de segurança como *watchdogs*, supervisão de tensão e correntes; parâmetros de configuração como velocidade máxima, aceleração máxima, impulsos por rotação; valores de controlo como velocidade dos motores, número de impulsos dos *encoders*, valores de tensão, etc.

Figura 17- Roboteq MDC2230 [24]

O comando das saídas de potência podem ser realizadas individualmente pelo canal 1 e Canal 2 como comandos independentes para cada motor ou podem ser comandadas, conforme optado por nós, em modo misto, sendo o canal 1 responsável pelo comando da velocidade linear e o canal dois responsável pelo comando da velocidade angular.

Para garantir a segurança do utilizador foi implementada uma botoneira de emergência conforme referido no manual do controlador e identificado na Figura 18. Caso esta emergência seja acionada deve ser reiniciado o controlador.

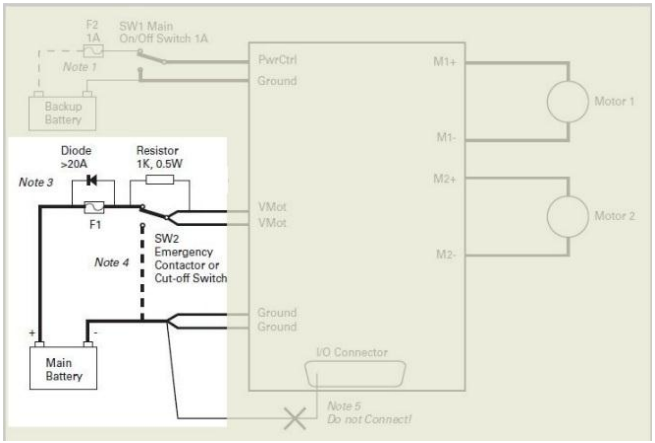

Figura 18- Alimentação do controlador com destaque para a implementação de paragem emergência [24]

Outra funcionalidade também importante e disponível no Roboteq é a possibilidade de controlo em malha fechada PID, podendo esta função ser realizada em controlo de velocidade ou posição. Para o nosso projeto o controlador foi configurado para controlo de velocidade de malha fechada, tendo sido obtidos pelo método experimental os seguintes valores que se encontram desenvolvidos no Anexo I:

$$G_{DC} = 0.85 \qquad \tau = 0{,}69 \text{ s} \qquad (9)$$

Tendo sido obtidos os valores para o controlador de:

$$\text{Proporcional} : K_p = 1.63 \quad \text{Integral} : k_i = 2.61 \quad \text{Diferencial} : K_d = 0.45 \qquad (10)$$

Através da ligação USB disponível no controlador Roboteq e da aplicação Roborun+ é também possível simular e monitorizar o desempenho do controlador e dos motores com o computador.

3.2.4. Motores e *Encoders*

Os motores utilizados no projeto são os que já se encontravam na cadeira Salsa R2. São motores de 22.5V DC com uma caixa de engrenagem integrada de relação 1:26 e de 163 rpm. Verificámos que os motores em causa não possuíam *encoders*, pelo que foi necessário estudar a forma de aplicação destes sensores nos motores. Desta forma foi decidido remover os freios eletromagnéticos para ser possível aceder ao eixo de rotação do motor para acoplar o sensor (Figura 19). Verificámos que é necessário os freios eletromagnéticos permanecerem ligados ao sistema original para o mesmo continuar funcional sem falhas, tendo os freios sido ligados mas não acoplados aos motores.

Figura 19- Desmontagem de freio eletromagnético e aplicação de encoder no veio do motor

Devido ao tamanho disponível do veio para aplicação dos *encoders* e aos campos eletromagnéticos ali existentes, optamos pela aplicação de *encoders* capacitivos AMT103-V [26].

Figura 20- Encoder AMT103-V [26]

Estes *encoders* são bastante versáteis, pois são fornecidos com uma base de montagem e um kit de adaptadores conforme ilustrado na Figura 20, permitindo o correto ajuste ao veio para aplicação do *encoder*. É também possível configurar os impulsos por revolução através da configuração de um "*dip switch*" existente no *encoder*. No caso do nosso modelo estes valores são configuráveis de 48 a 2048 ppr, tendo sido configurado para 1024 ppr.

3.2.5. Hokuyo UTM-30LX *Laser Rangefinder*

Para implementação do Sistema de perceção foi usado um laser *Rangefinder* Hokuyo UTM-30LX [27] (Figura 21). Este LRF tem uma alimentação de 12V DC e realiza um varrimento de 270º com alcance de 30 m em 25 ms, sendo um modelo leve e compacto, com uma resolução angular de 0,25º, tendo uma precisão de ± 30mm de 0,1 a 10m e ± 50mm de 10 a 30m de distância.

Figura 21- Hokuyo UTM-30LX [27]

Para adaptar o LRF à cadeira de rodas foi analisado qual o melhor local para montagem do mesmo, evitando possíveis impactos e vibrações. Desta forma optamos por colocar o LRF nas barras de suporte dos apoios de pés existentes na cadeira, conforme Figura 22, permitindo desta forma realizar um varrimento superior a 180 º sem obstáculos provenientes do utilizador.

Figura 22- Suporte de montagem do LRF

3.2.6. Placa de alimentação e interligação de periféricos

Para melhor acondicionamento e integração dos componentes periféricos foi construída uma placa para montagem dos conversores DC/DC que servem de fonte de alimentação aos módulos de Laser (12 V) e ao RPi (5 V), assim como a placa de circuito impresso onde se encontram as fichas de ligação ao controlador Roboteq e os vários interruptores para isolamento das respetivas alimentações envolvidas. Na placa de circuito impresso foram também implementados dois led´s sinalizadores: sistema em funcionamento (Vermelho intermitente) e Sistema de navegação assistida ativo (Azul). A placa e respetivo esquemático encontram-se respetivamente nas Figura 23 e Figura 24.

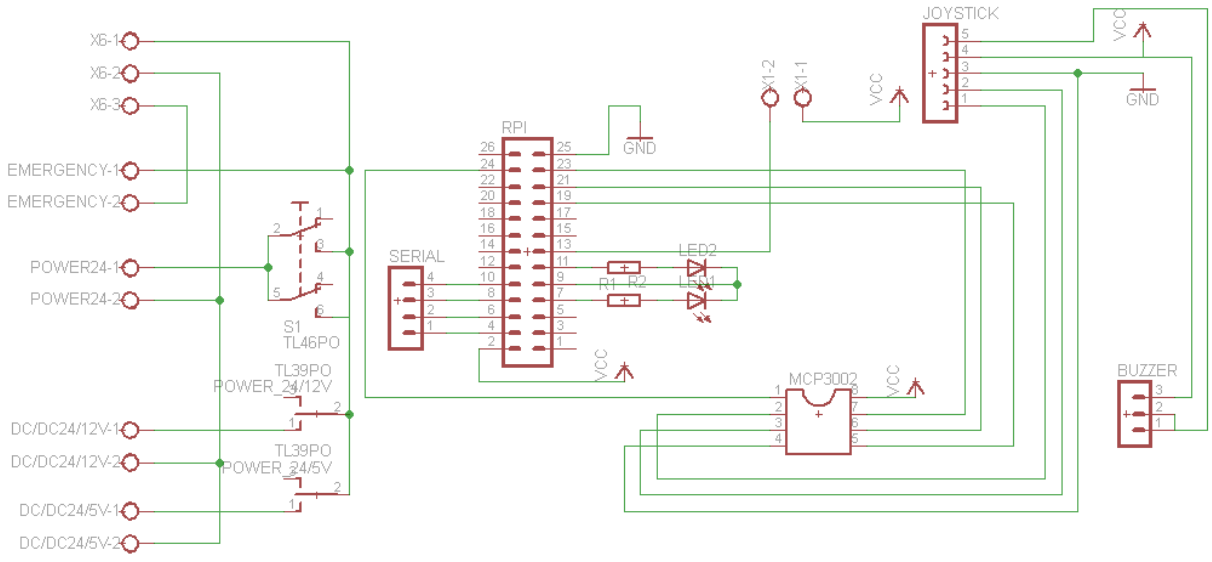

Figura 23- Esquema elétrico da placa circuito impresso

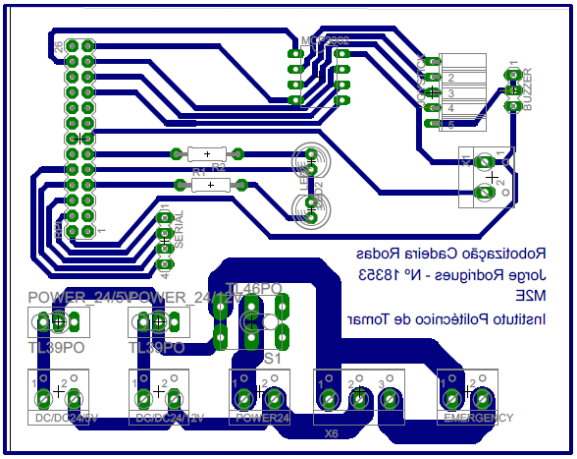

Figura 24- Placa de circuito impresso

3.2.7. Notebook Asus F200M

Foi definido que o processamento de dados relacionados com a navegação iriam ser realizados através de sistema ROS, pelo que se tornou necessário o uso de um *notebook*, optando-se por um Asus F200M, Figura 25, já existente no Laboratório VITA.IPT com sistema operativo Ubuntu 12.4 de 64bits, onde foi instalada a versão Hydro do ROS.

Este notebook de 11,6'' possui um processador Intel Celeron N2830 de 2.4GHz, estando equipado com 2 GB de RAM e um disco de 500 GB. Tem duas portas USB 2.0 e uma USB 3.0, sendo uma usada para ligar o nosso LRF, uma porta Ethernet, usada para realizar a ligação ao RPi. Possui ainda portas HDMI e VGA, assim como ligação *Bluetooth* e Wifi.

Figura 25- Asus F200M

3.3. Arquitetura de *software*

A fim de tornar este projeto num sistema modular, ao qual seja possível interligar outros módulos ou funcionalidades, foi desenvolvido no RPi o *software* responsável pela leitura do *joystick*, envio de comandos de velocidade para o controlador de potência Roboteq, obtenção de dados do mesmo controlador e processamento de dados para envio à aplicação externa através da porta Ethernet, no nosso caso, para o ROS, conforme ilustrado na Figura 26.

Foi implementada a seguinte estrutura a fim de otimizar a programação necessária:

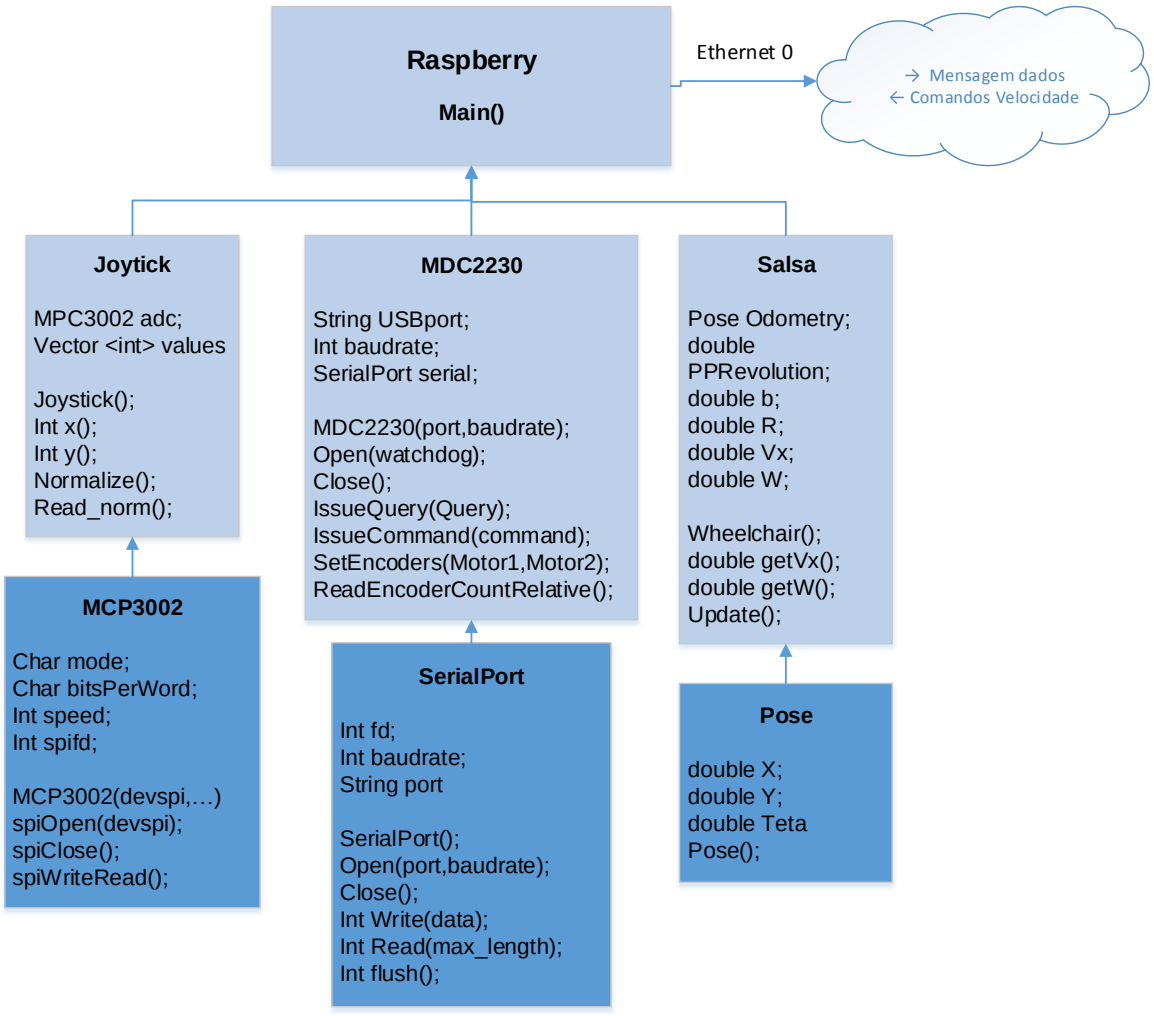

Figura 26- Estrutura do Software no RPi

O desenvolvimento do módulo *Joystick* permite-nos a substituição do *joystick* por outro controlador analógico, cujo ponto de repouso seja o valor central do mesmo, sendo o seu valor lido e enviado pelo módulo "MCP3002". O módulo *software* "MPC3002" foi desenvolvido para realizar a leitura analógica do dos dois canais do *joystick* e enviar os seus valores através de comunicação SPI para o RPi. A seleção do canal a ser lido é codificada no comando de leitura enviado pelo RPi para o conversor, sendo realizada a transferência de 2 bytes por leitura de canal, sendo o sinal analógico codificado em 10 bits.

A classe “SerialPort” foi implementada para comunicação com o controlador de potência, podendo assim enviar comandos e receber estados, permitindo o controlo da tração do nosso objeto “Salsa”, sendo alguns dados recebidos do controlador de potência partilhados com a classe “Pose” para cálculo da posição e deslocamento da base robótica. A porta série atribuída para esta comunicação foi "/dev/ttyACM0" com a seguinte configuração:

- 115200 bits/s
- 8-bit data
- 1 Start bit
- 1 Stop bit
- Sem Parity

Neste módulo foi também implementado um canal de comunicação Ethernet, com configuração de IP estático 192.168.1.2 porta:5171, para comunicação com o PC Asus e respetivo *software* ROS

Foram também configuradas as entradas e saídas do módulo GPIO (*general purpose input/output*) com as seguintes funcionalidades

- PIN GPIO 2 – Input – Este pino foi configurado como a entrada do seletor de modo de navegação Livre / Sistema de Anti Colisão. Este seletor está á disposição do utilizador para realização de manobras sem perceção do ambiente pelo sistema.

- PIN GPIO 0 – Output – Neste pino foi implementado um led azul para sinalização da navegação assistida ativa.
- PIN GPIO 7 – Output - Devido ao RPi não possuir nenhum elemento de interface de visualização de funcionamento, foi implementado um led nesta saída que nos permite verificar se o código se encontra em correto funcionamento através da sua presença intermitente a uma baixa frequência. Caso não se verifique esta comutação de estado do led, indica que o *software* dedicado se encontra inativo.

3.4. Fluxograma de *software* do controlador RPi

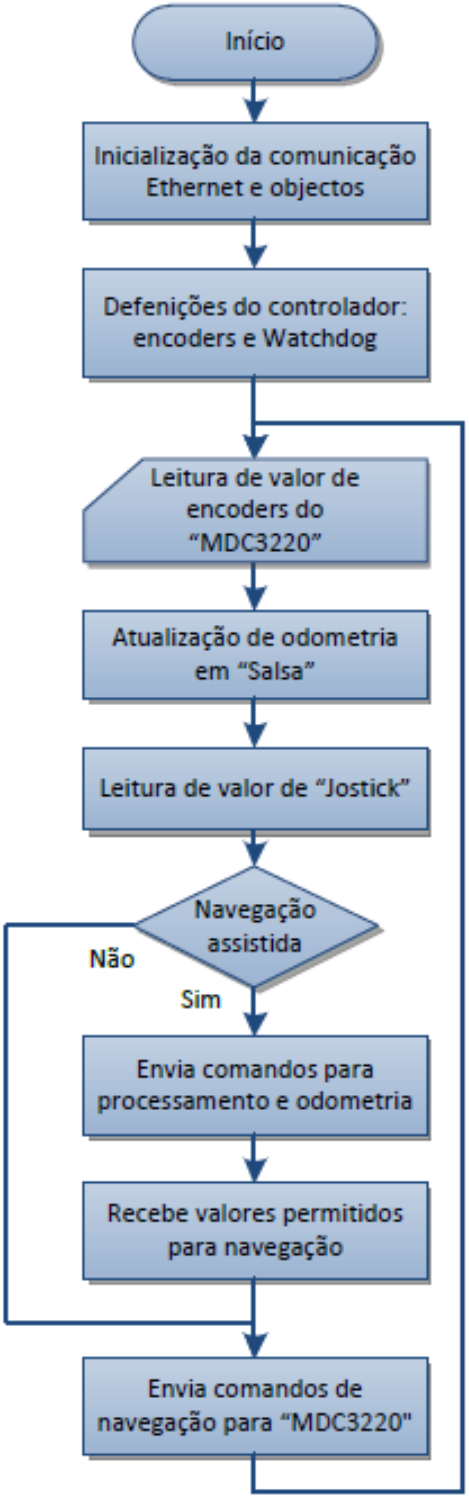

Figura 27- Fluxograma de software do controlador RPi

4. Localização e Navegação

4.1. Odometria para robôs diferenciais

A odometria é um dos métodos mais utilizados para estimar a posição de um robô, o qual se baseia na integração incremental do movimento do robô ao longo do tempo, causando por este facto uma inevitável acumulação de erros, sendo de qualquer forma considerada uma parte muito importante na navegação do robô [28].

4.1.1. Modelo dinâmico

Os robôs diferenciais são robôs que possuem controlo independente sobre cada uma das duas rodas motrizes. Com a aplicação de igual velocidade de rotação a ambas as rodas o robô desloca-se para a frente, enquanto a mudança de direção é realizada por aplicação de velocidade de rotação superior à roda contrária da direção a tomar.

Devido a estas caraterísticas, é mais eficaz o envio de comandos ao controlo de tração por velocidades angulares das rodas esquerda e direita, ω_e e ω_d, do que comandos de velocidade linear e angular.

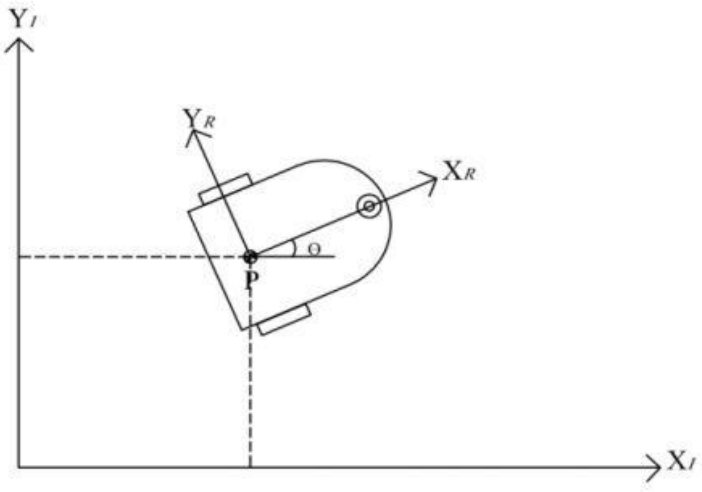

Figura 28- Representação do robô no sistema coordenadas cartesianas [28]

Através da velocidade de rotação das rodas (ω_e ou ω_d) e do seu raio (r) podemos obter a sua velocidade linear, $V_{e/d} = r\,\omega_{e/d}$. No caso de apenas a roda esquerda ter rotação, encontrando-se a roda direita parada, e devido ao ponto P se encontrar a meio do eixo das duas rodas, conforme ilustrado pela Figura 28, podemos verificar que o contributo da roda esquerda no deslocamento linear V_x do robô é $V_x = (1/2).V_e$. Devido a não existir deslocamento lateral nas rodas o valor de velocidade na direção V_y é sempre nulo. Assim, sendo d a distância entre rodas e R o raio de curvatura do robô temos:

$$V_e = \omega_e\,(R - \frac{d}{2}) \qquad V_d = \omega_d\,(R + \frac{d}{2}) \tag{11}$$

$$\omega_e = \frac{V_e}{R - \frac{d}{2}} \qquad \omega_d = \frac{V_d}{R + \frac{d}{2}} \tag{12}$$

No caso de robôs diferenciais estas contribuições de cada roda para o deslocamento podem ser somadas.

$$\omega = \frac{V_d - V_e}{d} \qquad V = \frac{V_d + V_e}{2} \tag{13}$$

4.1.2. Equações cinemáticas

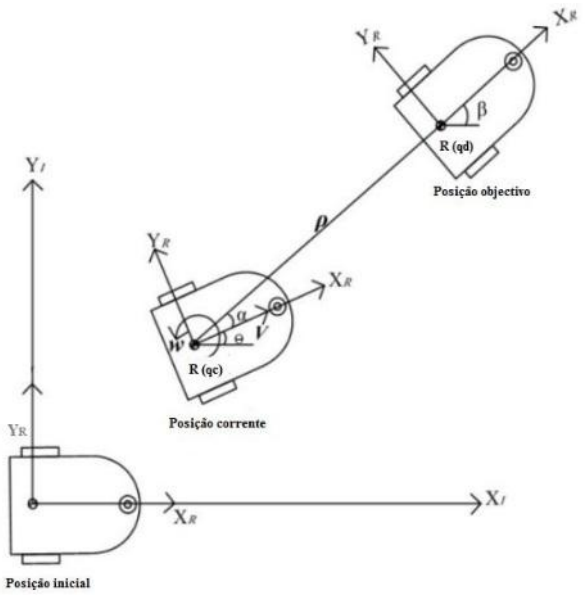

Figura 29- Posição e orientação do robô [28]

Considere que o robô se encontra na posição arbitrária P (q_c) [x_c y_c θ_c] encontrando-se o seu objetivo na posição R (q_d) [x_d y_d θ_d]. O seu sistema de coordenadas é gerido pela ação conjunta da velocidade linear v e da velocidade angular ω, podendo desta forma representá-las de forma cartesiana em relação à origem por

$$\dot{x} = v \cdot \cos \Theta \quad (14)$$

$$\dot{y} = v \cdot \operatorname{sen} \Theta \quad (15)$$

$$\dot{\theta} = \omega \quad (16)$$

Onde ẋ e ẏ são as componentes de v ao longo dos respetivos eixos e θ o seu ângulo.

4.1.3. Odometria

A localização é um dos aspetos mais importantes de um robô. É através do conhecimento da posição que o robô pode facilmente deslocar-se para o seu destino. Existem vários métodos para realizar a medição de deslocamento realizado pelo robô mas iremos descrever o método baseado na aquisição de impulsos dos *encoders* dos motores, que nos fornecem a rotação das rodas. A posição do robô é calculada com integração da rotação das rodas pela aproximação do seu modelo cinemático no intervalo $[t_k, t_{k+1}]$.

Assumindo o robô na posição q_k $[x_k\ y_k\ \theta_k]$ e as velocidades constantes de v_k e w_k são conhecidas em t_k, e sendo T_s o tempo de amostragem, então pela integração de Euler

$$x_{k+1} = x_k + v_k T_s \cos \theta_k \quad (17)$$

$$y_{k+1} = y_k + v_k T_s \operatorname{sen} \theta_k \quad (18)$$

$$\theta_{k+1} = \theta_k + w_k T_s \quad (19)$$

onde

$$v_k T_s = \Delta s \ \text{ e } \ w_k T_s = \Delta\theta \quad (20)$$

$$T_s = t_{k+1} - t_k \quad (21)$$

Sendo $\Delta\varphi_e$ e $\Delta\varphi_d$ o número de rotações das rodas esquerda e direita medidas num intervalo de tempo T_s, r o raio das rodas e d a distância axial entre rodas, obtemos assim o deslocamento linear e angular

$$\Delta s = \frac{r}{2}\,(\,\Delta\varphi_d + \Delta\varphi_e\,) \qquad \Delta\theta = \frac{r}{d}\,(\,\Delta\varphi_d - \Delta\varphi_e\,) \tag{22}$$

A posição de um robô diferencial pode ser estimada através da integração do seu movimento, podendo ser calculada para o tempo t_k

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \cos\theta & 0 \\ sen\,\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta\theta \end{bmatrix} \tag{23}$$

4.2. Procedimento de verificação e calibração UMBmark

Devido a erros existentes no modelo do robô, tais como diferença de diâmetro de rodas, desalinhamento de eixo ou mesmo capacidade de medição limitada, podem ser introduzidos erros no cálculo da odometria e respetiva pose do robô. Desta forma foi desenvolvido o método de verificação e calibração designado por UMBmark [29].

4.2.1. Procedimento de verificação

Este procedimento é realizado através de repetidos ensaios de execução de um percurso padrão previamente conhecido e aferido, por exemplo um quadrado de 4x4 m de comprimento, percorrendo-o nos dois sentidos de rotação possíveis (cw e ccw), a fim de verificar a ocorrência de desvios e respetiva quantificação.

Definindo os valores X_{abs} e Y_{abs} como as posições finais do ensaio com referência a X e Y, e definindo como X_{calc} e Y_{calc} os valores calculados na odometria, podemos deduzir os erros e_x e e_y existentes por:

$$e_x = X_{abs} - X_{calc} \tag{24}$$

$$e_y = Y_{abs} - Y_{calc} \quad (25)$$

Utiliza-se a média dos valores obtidos através de vários ensaios, $mX_{cw/ccw}$ e $mY_{cw/ccw}$, a fim de diminuir a dispersão de desvios provenientes de erros não-sistemáticos.

$$mX_{cw/ccw} = \frac{1}{n} \times \sum_{i=1}^{n} e_x\,(i) \quad (26)$$

$$mY_{cw/ccw} = \frac{1}{n} \times \sum_{i=1}^{n} e_y(i) \quad (27)$$

O resultado é obtido pelo cálculo da distância euclidiana $euc_{cw/ccw}$ dos pontos X e Y médios ($mX_{cw/ccw}$, $mY_{cw/ccw}$) conforme demonstrado na Figura 30.

$$euc_{cw} = \sqrt{(mX_{cw})^2 + (mY_{cw})^2} \quad (28)$$

$$euc_{ccw} = \sqrt{(mX_{ccw})^2 + (mY_{ccw})^2} \quad (29)$$

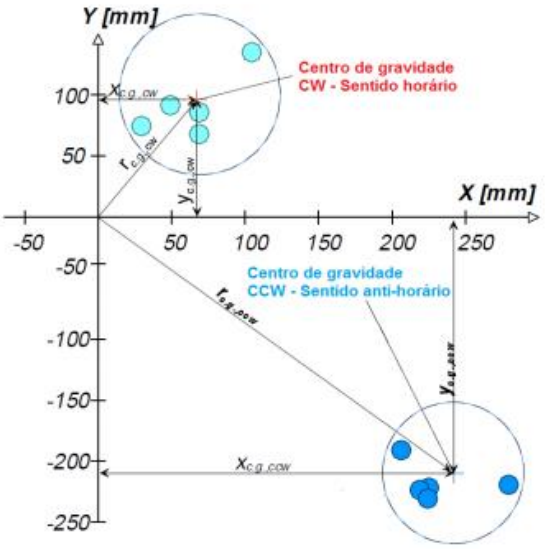

Figura 30- Dados adquirido no processo de calibração [29]

Sendo o maior erro e_{Max} selecionado para avaliação

$$e_{Max} = max(euc_{cw}, euc_{ccw}) \quad (30)$$

O erro obtido pode ter origem em dois tipos diferentes que passamos a descrever.

- Erro tipo A

Quando o valor de distância entre rodas é diferente do configurado para os cálculos de odometria, obtemos um erro de tipo A (ver Figura 31 a)). Este é o valor calculado para calcular os movimentos angulares do robô. Caso deste valor estar incorreto temos o robô não realiza a rotação desejada, causando assim um desvio na rotação sobre o seu eixo.

- Erro tipo B

Este tipo de erro é causado por diferenças nos diâmetros das rodas de tração, tal como pode ser observado na Figura 31 b). Este erro provoca um movimento não linear no robô causando um desvio à posição final desejada.

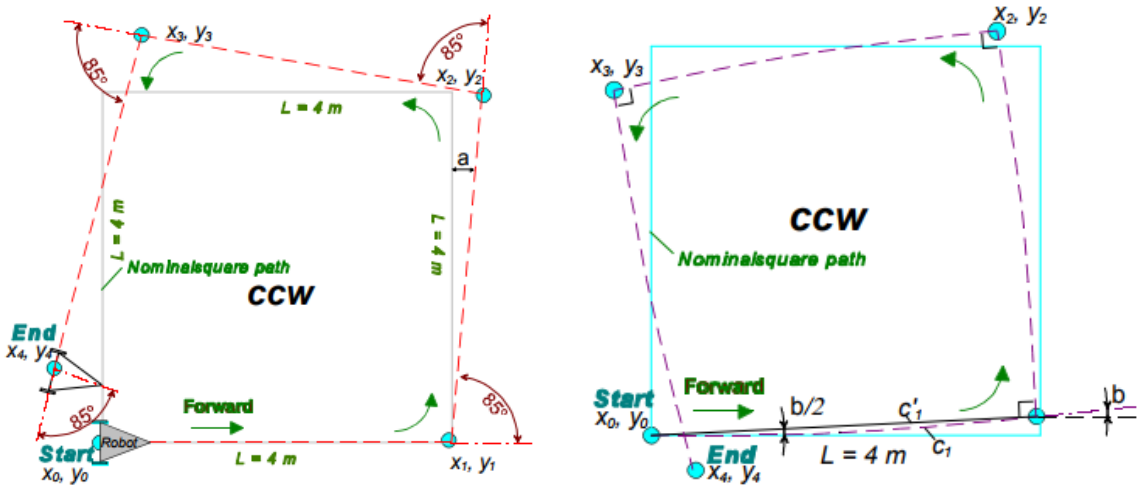

Figura 31- a) Erro característico Tipo A (d); b) Erro característico Tipo B(diâmetro de rodas) [29]

4.2.2. A calibração da odometria

Segundo o procedimento UMBmark, o cálculo de erros tipo A ou tipo B é feita separadamente, não existindo ordem para o cálculo dos mesmos.

- Erro tipo A

Através da relação que a distância entre rodas, d, é inversamente proporcional à quantidade de rotação realizada, apresentada por Borenstein e Feng [30], e sendo α o erro de rotação real da base temos:

$$\alpha = \frac{(mX_{cw} + mX_{ccw})}{-4L} \quad (31)$$

Sendo E_b o valor percentual de erro do valor d que deve ser corrigido para eliminar o erro verificado na Figura 31.

$$E_b = \frac{90}{90 - \alpha} \quad (32)$$

- Erro tipo B

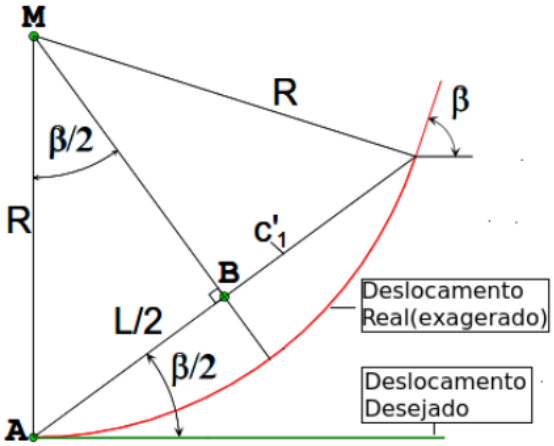

Figura 32- Relações geométricas de deslocamento [30]

Definimos β como ângulo de rotação devido à trajetória não linear (Figura 32). Podemos então calcular o termo corretivo para o erro tipo B como

$$\beta = \frac{(mX_{cw} - mX_{ccw})}{-4L} \quad (33)$$

O erro de diâmetro entre rodas vai causar transformar uma trajetória supostamente linear numa trajetória curvilínea de raio r

$$r = \frac{L/2}{sin(\beta/2)} \quad (34)$$

Podemos desta forma calcular também a relação do desvio, $E_{d,}$ associado ao erro Tipo B

$$E_d = \frac{r_e}{r_d} \times \frac{r - \mathrm{d}/2}{r + \mathrm{d}/2} \quad (35)$$

4.2.3. Procedimento de calibração conjunta

Existe também o procedimento de calibração conjunta, que, de acordo com o estudo realizado por [31], os erros devem ser analisados conjuntamente, obtendo assim os fatores de correção para os erros sistemáticos.

4.3. Navegação

Para maior versatilidade e possibilitando a navegação em vários tipos de ambiente, foi desenvolvido um sistema de comutação entre dois tipos de navegação da cadeira de rodas elétrica através de um interruptor acessível ao utilizador: “Navegação Livre” e “Navegação Assistida Anti-Colisão”.

4.3.1. Navegação Livre

Com a seleção deste modo de navegação, é dada ao utilizador total permissão de manobra da cadeira de rodas elétrica, inibindo assim o sistema de navegação assistida. Com esta funcionalidade, e em detrimento da segurança imposta pelo sistema ao utilizador, torna-se assim possível realizar a aproximação a espaços/locais mais confinados, tais como mesas ou passagens estreitas, sendo desta forma necessário alguma perícia do utilizador para evitar possíveis colisões com a cadeira.

4.3.2. Navegação com Sistema de Anti-Colisão

O Sistema de navegação assistida de anti-colisão reativa é baseado na perceção do meio ambiente circundante para análise de quais as velocidades permitidas na base robótica, para que a mesma se desloque em segurança, sem colisão com possíveis objetos, interpretando e controlando desta forma os comandos do utilizador.

A navegação da cadeira robotizada é realizada com base na informação recolhida pelo LRF instalado no suporte frontal da cadeira. Uma vez que estamos a realizar navegação reativa, é suficiente a leitura do estado do ambiente circundante atual para controlar a velocidade a aplicar na base robótica, não existindo a necessidade de manter um mapa.

Muitos dos métodos de navegação reativa assistida baseiam-se em encontrar a melhor trajetória no sentido de alcançar o destino final. Entre esses métodos destacam-se os métodos VFF[18], DWA[17]. No nosso projeto o utilizador tem domínio total sobre a direção a executar, devendo o sistema evitar a possível colisão com possíveis obstáculos apenas através do controlo da velocidade da CR, imobilizando a mesma a uma distância dos obstáculos considerada segura.

Devido a apenas possuirmos um laser como sensor da CR, este foi instalado numa posição central na zona frontal da CR, permitindo assim uma melhor perceção do espaço local, sendo necessário proceder à limitação e configuração do seu varrimento para 180°,

conforme exemplificado pela Figura 33, a fim de não existir interferências na perceção devido à posição de condução do utilizador.

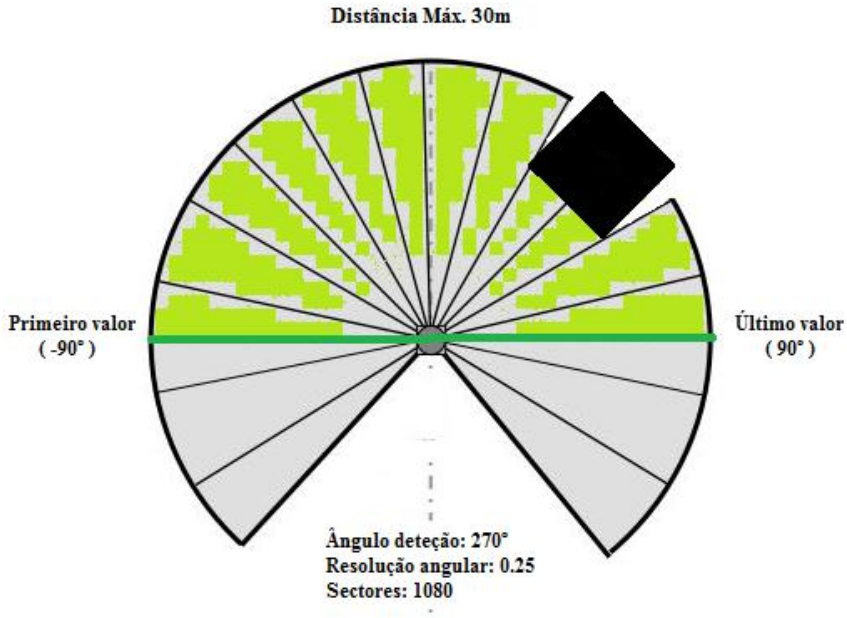

Figura 33- Varrimento do LRF Hokuyo UTM-30LX, características e configuração no sistema

Devido a esta configuração não é possível ter perceção do ambiente nos restantes 180° que circundam a CR, não estando assim garantida qualquer segurança em movimentos que desloquem a CR nesse espaço. A fim de evitar possíveis colisões com objetos que se encontrem na área de deslocamento angular apenas é permitido o recuo com velocidade linear, ficando o controlo da velocidade aplicada atribuído ao utilizador.

Para implementar esta navegação optámos por desenvolver um método de análise das distâncias recolhidas através do laser, subdividindo-as em vários sectores, podendo-se assim classificar as distâncias aos objetos existentes, através da avaliação do risco envolvido na distância aos mesmos. É deste modo possível atuar sobre as velocidades lineares e angulares solicitadas pelo utilizador e permitindo uma navegação segura sem colisão em conformidade com a análise realizada.

4.3.3. Projeção do Scanner

Devido à posição do sensor laser se encontrar deslocada do centro de rotação do nosso veículo, é necessário proceder à transformada das coordenadas polares recolhidas para a localização do eixo de rotação, podendo assim avaliar corretamente as distâncias e velocidades possíveis para a navegação segura da CR.

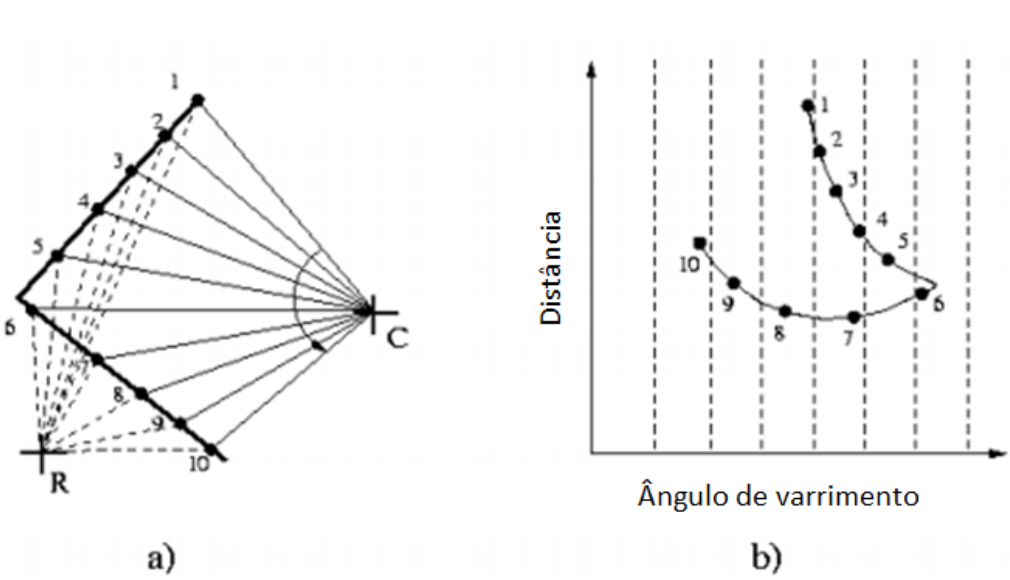

Figura 34 - a) Projeção dos pontos obtidos em C para R. b) Pontos projetados em R em coordenadas polares

Na Figura 34 podemos verificar um scan realizado na posição C, sendo depois tomada a posição R como referencia. Sendo o corrente scan descrito por $C = (\mathrm{Xc}, \mathrm{Yc}, \theta\mathrm{c}, \{\mathrm{r_{ci}}, \emptyset_{ci}\}_{i=1}^{n})$ onde $\mathrm{X_c}$, $\mathrm{Y_c}$, θ_c descreve a sua posição e orientação e $\{\mathrm{r_{ci}}, \emptyset_{ci}\}_{i=1}^{n}$ descreve os n valores de distância $\mathrm{r_{ci}}$ nos deslocamentos anelares de $\emptyset_{ci}$ (n de 1 a 10 conforme exemplo da Figura 34 a)), os valores r'_{ci} e ângulos $\emptyset'_{ci}$ relativamente ao ponto R da Figura 34 b) podem ser calculados através das seguintes fórmulas:

$$r'_{ci} = \sqrt{(r_{ci}\cos(\theta_c + \emptyset_{ci}) + x_c)^2 + (r_{ci}\sin(\theta_c + \emptyset_{ci}) + y_c)^2} \qquad (36)$$

$$\emptyset'_{ci} = atan2((r_{ci}\sin(\theta_c + \emptyset_{ci}) + y_c\,, (r_{ci}\cos(\theta_c + \emptyset_{ci}) + x_c) \qquad (37)$$

Na Figura 34 b) as linhas tracejadas verticais representam as amostras dos ângulos anelares do laser na posição R.

4.3.4. Segurança Ativa baseada na análise de risco

Através das configurações, quer geométricas, quer dinâmicas, do sistema, podemos definir três zonas distintas de risco associado à navegação conforme demonstrado na Figura 35.

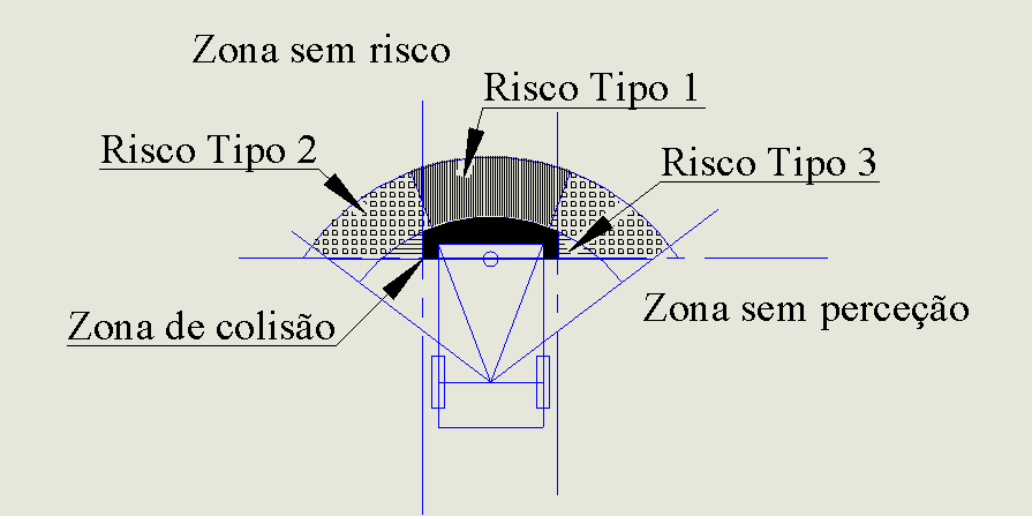

Figura 35- Identificação das diferentes zonas de possível colisão

Sem Risco – distância dos possíveis objetos existentes é superior á distância máxima necessária para imobilização da CR

Com Risco Tipo 1, 2 e 3 (linhas verticais, quadrados, linhas horizontais) – Existe a possibilidade de colisão com objetos na zona de navegação da CR, sendo desta forma necessário analisar possíveis velocidades solicitadas para verificar a sua viabilidade de execução ou limitação (zonas sombreadas da figura).

Zona de colisão (zona preta) – A distância aos objetos detetados não permite movimento na direção desejada.

Deadlock – A fim de permitir a saída da cadeira de possíveis situações de bloqueio, é sempre possível a aplicação de movimento linear à retaguarda, evitando assim a possibilidade de colisão lateral com possíveis objetos na "zona morta".

Os três tipos de zonas de risco estão divididos segundo os seguintes critérios:

- Tipo 1: Obstáculo situado na zona frontal da CR
- Tipo 2: Obstáculo situado na zona lateral da CR fora da distância limite de colisão
- Tipo 3: Obstáculo situado na zona lateral da CR dentro da distância limite de colisão

Todas as zonas referentes aos diferentes tipos de risco são calculadas de acordo com as definições físicas, dinâmicas e limitações do sistema

- Largura da cadeira – Física e margem de segurança
- Distância mínima frontal ao objeto (limite de colisão)
- Velocidade Máxima Linear
- Velocidade Máxima Angular
- Desaceleração Linear

Através das duas primeiras definições construímos o ângulo de análise frontal e a zona de colisão, Através das definições dinâmicas calculamos os ângulos de varrimento lateral e a distância máxima de paragem, definindo assim todas as zonas de risco.

4.3.5. Dinâmica de navegação

Para calcular as velocidades admissíveis para progressão da CR, aplicamos um método inspirado no método de "Dynamic Window Approach", isto é, através da análise das velocidades permitidas para navegação sem colisão.

Sabendo que a distância de paragem (d_p) pode ser calculada através da distância de "livre deslocamento" (d_l) e da distância de travagem (d_t) pela fórmula

$$d_p = d_l + d_t \quad (38)$$

E podendo desprezar a d_l, devido ao nosso período de ciclo ser aproximadamente 20 ms, podemos concluir pela equação de Torricelli [32]

$$d_p = d_t = \frac{{V_{Max}}^2}{2a} \quad (39)$$

Fazendo a analogia com o conceito aplicado no método da "Dynamic Window Approach", podemos validar se as velocidades pretendidas em função das distâncias verificadas aos objetos.

$$V \leq \sqrt{2.dist(v,w).\dot{v}} \quad (40)$$

$$W \leq \sqrt{2.dist(v,w).\dot{w}} \quad (41)$$

Desta forma podemos verificar se a velocidade linear e angular solicitadas pelo utilizador são válidas, ou seja, caso essa velocidade seja aplicada na CR, a mesma não colide com qualquer objeto existente no campo de perceção do sensor.

A fim de otimizar o algoritmo de análise do ambiente envolvente verificamos se existe componente angular na velocidade desejada pelo utilizador, para que, caso essa velocidade exista, a análise compreenda a área frontal e a área lateral do quadrante correspondente, caso contrário, apenas é analisada a área frontal da CR.

No caso de existir possibilidade de colisão com um obstáculo, sabemos que a possibilidade de colisão ocorre com o obstáculo que se encontrar mais perto, pelo que adotámos as seguintes premissas no nosso algoritmo de acordo com a perceção do laser e as velocidades solicitadas pelo utilizador.

Risco Tipo 1

$W = 0$ apenas é analisada a área frontal da CR, sendo a velocidade limitada à velocidade de segurança, permitindo o utilizador realizar uma aproximação ao objeto detetado.

$W \neq 0$ é analisada a área frontal e a área lateral da respetiva direção selecionada pelo utilizador, para verificação de existência de Risco Tipo 2 ou Tipo 3.

Risco Tipo 2

São verificadas as velocidades solicitadas, após limitação se existir risco Tipo 1, para aproximação frontal ao objeto. Isto significa que a velocidade linear terá de ser inferior à velocidade de segurança, sem aplicação de restrições para a velocidade angular. Caso as velocidades solicitadas não sejam válidas será considerada paragem da CR.

Risco Tipo 3

Para o Risco Tipo 3, o objeto encontra-se dentro do perímetro de rotação e colisão da CR, mas devido a encontrar-se a uma distância lateral superior á definida como segura, é possível prosseguir com velocidade linear, limitada ou não por outros tipos de risco. Porém a velocidade angular é limitada para impedir possível rotação que possa originar uma colisão com um obstáculo.

5. Integração do sistema na plataforma ROS

A implementação do sistema na plataforma ROS foi estruturada de forma a criar módulos específicos para o processamento de cada tarefa, tendo sido desenvolvida a estrutura que se apresenta no capítulo 5.1

5.1. Arquitetura do software na plataforma ROS

No desenvolvimento da aplicação baseada em ROS, foi tido em conta a possível integração de outros nós *software*/*hardware* de controlo/comando a integrar futuramente este projeto. Desta forma foi implementada a arquitetura de *software* baseada em grafos definida na Figura 36, identificando os tópicos existentes, assim como os módulos, a tracejado, que poderão ser implementados futuramente, quer de comando de velocidade como de controlo de proximidade, permitindo assim tornar a cadeira funcional para outras aplicações.

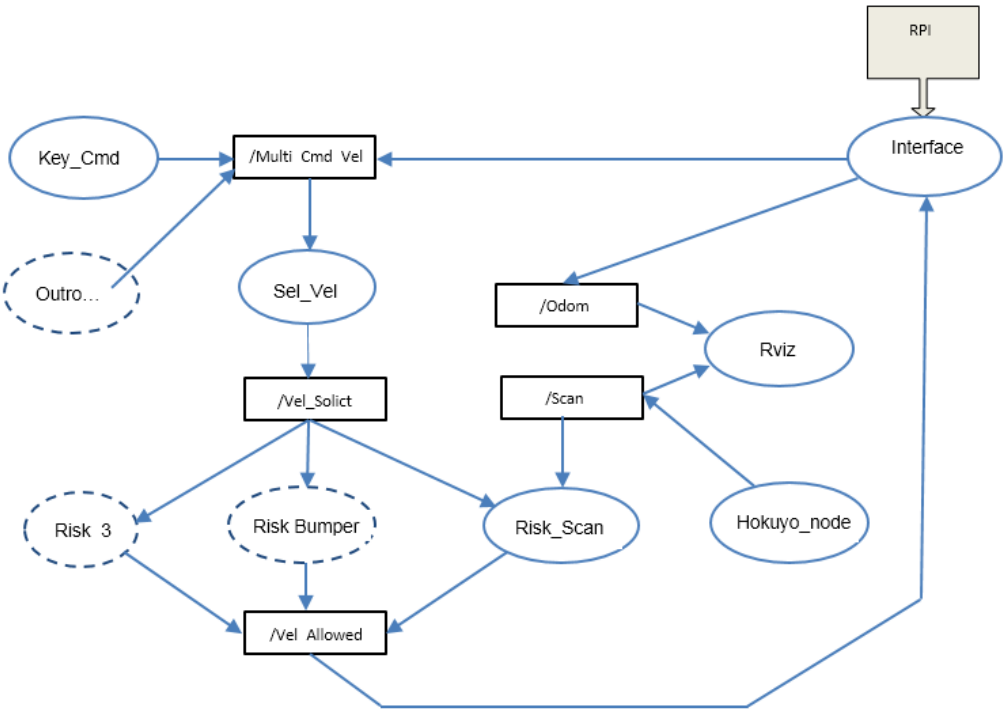

Figura 36- Grafo de nós e tópicos do ROS implementado

Na Figura 36 estão identificados os vários nós e os tópicos existentes no sistema. É através da partilha das várias mensagens que é possível ter vários nós a tratarem simultaneamente da mesma informação para fins comuns ou mesmo independentes.

O nó "Interface" é o responsável pela troca de informação entre o RPi e o Sistema ROS. Esta comunicação é realizada através de ligação Ethernet, configurada para a rede 192.168.1.0. Através desta comunicação são recebidos os dados do comando do *Joystick* e da odometria da CR, sendo estes últimos dados processados no RPi. É também este módulo que envia o comando de velocidade de volta para o RPi a fim de ser executado pelo controlador MDC3220 no caso de estar selecionado o modo de Navegação com Sistema Anti Colisão.

Através do tópico fornecido pelo nó do laser "Hokuyo_node" podemos analisar as distâncias recolhidas, e de acordo com as velocidades solicitadas implementar o algoritmo de navegação, processando os dados no nó "Risk_Scan" e publicando o tópico de velocidade permitida, a fim de esta ser consultada pelo módulo de "Interface" devolvida ao RPi para posterior envio ao controlador de potência.

O nó "Key_Cmd" foi introduzido apenas para demonstrar a simplicidade de integração de novos sistemas, sem grande alteração da estrutura ou código existentes, permitindo assim atualizações/modificações de nós sem necessidade de alterações de código.

Como não estamos a realizar nem a utilizar mapeamento, e apesar do seu desenvolvimento, o tópico da odometria não é relevante para o funcionamento do nosso algoritmo porque não necessitamos de conhecer a localização do robô no mundo. No entanto a odometria está disponível para possíveis verificações e interações com o nó RVIZ e para desenvolvimentos futuros.

O nó Risk_Scan é o responsável pela avaliação da perceção local obtida com base nos dados fornecidos pelo sensor laser, encontrando-se este módulo descrito no capítulo 5.2.

5.2. Análise de Risco

.

Com base nos parâmetros apurados no decorrer dos ensaios experimentais para o sistema foram calculados os ângulos e distâncias apresentadas na .

Variável	Valor	
Diâmetro (m)	0.3556	
Dist. entre Rodas (m)	0.65	
V Máx (Km/h)	2	
Relação Vmáx / Wmáx	0.5	
Dist. Colisão (m)	1.15	
largura da CR (m)	0.7	
V Máx (m/s)	0.555555556	
W Máx (m/s)	0.854700855	
Acel V Máx (m/s^2)	0.358061009	
Acel W Máx (m/s^2)	0.550863091	
Dist. Min. Seg.(m)	1.580990764	
ang[0] (rad / °)	-0.75129551	-43.046062
ang[1] (rad / °)	-0.309253701	-17.718932
ang[2] (rad / °)	0.309253701	17.7189319
ang[3] (rad / °)	0.75129551	43.0460619

Tabela 3- Parâmetros para a configuração de zonas de risco

Para realizar a análise de risco de navegação é necessário proceder à definição dos limites das zonas de risco identificadas na Figura 35

Através dos parâmetros de largura da CR com margem de segurança pretendida e da distância mínima a partir da qual se considera zona de colisão, definimos o ângulo do setor frontal (linhas 2 e 3 da Figura 37) e o arco de distância de colisão (arco 1). Através dos valores de aceleração e velocidade linear máxima calculamos o limite máximo da zona de risco (arco 4), sendo o modelo completado através da aplicação dos valores de velocidade e aceleração angular máxima no cálculo dos ângulos de varrimento lateral (linhas 5 e 6). Podemos verificar a construção dos passos identificados através da Figura 37.

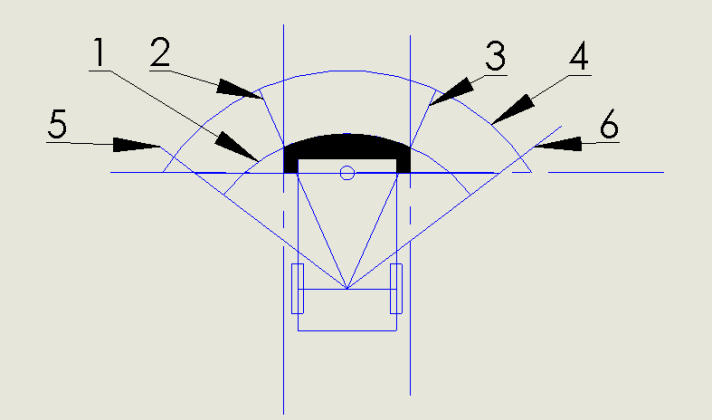

Figura 37- Definição de limites de zonas de risco

Com a realização da leitura e análise das mensagens publicadas pelo LRF, este nó verifica o valor de maior risco para vários sectores definidos aplicando a seguinte sequência de análise de risco de acordo com a definição no capítulo 4.3.5:

- Sector 1 – Análise Risco Tipo 1
- Sector 2 e 4 - Análise Risco Tipo 2
- Sector 3 e 5 – Análise Risco Tipo 3

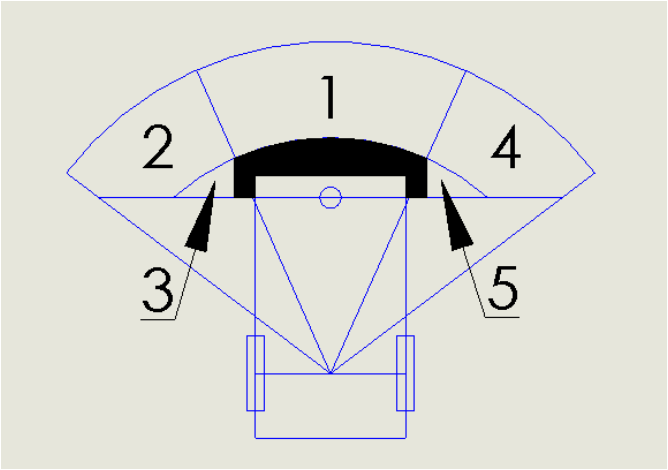

Figura 38- Identificação dos diferentes sectores para análise de risco

Após a análise de risco dos vários sectores identificados na Figura 38, este módulo publica as velocidades linear e angular a serem enviadas para o nó de interface para posterior tratamento.

5.3. Fluxograma da Análise de Risco

Sendo Vi e Wi as velocidades recebidas através do tópico /Vel_Solicit, Vc e Wc as velocidades máximas calculadas para cada tipo de risco e Vp e Wp as velocidades permitidas ao longo da análise, foi implementado o fluxograma apresentado na Figura 39. Após conclusão da análise, a velocidade permitida é enviada ao sistema pelo tópico /Vel_Allowed.

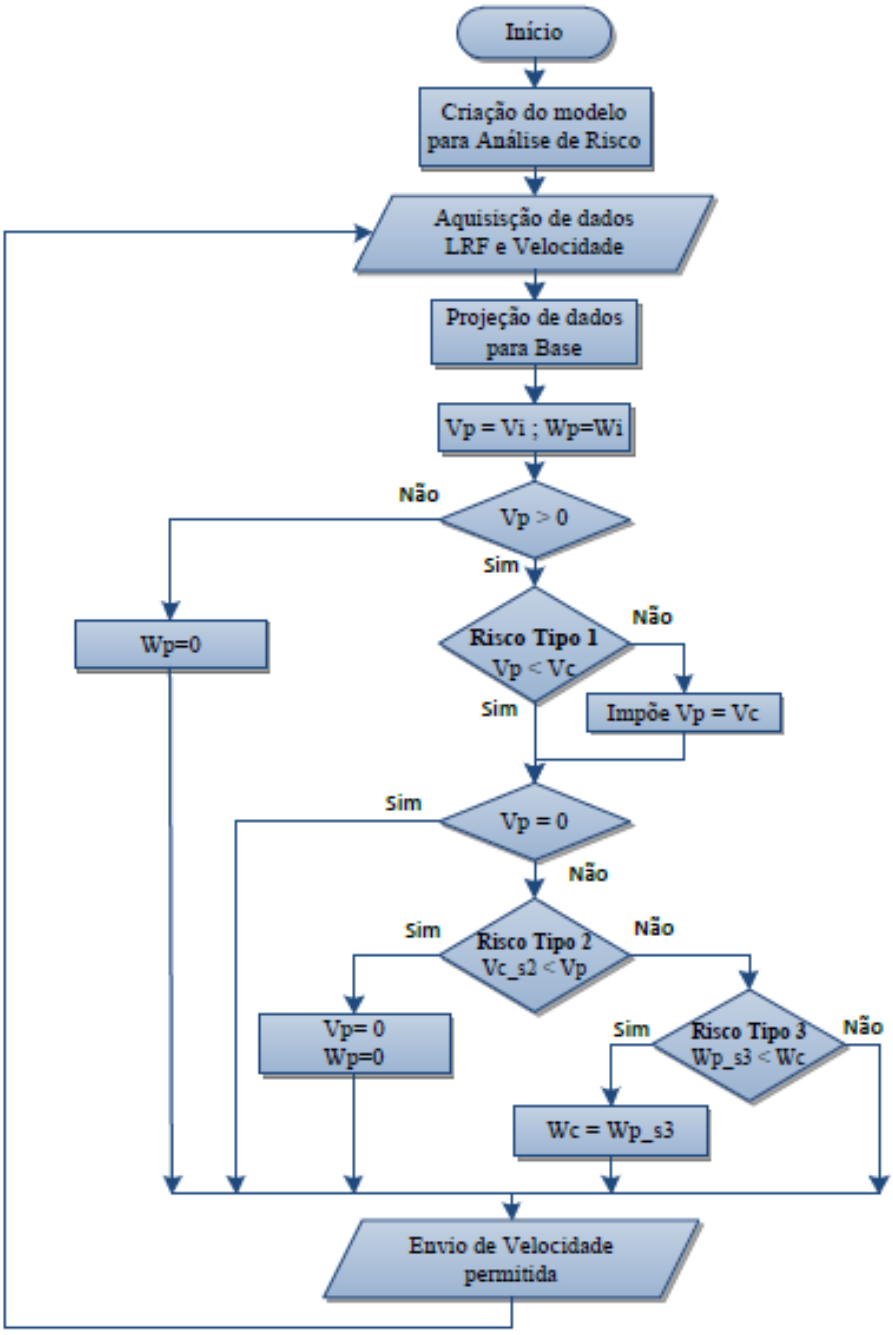

Figura 39- Fluxograma da análise de risco

6. Ensaios e resultados experimentais

Neste capítulo encontram-se descritos os ensaios realizados a fim de verificar e validar os métodos estudados e implementados. Os testes iniciais incidiram sobre a implementação do sistema de tração da cadeira de rodas sem qualquer tipo de assistência na navegação para garantir a estabilidade da integração dos módulos de interface e tração. Posteriormente foram realizados testes do sistema de navegação assistida.

6.1. Ensaios sistema de comando e tração

Conforme já descrito no capítulo 3, foram realizados ensaios experimentais aos vários módulos desenvolvidos no decorrer da robotização da cadeira para garantir a sua correta integração no projeto, tais como:

- Integração de um *joystick* analógico, sendo aplicado um conversor A/D para interligação com o RPi. Para tratamento dos dados foi desenvolvida uma biblioteca específica para este controlador, tendo sido realizados ensaios para verificar a funcionalidade e linearidade dos valores recebidos.

- Integração de *encoders* – Foram ensaiados os sentidos de rotação e valores dos *encoders* de ambos os motores, assim como a sua leitura pelo controlador de potência, garantindo assim o correto controlo sobre o sistema de tração da CR.

- Comunicação série – foi desenvolvida e ensaiada uma classe específica para comunicação entre o RPi e o controlador Roboteq. Foram ensaiados e validados envios de comandos de velocidade e receção de resposta sobre taquimetria, garantindo assim a correta comunicação entre ambos os equipamentos. Foi também ensaiada a funcionalidade de segurança de *Watchdog* existente no Roboteq que imobiliza a CR em caso de perda de comunicação no tempo programado.

- Comunicação Ethernet – Foi configurada a rede 192.169.1.0 para interligação entre o RPi e o PC com o ROS. Foram definidos endereços estáticos, 192.168.1.2 e 192.168.1.5 para o PC e para o RPi respetivamente, a fim de evitar configurações dinâmicas no decorrer da inicialização do sistema. Devido à necessidade de mobilidade do sistema foi também

necessário utilizar o *software* Putty para acedermos via SSH à consola do RPi, podendo assim visualizar, programar e executar o código necessário.

6.2. Ensaios dinâmicos de tração

Os testes dinâmicos iniciais permitiram verificar a manobrabilidade da cadeira e a sua resposta aos comandos do interface. Foram ensaiados vários valores para a velocidade linear e angular a fim de garantir uma correta perceção da velocidade solicitada. Também foram selecionados vários valores para a velocidade máxima a fim de garantir uma navegação suave e contínua em espaços confinados como por exemplo o laboratório VITA.IPT onde foram realizados estes ensaios.

Após realizados os referidos ensaios práticos configurámos o controlador com os valores de velocidades, acelerações e valores do controlador PID verificados na Figura 40.

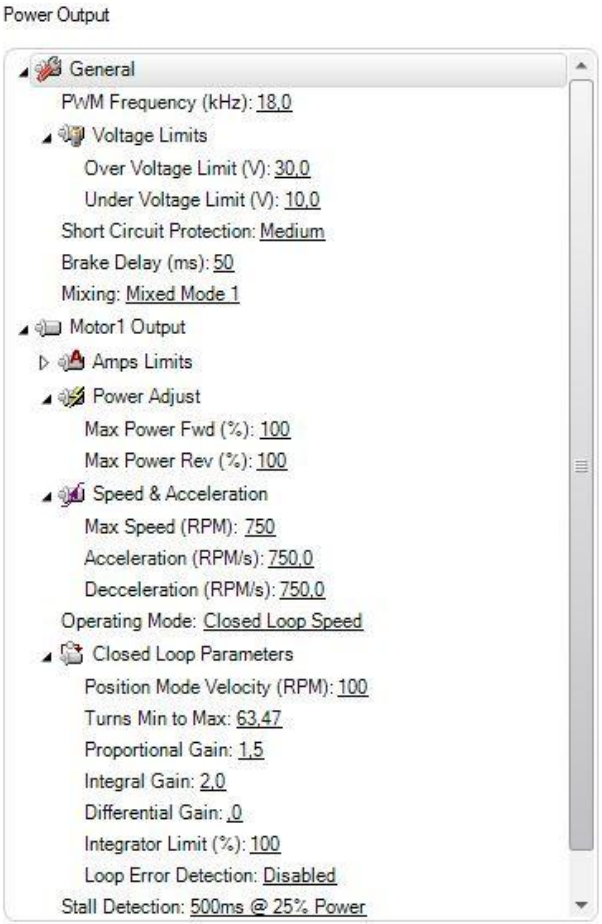

Figura 40- Configuração das saídas de potência Roboteq

Para validação da odometria, realizaram-se deslocamentos da cadeira de num percurso quadrado de 3 metros de lado a fim de verificar os desvios ocorridos entre os valores calculados e reais.

Para minimização de erros, foram realizados 5 percursos no sentido dos ponteiros de relógio (CW) e no sentido contrário (CCW). Podemos verificar os resultados obtidos na seguinte tabela e a respetiva dispersão no gráfico apresentado na Figura 41.

Ensaio	X (m)	Y (m)
ccw1	0,109813	0,157246
ccw2	0,409585	0.2748.26
ccw3	0,36506	0,26523
ccw4	0,368739	0,210552
ccw5	0,134854	0,208333
cw1	-0,05866	0,117376
cw2	-0,11994	0,148398
cw3	-0,02511	0,120827
cw4	-0,16618	0,115481
cw5	-0,18474	0,333718
Média ccw	0,27761	0,168272
Média cw	-0,11093	0,16716

Tabela 4- Resultados práticos método UMBmark

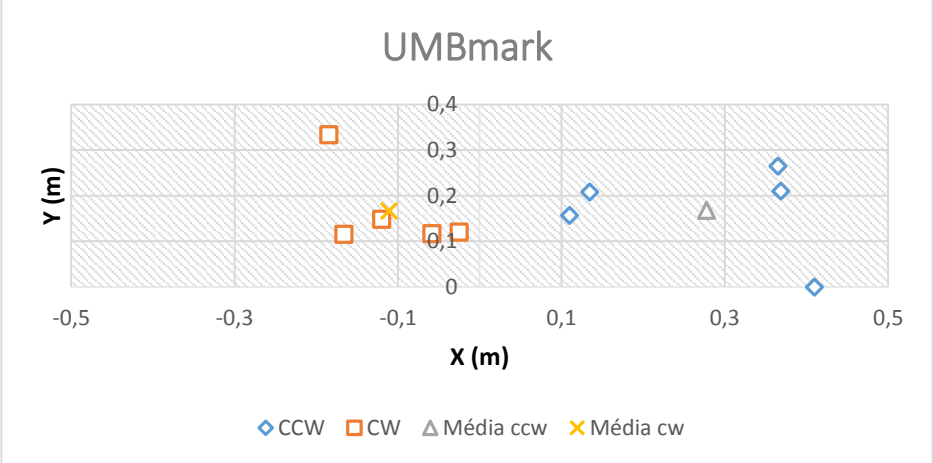

Figura 41- Gráfico de dispersão dos resultados do método UMBmark

Através da aplicação do método de verificação UMBmark descrito no capítulo 4.2.1 obtemos assim $e_{Max} = 0{,}324627$. Aplicando o seu o procedimento de calibração descrito no capítulo 4.2.2, obtemos os seguintes fatores de correção de erro:

$$\alpha = 0.01389 \qquad \beta = 0.0000695 \qquad R = 2473199 \tag{42}$$

$$Eb = 0{,}99846 \qquad Ed = 0{,}999999 \approx 1 \tag{43}$$

Após correção do valor da distância entre rodas através do fator de correção E_b foram realizados os mesmos ensaios onde foram obtidos os seguintes resultados

Ensaio	X (m)	Y (m)
ccw1	0,128355	0,137577
ccw2	0,078022	0,129848
ccw3	-0,02194	0,130897
cw1	-0,14608	0,182646
cw2	-0,05945	0,07937
cw3	-0,07637	-0,06149
Média ccw	0,036888	0,079664
Média cw	-0,05638	0,040105

Tabela 5- Resultados práticos método UMBmark após fator correção

Verificámos assim que o erro máximo diminuiu para $e_{Max} = 0{,}069189$. Podemos constatar experimentalmente que, dependendo da posição em que se encontram e devido ao seu tamanho e constituição, as rodas livres podem exercer grande oposição ao movimento desejado pelo utilizador, criado assim escorregamento das rodas de tração e influenciando assim os cálculos de odometria.

Foram gravados os dados de alguns ensaios de odometria através do comando rosbag, comando da plataforma ROS que regista as mensagens dos tópicos selecionados pelo utilizador, tendo os percursos registados sido representados através do Matlab e integrados na planta do piso térreo do Bloco I do IPT conforme ilustrado na Figura 42.

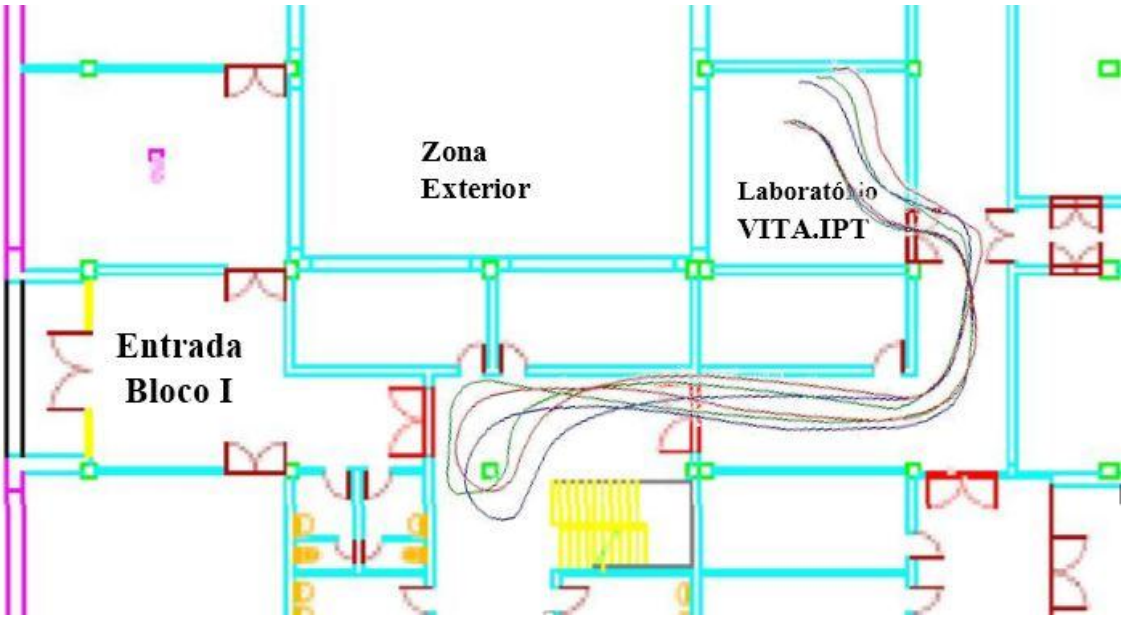

Figura 42- Planta do Bloco I do IPT com percursos de ensaios de odometrias

6.3. Ensaios do sistema de navegação

Para iniciar os ensaios de navegação construímos alguns cenários com a colocação de vários objetos na zona de perceção do LRF de forma a validar os dados recolhidos do *software*, podendo verificar na Figura 43 o mapeamento local obtido no Rviz e na Figura 44 os gráficos construídos com a informação recolhida pelo LRF e a sua projeção.

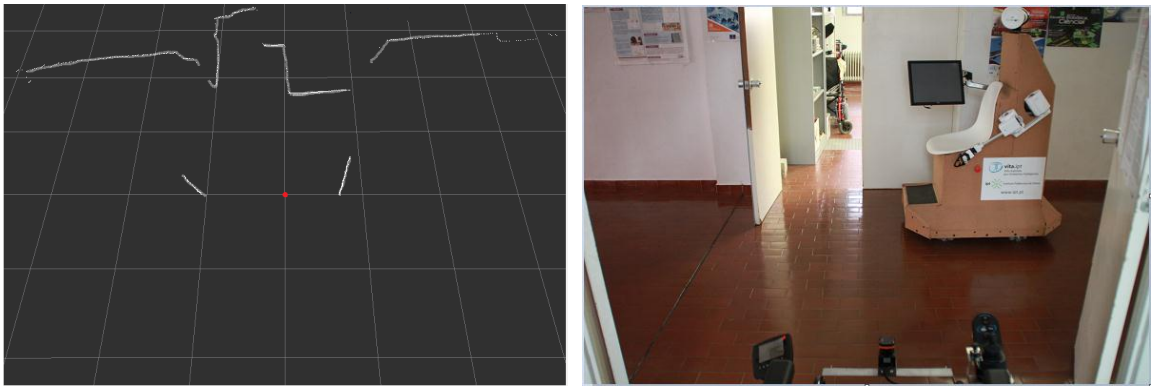

Figura 43- Visualização de dados do Scanner com Software RVIZ e sua imagem

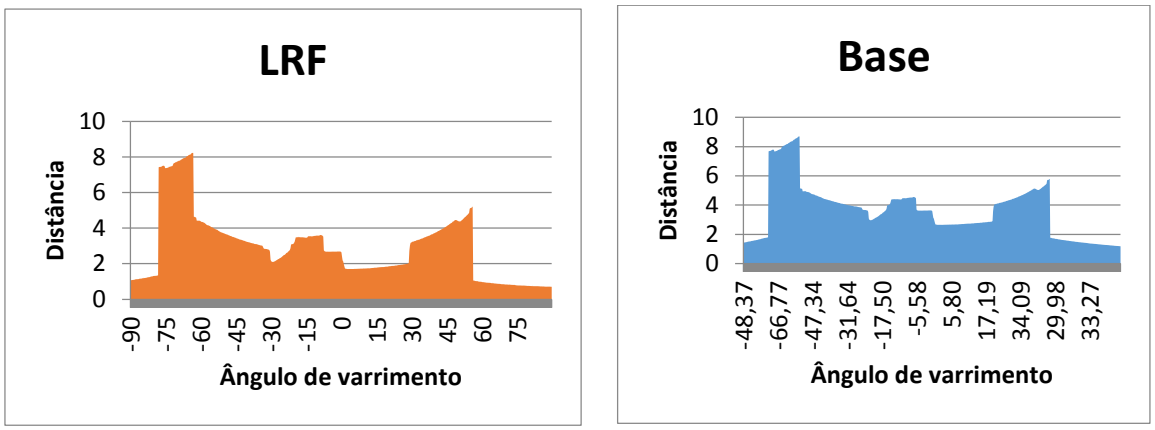

Figura 44- Gráficos com representação das distâncias referenciadas ao laser e ao centro de rotação

A validação dos sectores de análise de risco foi realizada com ensaios práticos através da disposição de objetos em zonas específicas, verificando e avaliando os resultados obtidos apresentados na Tabela 6.

Ensaio	Velocidade solicitada		Setor 1	setor 2	Setor 3	Setor 4	setor 5	Velocidade permitida	
	Linear	Angular	Distância	Distância	Arco	Distância	Arco	Linear	Angular
1	0,554	0	1,282	1,454	0,763	1,753	0,763	0,308	0
2	0,554	0,385	2,359	1,448	0,763	1,318	0,763	0	0
3	0,155	0,247	2,360	1,458	0,763	1,316	0,763	0,155	0,247
4	0	0,426	2,345	1,454	0,763	2,869	0,089	0	0,217
5	0	-0,426	2,350	1,163	0,003	1,750	0,763	0	0
6	0,129	-0,164	1,528	1,452	0,763	1,756	0,763	0,129	-0,164
7	-0,554	-0,427	1,531	1,45	0,763	1,759	0,763	-0,554	0

Tabela 6- Valores experimentais de distâncias e ângulos dos vários sectores

Os valores assinalados a amarelo são os responsáveis pela limitação ou inibição da velocidade.

Conforme exemplificado no ensaio Nº 1, é solicitada uma velocidade linear de 0,554 m/s com velocidade angular nula. Devido à existência de um objeto a uma distância de

1,282m a velocidade máxima para a cadeira é limitada a 0,308 m/s, garantindo assim uma navegação segura ao utilizador. No caso do exemplo Nº 7, ao ser solicitada uma velocidade negativa, é anulada a velocidade angular para evitar colisões laterais em possíveis objetos na zona sem perceção.

Foram realizados vários ensaios dinâmicos para validar o método NAAC (Navegação Assistida Anti-Colisão). No percurso realizado foram colocados vários tal como exemplificado na Figura 45, validando a deteção dos obstáculos e respetiva reação aos comandos do utilizador através da atuação do sistema anti-colisão. É também possível verificar na Figura 45 a indicação da velocidade pretendida pelo utilizador e a permitida pelo sistema NAAC.

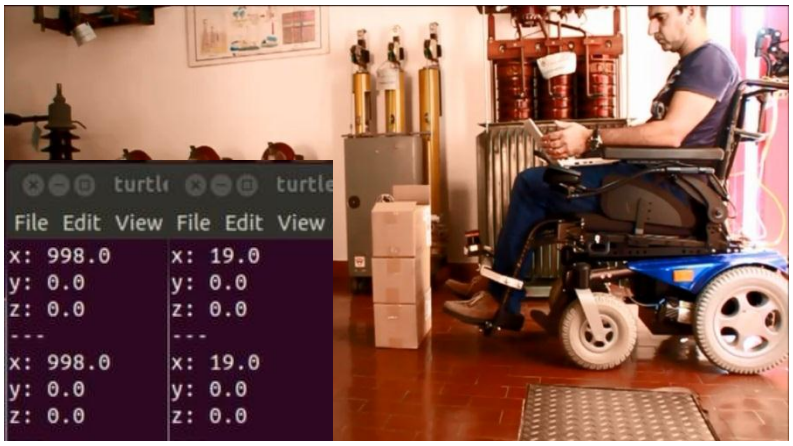

Figura 45- Ensaios dinâmicos de deteção de obstáculos com indicação de velocidade pretendida e permitida

Foi solicitada a participação de alguns voluntários para experimentarem e avaliarem a prestação do sistema implementado tendo em conta o interface, a manobrabilidade, a apresentação geral da implementação do sistema na cadeira e propostas de melhoria. Estes ensaios foram realizados por 6 utilizadores do sexo masculino com idades compreendida entre os 20 e os 32 anos, não tendo qualquer um dos utilizadores experiencia prévia em condução de cadeiras de rodas.

O ensaio dinâmico realizado pelos utilizadores voluntários consistiu na condução da cadeira de rodas com o sistema NAAC, tendo como ponto inicial o interior do laboratório VITA.IPT, conforme ilustrado pela Figura 46, saindo para os corredores do pavilhão através

da porta com uma abertura de 90 cm, percorrendo depois um percurso livre e retornando ao ponto de origem com mesma localização e orientação.

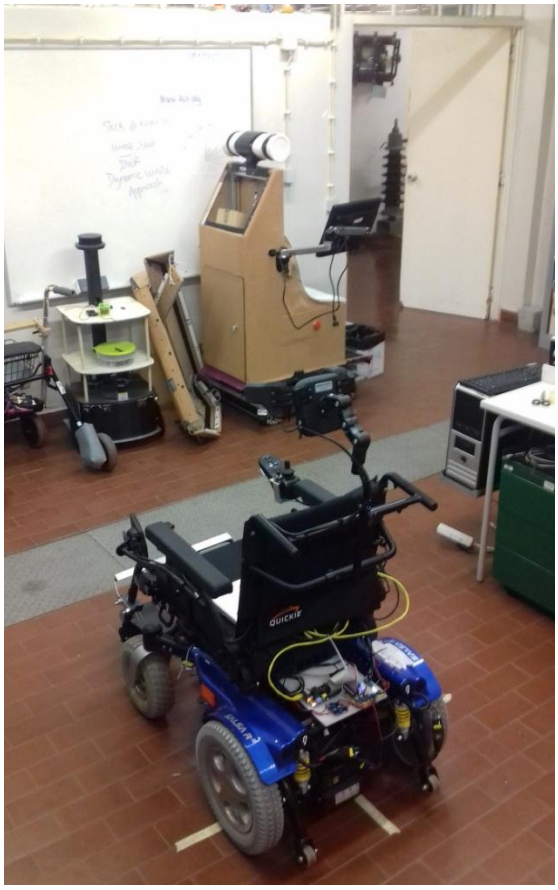

Figura 46- Posição inicial e final de ensaios experimentais por utilizadores voluntários

Do inquérito realizado concluímos as seguintes avaliações sobre o sistema implementado:

- Navegação - Ocorreu por três vezes a possibilidade de colisão em objetos situados na zona morta do sistema aquando da passagem da cadeira pela zona da porta do laboratório, devido a falta de experiência dos utilizadores na zona de perceção do sistema.
- Interface – Todos os utilizadores indicaram a fraca sensibilidade do *joystick*, devido ao seu pequeno curso de manobra, causando alguma irregularidade na navegação, mais acentuada a utilizadores sem experiencia pática de utilização deste elemento, encontrando-se este ponto referido nas propostas de melhoria.

- Manobrabilidade – Os utilizadores avaliaram como boa a manobrabilidade da cadeira, permitindo facilmente manobrar a mesma em espaços limitados como o interior do laboratório.
- Montagem geral – relativamente ao aspeto geral da montagem do sistema na cadeira de rodas os utilizadores também o classificaram de bem concebido e estruturado, sendo apenas referida a dificuldade de entrada na cadeira devido ao suporte do laser se encontrar montado acima dos apoios de pés da mesma, tendo sido referido que existe a hipótese de desmontagem e montagem da barra de suporte para a entrada e saída de utilizadores.
- Pontos de melhoria – neste tópico foi referido pelos utilizadores a possibilidade de integração de mais sensores para anular a existência da zona morta lateral.

7. Conclusões e trabalho futuro

Através do desenvolvimento desta tese foi realizada a conversão de uma cadeira de rodas elétrica para uma cadeira de rodas robotizada, tendo sido implementado um sistema de controlo que nos permite controlar a velocidade de deslocamento da cadeira assim como obter a sua posição estimada, cumprindo assim os objetivos inicialmente propostos para este desenvolvimento, podendo este ser aplicado a futuros projetos.

Para melhorar a prestação da cadeira perante a sua utilização por pessoas com alguma limitação física/cognitiva de perceção e/ou reação ao meio ambiente circundante foi desenvolvido e implementado um sistema de Navegação Assistida Anti-Colisão (NAAC) o qual teve as seguintes contribuições:

- Implementação de perceção local através de um LRF
- Desenvolvimento de *software* de controlo na plataforma ROS
- Implementação de um modo assistido de navegação anti-colisão

Como propostas para trabalho futuro propomos os seguintes melhoramentos e desenvolvimentos:

- Substituição por um *joystick* mais ergonómico e que permita um controlo mais preciso dos comandos a enviara ao controlador.
- Substituição das rodas livres a fim de reduzir o atrito e obstrução criado pelas mesmas em situações pontuais da navegação, criando desta forma possíveis erros de odometria.
- A utilização desta cadeira robotizada em aplicações baseadas em outras interfaces com o utilizador

ANEXOS

I. Cálculo dos parâmetros para controlo PID

Para o controlo de velocidade nos motores da cadeira de rodas optámos por aplicar um controlador PID, usando o método de aplicar um degrau na entrada em malha aberta e verificar o comportamento dos motores. Através deste método já estão considerados os atritos existentes na resposta do motor.

a. Função de transferência

Para a obtenção da função de transferência aplicámos um degrau de 50% do valor nominal de alimentação dos motores e registámos a sua resposta conforme representado na Figura 47.

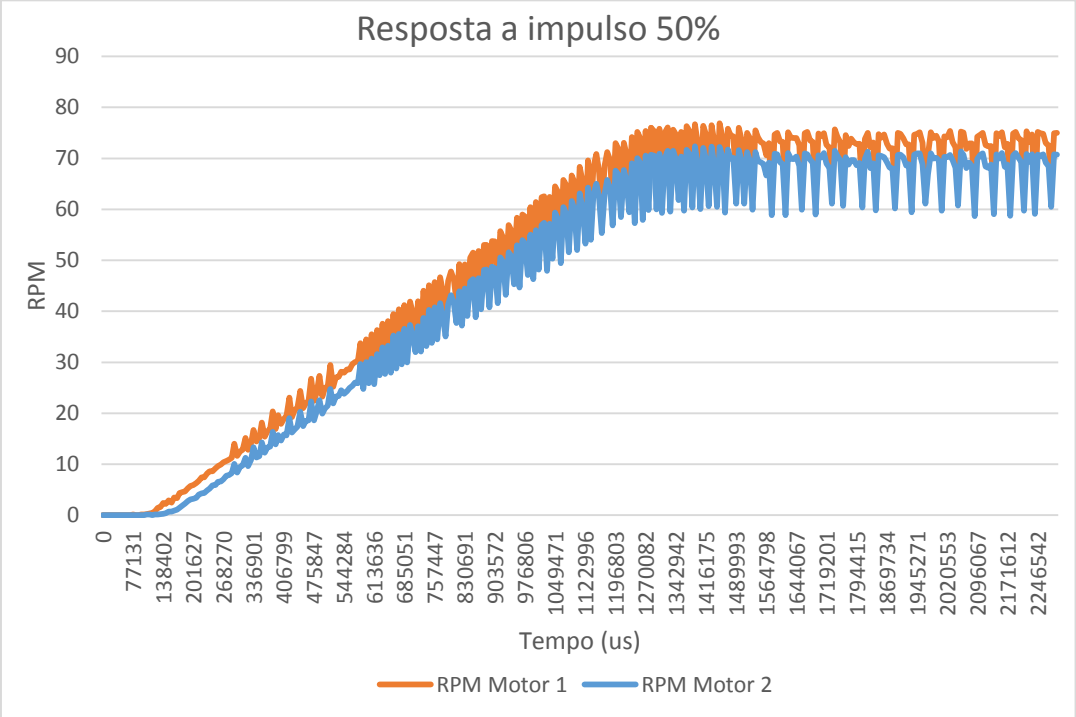

Figura 47- Resposta do motor a um degrau de 50%

Sabendo que a velocidade nominal do motor é de 163 rpm e que a função de transferência de 1º ordem é dada por

$$Gs(s) = \frac{GDC}{\tau s + 1} \qquad (44)$$

Podemos determinar sendo G_{DC} o ganho DC e $\tau = 0.632$ do tempo de estabilização, temos:

$$G_{DC} = \frac{70}{163 * 0.5} = 0{,}85 \quad \tau = 0{,}69 \qquad (45)$$

b. Cálculo de Kp, Ki, Kd

Sabemos que a resposta do sistema depende do fator de amortecimento ζ e da sua frequência natural ω_n. Para implementação do nosso controlador PID, Figura 48, optámos por solicitar uma resposta sub-amortecida e não muito rápida, a fim de evitar fortes acelerações na cadeira, o que seria desconfortável para o utilizador, tendo por isso atribuído um $\zeta = 0{,}8$ e $\omega_n = 1.5$.

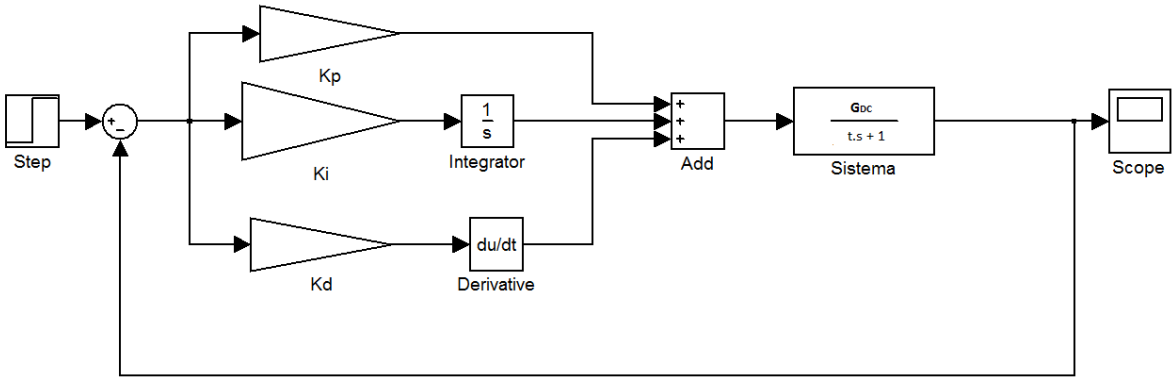

Figura 48- Diagrama do sistema em malha fechada

Sabemos que a equação característica de sistemas de 2ª ordem é dada por:

$$S^2 + 2\zeta\omega_n s + {\omega_n}^2 = 0 \quad (46)$$

A função de transferência de malha fechada do nosso controlador é dada por:

$$C(s) = G(s).E(s) \quad (47)$$

$$E(s) = R(s) - B(s) \quad (48)$$

$$E(s) = R(s) - H(s).C(s) \quad (49)$$

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s).H(s)} \quad (50)$$

Sendo C(s) a saída, G(s) a função de transferência do sistema, H(s) é o feedback, B(s) e E(s) é o erro entre R(s) e B(s). sabemos também que os polos da malha fechada devem ser calculados. Desta forma:

$$1 + G(s).H(s) = 0 \quad (51)$$

$$G(s) = Gc(s).Gs(s) \quad (52)$$

Com H(s)=1

$$1 + Gc(s).Gs(s) = 0 \quad (53)$$

Onde Gc(s) é a função de transferência do nosso controlador, dada por:

$$Gc(s) = \frac{K_d s^2 + K_p s + K_I}{s} \quad (54)$$

Através da substituição nas fórmulas anteriores temos

$$1 + \frac{K_d s^2 + K_p s + K_I}{s} . \frac{G_{DC}}{\tau . s + 1} \quad (55)$$

Simplificando

$$(G_{DC} . K_d + \tau) . s^2 + (G_{DC} . K_p + 1) . s + G_{DC} . K_I = 0 \quad (56)$$

Podemos agora extrair os ganhos para os parâmetros do PID igualando a equação característica do sistema de segunda ordem à equação aqui obtida:

$$K_p = \frac{2\zeta\omega_n - 1}{G_{DC}} = 1{,}63 \quad (57)$$

$$K_I = \frac{{\omega_n}^2}{G_{DC}} = 2{,}61 \quad (58)$$

$$K_D = \frac{1-\tau}{G_{DC}} = 0{,}45 \quad (59)$$

II. Colocação ao serviço do Sistema

Neste capítulo passamos a descrever o modo de iniciar o sistema, sendo que é necessário garantir alguns pré-requisitos que passamos a descrever:

- Computador com plataforma ROS, com rede local configurada com IP 192.168.1.5 para ligação Ethernet ao RPi.

- Verificar que interruptor de Alimentação do RPi se encontra desligado para evitar inicialização do mesmo, devendo estar ligado o interruptor de Alimentação do LRF, encontrando-se os interruptores dispostos de acordo com a Figura 49.

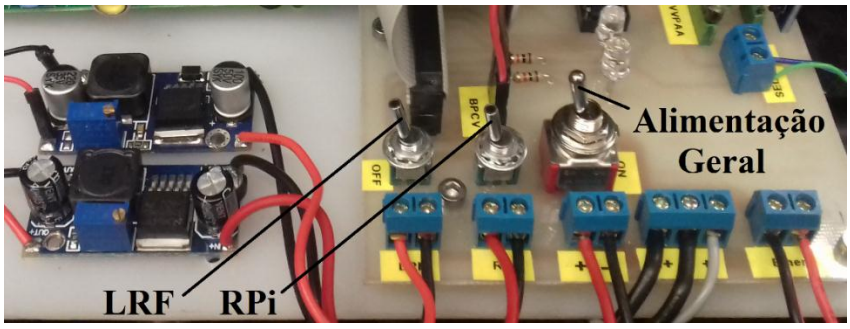

Figura 49- Identificação dos interruptores na PCI

- Ligar interruptor de alimentação geral do equipamento a fim de iniciar o Controlador Roboteq e o LRF.

- Iniciar o PC e correr os seguintes ficheiros que se encontram na pasta /home/pi/salsa:
  - Roscore
  - Rosrun hokuyo_node hokuyo_node
  - ./itf
  - ./Risk
  - ./sel_Vel
  - ./Key_cmd

Ou em alternativa correr o script Salsa_Script que se encontra na mesma localização

- Ligar o interruptor de alimentação do RPi. O programa dedicado neste equipamento arranca automaticamente a sua inicialização, sinalizando o seu estado de funcionamento pela intermitência do led vermelho.

- Através do comutador de modo, é possível selecionar o modo de navegação pretendido, “Livre” ou “NAAC”. Esta seleção é sinalizada através do led azul ativo quando selecionado o modo “NAAC”.

III. Processo de comutação entre sistemas

Conforme requisito inicial, foi mantido na íntegra o sistema inicialmente existente na cadeira de rodas Salsa R2. Neste capítulo descrevemos o processo para comutar entre o sistema Salsa R2 e o sistema Salsa NAAC (Navegação Assistida Anti-Colisão).

a. Instalação do NAAC

No sistema inicial Salsa R2, a alimentação da bateria e os motores estão ligados ao conversor de potência deste sistema conforme Figura 50.

Figura 50- Identificação das fichas dos motores e bateria

Para comutar para o sistema Salsa NAAC é necessário desligar estas três fichas do controlador a fim de as ligar aos respetivos cabos de alimentação do sistema Salsa NAAC e ligar os motores ao controlador do mesmo sistema.

Desta forma, após desligar as fichas as mesmas serão ligados de acordo com o apresentado na Figura 51.

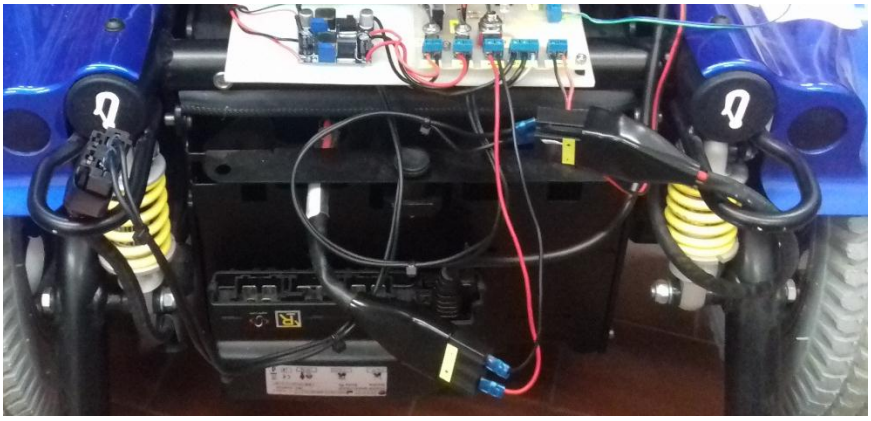

Figura 51- Ligação de alimentação e motores do sistema NAAC

Para a reposição do sistema Salsa R2 apenas temos de proceder do modo inverso, desligar os cabos e ligar as fichas no respetivo controlador.

b. Carregamento de baterias

Para carregar as baterias do sistema deve ser usado o carregador de baterias (Figura 52), o qual possui um indicador do seu estado de funcionamento, permitindo assim saber o estado da carga da bateria.

Figura 52 - Carregamento das baterias da cadeira de rodas

O processo de carregamento das baterias do sistema NAAC é semelhante ao processo original, sendo apenas necessário recolocar a ficha de alimentação do sistema no controlador da cadeira Salsa R2 conforme exemplificado na Figura 53.

Figura 53- Ficha para carregamento de baterias

BIBLIOGRAFIA

[1] Scudellari, Megan - "Self-Driving Wheelchairs Debut in Hospitals and Airports" [Em linha] IEEE SPECTRUM 2017 [Consult. 27 Outubro 2017] https://spectrum.ieee.org/%20the-human-os/biomedical/devices/selfdriving-wheelchairs-debut-in-hospitals-and-airports

[2] "Public Testing of Information Universal Design begins at Haneda Airport" [Em linha] Panasonic 2017 [Consult. 27 Outubro 2017] http://news.panasonic.com/global/press/data/2017/08/en170808-6/en170808-6.html

[3] Simpson RC, et al, "The smart wheelchair component system", J Rehabil. Res.Dev.2004, 41(3B):429–42.

[4] Simpson RC, Poirot D, Baxter MF. , "The Hephaestus smart wheelchair system", IEEE Trans Neural Syst Rehabil Eng. 2002, 10(2):118– 22.

[5] Lopes, A., Pires, G., and Nunes, U. (2012). Robchair: Experiments evaluating brain-computer interface to steer a semi-autonomous wheelchair. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS12).

[6] Vanhooydonck, D., Demeester, E., Hantemann, A., Philips, J., Vanacker, G., Brussel, H. V., and Nuttin, M. (2010). Adaptable navigational assistance for intelligent wheelchairs by means of an implicit personalized user model. Robotics and Autonomous Systems, 58(8):963–977.

[7] Bonarini, A., Ceriani, S., Fontana, G., and Matteucci, M., "Introducing LURCH: a Shared Autonomy Robotic Wheelchair with Multimodal Interfaces "

[8] "Rolland" - The Bremen Autonomous Wheelchair [Em linha] Universitaet Bremen [Consult. 27 Outubro 2017] http://www.informatik.uni-bremen.de/rolland/index_e.htm

[9] Bourhis, G., O. Horn, O. Habert, e A. Pruski. “An autonomous vehicle for people with motor disabilities.” IEEE Robotics & Automation Magazine 8. March 2001. pp. 20-28. ISSN 1070-9932.

[10] A. Fattouh, M. Sahnoun, and G. Bourhis, "Force feedback joystick control of a powered wheelchair: preliminary study" in Systems, Man and Cybernetics, 2004 IEEE International Conference on, 2004

[11] Rory A. Cooper “Rehabilitation Engineering Applied to Mobility and Manipulation”, cap 8.12 User interfaces, 330-331

[12] PG Drives Technology, “R-Net Omni Thechnical manual SK78813”

[13] Snehlata Yadav, bPoonam Sheoran “Smart Wheelchairs - A literature review” - International Journal of Innovative and Emerging Research in Engineering Volume 3, Issue 2, 2016

[14] Urdiales C. “Collaborative Assistive Robot for Mobility Enhancement”, ISRL 27, pp. 41-46

[15] Lopes A., Pires G., Nunes U. (2013) – “Assisted navigation for a brain-actuated intelligent wheelchair”, Robotics and Autonomous Systems 61 (3), 245-258,

[16] Boresnstein J., Koren Y. “Tele.Autonomous Guidance for Mobile Robots, IEEE, Tran. System, Man, and Cybernetics, Special Issue on Unmanned Systems and Vehicles, December, 1990, pp. 1437-1443.

[17] Fox D., Burgard W., and Thrun S. “The dynamic window approach to collision avoidance. IEEE Robotics and Automation Magazine”, 4(1):23–33, March 1997.

[18] Boresnstein J., Koren Y., “The vector Field Histogram a Fast Obstacle-Avoidance for Mobile robots IEEE journal of Robotics and Automation”, Vol. 7, Nº 3., June 1991, pp. 278-288.

[19] http://www.ros.org/

[20] Salsa R2Cadeira de rodas eléctrica [Em linha] SunriseMedical [Consult. 1 Outubro 2017] http://www.sunrisemedical.pt/cadeiras-de-rodas/quickie/cadeiras-de-rodas-electricas/salsa-r2

[21] MAX3232 3-V to 5.5-V Multichannel RS-232 Line Driver/Receiver With ±15-kV ESD Protection [Em linha] Texas Instruments [Consult. 1 Outubro 2017] http://www.ti.com/lit/ds/symlink/max3232.pdf

[22] https://www.raspberrypi.org

[23] MCP3002 [Em linha] Microchip [Consult. 1 Outubro 2017] http://www.microchip.com/wwwproducts/en/MCP3002

[24] "MDC2xxx Motor Controller Datasheet " [Em linha] Roboteq [Consult. 1 Outubro 2017] https://www.roboteq.com/index.php/docman/motor-controllers-documents-and-files/documentation/datasheets/mdc2xxx-datasheet-1/2-datasheet-mdc2xxx/file.

[25] "Advanced Digital Motor Controllers" [Em linha] Roboteq [Consult. 1 Outubro 2017] https://www.roboteq.com/index.php/roboteq-products-and-services/brushed-dc-motor-controllers

[26] "MODULAR INCREMENTAL ENCODER" [Em linha] CUI [Consult. 1 Outubro 2017] http://www.cui.com/product/resource/amt10-v.pdf

[27] "Distance Data Output/UTM-30LX" [Em linha] HOKUYO [Consult. 1 Outubro 2017] https://www.hokuyo-aut.jp/search/single.php?serial=169

[28] Kumar Mau S., Majumdar J., "Kinematics, Localization and control of Differential Drive Mobile Robot", Global Journal of researches in Engineering:H,Robotics & Nano-Tech, Volume 14, Issue 1, Version 1.0 Year 2014

[29] "UMBmark — A Method for Measuring, Comparing, and Correcting Dead-reckoning Errors in Mobile Robots" [Em linha] The University of Michigan [Consult. 1 Outubro 2017] http://www-personal.umich.edu/~johannb/Papers/umbmark.pdf

[30] Borenstein, J. and Feng, L. (1995). "Umbmark: A benchmark test for measuring odometry errors in mobile robots". In SPIE Conference on Mobile Robots, pages 1–12, Philadelphia - USA.

[31] Jung, C. and Chung, W. (2012). Accurate calibration of two wheel differential mobile robots by using experimental heading erros. In IEEE International Conference on Robotics and Automation, Minnesota - USA.

[32] "Equação de Torricelli" [Em linha] Wikipédia [Consult 1 Outubro 2017] https://pt.wikipedia.org/wiki/Equa%C3%A7%C3%A3o_de_Torricelli

[33] - Lopes, A., Rodrigues, J., Perdigão, J., Pires, G., and Nunes, U. (2016). A new hybrid motion planner applied in a brain-actuated robotic wheelchair. IEEE Robotics and Automation Magazine. Vol. 23, N. 4, December, 2016.

[34] "Romeo, the helpful companion robot" [Em linha] Design Indaba [Consult. 1 Outubro 2017] http://www.designindaba.com/articles/creative-work/romeo-helpful-companion-robot

[35] Paiva, Hugo A. D. - "Human-Machine Interface Modules to Support Indoor Navigation of a Robotic Wheelchair", Coimbra, 2015. Dissertação de mestrado.