

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Projeto de Cadeira de Rodas Autônoma

Autor: Lucas Martinelli Reia

Orientador: Prof. Dr. Marco Henrique Terra

São Carlos

2016

LUCAS MARTINELLI REIA

PROJETO DE CADEIRA DE RODAS AUTÔNOMA

Trabalho de Conclusão de Curso apresentado
à Escola de Engenharia de São Carlos, da
Universidade de São Paulo

Curso de Engenharia Elétrica com ênfase em
Eletrônica

ORIENTADOR: Prof. Dr. Marco Henrique Terra

São Carlos
2016

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

R347p Reia, Lucas Martinelli
 Projeto de cadeira de rodas autônoma / Lucas
 Martinelli Reia; orientador Marco Henrique Terra. São
 Carlos, 2016.

Monografia (Graduação em Engenharia Elétrica com
ênfase em Eletrônica) -- Escola de Engenharia de São
Carlos da Universidade de São Paulo, 2016.

1. Navegação autônoma. 2. Robótica móvel. 3.
Cadeira de rodas. 4. Controle. 5. Campos potenciais. 6.
Acessibilidade. I. Título.

FOLHA DE APROVAÇÃO

Nome: Lucas Martinelli Reia

Título: "Projeto de cadeira de rodas autônoma"

Trabalho de Conclusão de Curso defendido e aprovado
em 28 / 11 / 16,

com NOTA 9,2 (nove, dois), pela Comissão Julgadora:

Prof. Titular Marco Henrique Terra - Orientador - SEL/EESC/USP

Prof. Dr. Manoel Luis de Aguiar - SEL/EESC/USP

Prof. Dr. Valdir Grassi Junior - SEL/EESC/USP

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado José Carlos de Melo Vieira Júnior

Resumo

Este projeto propõe a implementação de um sistema para tornar autônoma ou semi-autônoma uma cadeira de rodas motorizada com a utilização de um algoritmo baseado em campos potenciais. Para isso, são apresentados a lista dos componentes eletrônicos que devem ser adquiridos para a implementação de tal sistema e esquemas de como deve ser a instalação destes componentes.

Os resultados das simulações computacionais mostram que o modo semi-autônomo aumenta a segurança do cadeirante ao evitar colisões com obstáculos. Também foi observado que o modo autônomo consegue guiar a cadeira em ambientes fechados para que atinja um destino predeterminado ou para que siga automaticamente o acompanhante em ambientes com vários obstáculos fixos e móveis. No entanto, no modo autônomo a cadeira apresentou grande dificuldade de passar por aberturas estreitas, como portas, e por isso o algoritmo precisa de aperfeiçoamentos para ser utilizado no mundo real.

Palavras-Chave: navegação autônoma, robótica móvel, cadeira de rodas, controle, campos potenciais, acessibilidade.

Abstract

This project proposes the implementation of a system to make autonomous or semi-autonomous a motorized wheelchair with the use of an algorithm based on potential field. In order to do this, it is presented not only the list of electronic components that must be acquired for the implementation of such a system but also the schemes of how the installation of these components must be done.

The results of the computational simulations show that the semi-autonomous mode increases the safety of the wheelchair user by avoiding collisions with obstacles. It has also been observed that the autonomous mode can guide the wheelchair in indoor places to reach a predetermined destination or to automatically follow the companion in environments with several fixed and mobile obstacles. However, in the autonomous mode the wheelchair presented great difficulty in passing through narrow openings such as doors, and so the algorithm needs refinements to be used in the real world.

Keywords: autonomous navigation, mobile robotics, wheelchair, control, potential field, acessibility.

Lista de Figuras

1.1	Quantidade de pessoas com algum tipo de deficiência motora em diferentes faixas de idade, segundo o Censo Demográfico 2010 [1].	21
1.2	Dois tipos diferentes de bengalas.	22
1.3	Dois tipos diferentes de muletas.	22
1.4	Dois tipos diferentes de andadores.	23
1.5	Cadeira de rodas de praia. As rodas grandes facilitam a locomoção sobre a areia. . . .	24
1.6	Cadeira de rodas de banho.	24
1.7	A cadeira de rodas motorizada Freedom SE apresenta um <i>joystick</i> para o acompanhante.	25
2.1	Robô de acionamento diferencial com rodas fixas e seus parâmetros.	28
2.2	Robô de acionamento diferencial representado no sistema inercial.	29
2.3	Trajectoria de um robô de acionamento diferencial.	31
2.4	Exemplo de campo que apresenta um mínimo local.	32
2.5	Diagrama de blocos do algoritmo para robos retangulares.	33
2.6	Geração de forças de repulsão a partir dos pontos de obstáculos.	34
2.7	Geração da força de atração.	35
3.1	Cadeira de rodas Freedom SX.	40
3.2	Drive S da cadeira de rodas Freedom SX.	40
3.3	Dimensões da cadeira Freedom SX em centímetros.	41
3.4	Controlador MDC2460: entrada para <i>encoders</i> , conectores DB15 e USB.	42
3.5	Controlador MDC2460: terminais Faston de 0,25 polegadas.	42
3.6	Esquema das conexões elétricas do controlador MDC2460. As linhas mais espessas representam conexões obrigatórias. Imagem adaptada de [23].	44
3.7	Exemplo de esquema das conexões elétricas do controlador MDC2460 com contator de segurança. Imagem adaptada de [23].	45
3.8	Numeração dos pinos do conector DB15. Imagem adaptada de [23].	45

3.9 Adaptador CABLE-ENC1 para a entrada de <i>encoder</i>	46
3.10 Adaptador BOB15.	47
3.11 Disco de <i>encoder</i>	48
3.12 Sinais de um <i>encoder</i> de quadratura.	49
3.13 Imagem do <i>encoder</i> HEDS-5540#I06.	50
3.14 Resistores de <i>pull-up</i> no <i>encoder</i> HEDS-5540#I06. Imagem adaptada do <i>datasheet</i> do componente.	50
3.15 Conector HEDS-8903.	51
3.16 Diagrama esquemático do sensor de distância baseado em <i>laser</i>	52
3.17 Foto do sensor UST-10LX, fabricado pela Hokuyo.	53
3.18 Diagrama da varredura do UST-10LX.	54
3.19 O círculo vermelho indica o conector JST do sensor UST-10LX.	55
3.20 Foto do Kinect para Xbox One.	56
3.21 Adaptador Kinect para Windows.	57
3.22 Foto da fonte M4-ATX.	59
3.23 Diagrama esquemático da fonte M4-ATX. Imagem adaptada de [32].	60
3.24 Diagrama do sistema de controle.	62
3.25 Conexão dos <i>encoders</i> com o controlador. As cores representam as cores dos cabos dos HEDS-8903.	62
3.26 Esquema das ligações do computador com os sensores. As ligações vermelhas repre- sentam a alimentação, enquanto as pretas são conexões entre os componentes.	63
3.27 Esquema das ligações do controlador MDC2460. Cada cor representa uma espessura diferente de fio. Cinza = 16mm^2 (76A), vermelho = 10mm^2 (57A), amarelo = 4mm^2 (32A), verde = azul = $1,5\text{mm}^2$ (17,5A, menor espessura disponível em [36]). Em preto são as conexões entre componentes.	64
3.28 Esquema das ligações do computador com a bateria. Cada cor representa uma espes- sura diferente de fio. Cinza = 16mm^2 (76A), laranja = $2,5\text{mm}^2$ (24A).	65
3.29 Posição conveniente para os sistemas de coordenadas.	66
3.30 Modelo simplificado da cadeira de rodas modelado no Blender.	67
3.31 Hierarquia dos componentes do modelo da cadeira de rodas no V-REP.	68
3.32 Posicionamento do Kinect no modelo.	69
3.33 <i>fastHokuyo</i> posicionado na frente e <i>fastHokuyo#0</i> atrás da cadeira. Em vermelho são os feixes de <i>laser</i> emitidos pelos sensores.	69

3.34	<i>fastHokuyo</i> posicionado no vértice frontal esquerdo e <i>fastHokuyo#0</i> no vértice frontal direito da cadeira. Em vermelho são os feixes de <i>laser</i> emitidos pelos sensores.	70
3.35	Propriedades dinâmicas dos atuadores.	71
3.36	Contorno de forma retangular.	73
3.37	Contorno de forma de pentágono irregular.	74
3.38	Ambiente 1 da simulação.	76
3.39	Ambiente 2 da simulação.	77
3.40	Diagrama esquemático da interação do programa em C++ com o V-REP.	77
4.1	Ambiente 1, contorno retangular, um sensor na frente e outro atrás, modo autônomo. $K = 0,00025$, $kf = 0,8$ e $kr = 0,2$. Houve várias colisões.	81
4.2	Ambiente 1, contorno retangular, um sensor em cada vértice frontal, modo autônomo. $K = 0,00025$, $kf = 0,8$ e $kr = 0,2$. Foram alteradas as posições dos sensores LIDAR em relação à simulação da Figura 4.1. Houve uma colisão.	82
4.3	Ambiente 1, contorno retangular, um sensor em cada vértice frontal, modo autônomo. $K = 0,0003$, $kf = 0,8$ e $kr = 0,2$. Foi utilizado um coeficiente de repulsão K maior que o da simulação da Figura 4.2. A cadeira entrou em um mínimo local e não completou o trajeto.	82
4.4	Ambiente 1, contorno retangular, um sensor em cada vértice frontal, modo autônomo. $K = 0,00025$, $kf = 0,6$ e $kr = 0,4$. Foi utilizada uma proporção kr/kf maior que a da simulação da Figura 4.2. A cadeira entrou em um mínimo local e não completou o trajeto.	83
4.5	Ambiente 1, contorno de pentágono irregular, um sensor em cada vértice frontal, modo autônomo. $K = 0,00015$, $kf = 0,85$ e $kr = 0,15$. Diferentemente das simulações anteriores, foi empregado o contorno de pentágono irregular. A cadeira entrou em um mínimo local no momento de cruzar a porta.	83
4.6	Ambiente 1, contorno de pentágono irregular, um sensor em cada vértice frontal, modo semi-autônomo. $K = 0,00015$, $kf = 0,85$ e $kr = 0,15$. Com o modo semi-autônomo foi possível cumprir o trajeto.	84
4.7	Ambiente 1, contorno de pentágono irregular, um sensor em cada vértice frontal, modo autônomo. $K = 0,00015$, $kf = 0,85$ e $kr = 0,15$. A cadeira foi capaz de seguir o acompanhante até certo ponto, entrando em um mínimo local no momento de cruzar a porta.	84

4.8	Ambiente 1, contorno de pentágono irregular, um sensor em cada vértice frontal, modo autônomo. $K = 0,00015$, $k_f = 0,85$ e $k_r = 0,15$. Em um ambiente espaçoso, a cadeira foi capaz de seguir o acompanhante sem problemas.	85
I.1	Tipos de rodas.	97

Lista de Tabelas

3.1	Terminais da entrada para <i>encoder</i> do controlador MDC2460.	46
3.2	Estados de um <i>encoder</i> de quadratura.	49
3.3	Características elétricas do <i>encoder</i> HEDS-5540#I06 para $V_{CC}=5,0V$ a $25^{\circ}C$	49
3.4	Condições máximas de operação <i>encoder</i> HEDS-5540#I06.	50
3.5	Função de cada cabo do conector HEDS-8903.	51
3.6	Características do Hokuyo UST-10LX.	53
3.7	Cabos de alimentação e entrada/saída do Hokuyo UST-10LX.	55
3.8	Especificações do Kinect para Xbox One.	56
3.9	Especificações do computador.	58
3.10	Portas presentes na placa mãe IMBM-B75A.	59
3.11	Valores máximos de corrente que podem ser fornecidos pela fonte M4-ATX para cada nível de tensão.	60
3.12	Constantes da equação de cinemática inversa.	66
3.13	Componentes do modelo da cadeira de rodas no V-REP.	68
A.1	Modelos de sensores LIDAR	95

Siglas

ABNT	Associação Brasileira de Normas Técnicas
AC	<i>Alternating Current</i> - Corrente Alternada
API	<i>Application Programming Interface</i> - Interface de Programação de Aplicação
CC	Corrente Contínua
CPR	<i>Counts Per Revolution</i> - Contagens Por Revolução
DC	<i>Direct Current</i> - Corrente Contínua
DIP	<i>Dual In-line Package</i> - Pacote Duplo em Linha
FPS	<i>Frames Per Second</i> - Quadros Por Segundo
HDD	<i>Hard Disk Drive</i> - Unidade de Disco Rígido
IBGE	Instituto Brasileiro de Geografia e Estatística
IP	<i>Internet Protocol</i> - Protocolo de Internet
LED	<i>Light Emitting Diode</i> - Diodo Emissor de Luz
LIDAR	<i>Light Detection and Ranging</i> - Detecção e Medição de Distância por Luz
PC	<i>Personal Computer</i> - Computador pessoal
PID	<i>Proportional–Integral–Derivative</i> - Proporcional Integral Derivativo
PWM	<i>Pulse Width Modulation</i> - Modulação de Largura de Pulso
RGB	<i>Red Green Blue</i> - Vermelho, Verde e Azul
ROS	<i>Robot Operating System</i> - Sistema Operacional para Robô
SFML	<i>Simple and Fast Multimedia Library</i> - Biblioteca Multimídia Simples e Rápida
SDK	<i>Software Development Kit</i> - Kit de Desenvolvimento de <i>Software</i>
SSD	<i>Solid-State Drive</i> - Unidade de Estado Sólido
TTL	<i>Transistor-Transistor Logic</i> - Lógica Transistor-Transistor
USB	<i>Universal Serial Bus</i> - Barramento Serial Universal

Sumário

1	Introdução	21
1.1	Objetivos	26
1.2	Justificativa	26
2	Embasamento Teórico	27
2.1	Cinemática da cadeira de rodas motorizada	27
2.2	Algoritmo para condução de veículo autônomo	31
2.3	Intersecção entre segmentos de reta	36
3	Materiais e Métodos	39
3.1	Materiais	39
3.1.1	Cadeira de rodas motorizada	39
3.1.2	Controlador	41
3.1.3	Sensores	47
3.1.4	Computador	58
3.1.5	<i>Softwares</i>	61
3.2	Métodos	61
3.2.1	Projeto	61
3.2.2	Simulações	66
4	Resultados	79
4.1	Simulações	79
4.1.1	Ambiente 1	79
4.1.2	Ambiente 2	81
4.2	Implementação prática	85
5	Conclusão e Perspectivas	87
5.1	Conclusão	87
5.2	Perspectivas	89

A	Modelos de sensores LIDAR	95
I	Tipos de rodas	97

Capítulo 1

Introdução

O Censo Demográfico 2010 [1] foi realizado pelo Instituto Brasileiro de Geografia e Estatística (IBGE) com o objetivo de obter informações a respeito da população brasileira. Os dados dos brasileiros foram coletados por 191 mil recenseadores que visitaram 67,6 milhões de residências em 5.565 municípios. Os recenseadores aplicaram um questionário padrão abrangendo temas relacionados aos domicílios (como iluminação, abastecimento de água e saneamento) e aos moradores (como escolaridade, trabalho, idade, etc.) do Brasil

Um dos assuntos abordados pelo censo foram as deficiências física e mental. As pessoas entrevistadas tiveram de classificar o quão severa são suas deficiências por meio de 3 níveis: não consegue de modo algum, grande dificuldade ou alguma dificuldade. Constatou-se que cerca de 7% da população (mais de 13 milhões de pessoas) possui algum grau de deficiência motora, sendo que a maior parte possui idade mais avançada, como mostra a Figura 1.1.

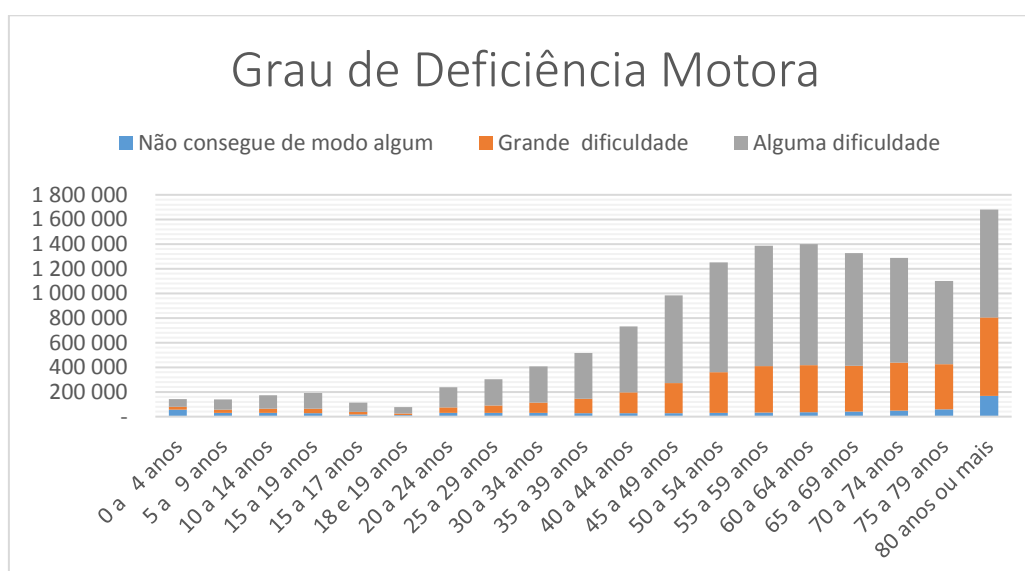


Figura 1.1: Quantidade de pessoas com algum tipo de deficiência motora em diferentes faixas de idade, segundo o Censo Demográfico 2010 [1].

Com o fim de auxiliar e facilitar a locomoção de pessoas com deficiência motora, diversos equipamentos foram desenvolvidos ao longo da história. Alguns dos mais conhecidos são bengalas (Figura 1.2), muletas (Figura 1.3) e andadores (Figura 1.4), que ajudam pessoas que precisam de assistência mas conseguem se sustentar de alguma forma com as pernas. Nesses aparelhos o indivíduo utiliza seus membros superiores para apoiar parte do peso de seu corpo, diminuindo a carga mecânica sobre as pernas. Há também o aumento da quantidade de pontos de contato do usuário com o solo, facilitando assim a estabilidade dos movimentos de locomoção.



Figura 1.2: Dois tipos diferentes de bengalas.¹



Figura 1.3: Dois tipos diferentes de muletas.²

¹Imagens retiradas de <http://www.cirurgicaromana.com.br/>. Acesso em 01 de dezembro de 2016.

²Imagens retiradas de <http://www.fibracirurgica.com.br/tecnologia-assistiva/muleta>. Acesso em 01 de dezembro de 2016.



Figura 1.4: Dois tipos diferentes de andadores.³

Os indivíduos que possuem grande dificuldade física ou não conseguem se sustentar com as pernas de forma alguma podem se locomover com o uso de cadeiras de rodas. Atualmente o mercado oferece diversos modelos diferentes, sendo alguns deles projetados para situações específicas (como a cadeira de praia ilustrada na Figura 1.5 e a cadeira de banho apresentada na Figura 1.6), enquanto que os modelos mais comuns são manuais e possuem 4 rodas: 2 rodas fixas e 2 rodas *caster*. As cadeiras de rodas manuais dependem de propulsão humana para se movimentarem, ou seja, o próprio cadeirante deve girar as rodas da cadeira ou um acompanhante deve empurrá-la. É importante mencionar que no Brasil é obrigatório que vias e espaços públicos, ou privados destinados ao uso coletivo, sejam adaptados para permitir a circulação de pessoas com deficiência motora ou mobilidade reduzida [2]. Outras leis e normas ABNT referentes ao assunto podem ser encontradas em [3].

Apesar de facilitar a locomoção de indivíduos com debilidade física, estudos apontam que uma alta porcentagem de usuários de cadeiras de rodas manuais sofrem de dores nos ombros e/ou síndrome do túnel do carpo [4, 5, 6] (apud [7]) devido aos movimentos repetitivos necessários para a condução da cadeira. Para essas pessoas, uma alternativa que o mercado oferece é a cadeira de rodas motorizada com motor elétrico.

Cadeiras de rodas motorizadas possuem acionamento diferencial, ou seja, são utilizados dois motores elétricos independentes, um para cada roda fixa, enquanto as rodas *caster* servem apenas de apoio. Os motores elétricos são alimentados por baterias que geralmente se encontram embaixo

³Imagens retiradas de <http://www.ortopediajaguaribe.com.br/index.php/produtos/andadores/andador-aluminio-pop-dobravel-e-articulavel-detail>, e <http://www.providacirurgica.com.br/index.php/servicos>. Acesso em 01 de dezembro de 2016.



Figura 1.5: Cadeira de rodas de praia. As rodas grandes facilitam a locomoção sobre a areia.⁴



Figura 1.6: Cadeira de rodas de banho.⁵

do assento. Por meio de um *joystick*, o usuário consegue controlar os movimentos da cadeira, evitando assim um alto desgaste físico para realizar a condução.

As cadeiras de rodas motorizadas disponíveis no mercado dependem da percepção e coordena-

⁴Imagem retirada de <http://www.ortobras.com.br/produto/ipanema>. Acesso em 01 de dezembro de 2016.

⁵Imagem adaptada de <http://www.ortopediajaguaribe.com.br/index.php/produtos/cadeiras/banho/banho-pop-aluminio-detail>. Acesso em 01 de dezembro de 2016.

ção do usuário para que os movimentos sejam executados, ou seja, o *feedback* de posição e velocidade é feito pelo próprio usuário. Cadeirantes com grau mais elevado de deficiência não são capazes de conduzir uma cadeira de rodas de forma segura, sendo necessária a presença de um acompanhante para efetuar a condução. Pensando nisso, alguns modelos de cadeira de rodas motorizada (como o apresentado na Figura 1.7) disponibilizam o *joystick* em uma posição de fácil acesso pelo acompanhante, localizado na parte de trás do encosto.



Figura 1.7: A cadeira de rodas motorizada Freedom SE apresenta um *joystick* para o acompanhante.⁷

Com o avanço da tecnologia, o desenvolvimento de novos componentes eletrônicos e o barateamento de diferentes tipos de sensores, têm-se estudado nos últimos anos a implementação de sistema autônomo em veículos de transporte de modo geral, como carros [8] e caminhões [9], e de pessoas com dificuldade motora, como cadeira de rodas. O desenvolvimento de uma cadeira de rodas autônoma permite que o cadeirante se locomova nos mais diversos ambientes sem exercer nenhum esforço físico e sem a necessidade de ser impulsionado pelo acompanhante. Encontramos na literatura diversos exemplos em que um sistema autônomo foi implementado em uma cadeira de rodas, como em [10, 11, 12]. Outra implementação muito encontrada na literatura é a chamada semi-autônoma, na qual os movimentos guiados pelo usuário são assistidos por diversos sensores espalhados pela cadeira e por um módulo de controle [13, 14, 15]. Desse modo, o sistema computadorizado da cadeira evita movimentos abruptos e colisões que possam ser causados pelo cadeirante ou acompanhante por acidente.

⁷Imagens retiradas de <http://www.shopfisio.com.br/cadeira-de-rodas-motorizada-freedom-se-p1063235>, e <http://www.orthopinhais.com.br/CADEIRA-MOTORIZADA-FREEDOM-SE/prod-673703/>. Acesso em 01 de dezembro de 2016.

1.1 Objetivos

Os objetivos deste projeto podem ser divididos em objetivo principal e objetivos específicos. O objetivo principal é a implementação de um sistema de controle em uma cadeira de rodas motorizada que permita a realização de alguns movimentos de forma autônoma, evitando obstáculos e transportando o usuário de maneira segura. Para atingi-lo, devem ser cumpridos os seguintes objetivos específicos: desenvolvimento teórico de um sistema autônomo com auxílio de simulações computacionais; definição da lista de componentes e sensores que devem ser adquiridos para implementação do sistema autônomo em uma cadeira de rodas real; planejamento das conexões desses componentes com base em suas características de operação (como corrente e tensão) listadas em seus *datasheets*. Espera-se que, ao final do projeto, a cadeira seja capaz de seguir automaticamente uma pessoa que esteja acompanhando cadeirante.

1.2 Justificativa

Esse trabalho se justifica pelo aumento na qualidade de vida que poderá proporcionar aos usuários de cadeira de rodas e seus acompanhantes. A implementação de um sistema semi-autônomo, por exemplo, aumentará a segurança na locomoção do usuário ao evitar colisões com obstáculos e suavizar trajetórias abruptas. Por sua vez, a implementação de um sistema autônomo poderá conferir maior liberdade para o usuário e/ou acompanhante ao fazer com que a cadeira de rodas siga automaticamente o acompanhante ou se desloque autonomamente para uma posição espacial específica.

Capítulo 2

Embasamento Teórico

Neste capítulo, são apresentados os aspectos teóricos necessários para o desenvolvimento do projeto.

2.1 Cinemática da cadeira de rodas motorizada

O movimento da cadeira de rodas pode ser descrito matematicamente com expressões que determinam sua velocidade e posição em função do tempo. Como os motores elétricos proporcionarão uma dada rotação às rodas, é preciso estabelecer como esta rotação se relaciona com o movimento geral da cadeira.

Para poder obter uma relação entre as velocidades angulares das rodas da cadeira e as velocidades linear e angular da cadeira como um todo, é necessário analisar a cinemática do problema, quais são suas restrições de movimento e condições de contorno. Em [16] são feitas deduções dos modelos de cinemática direta e inversa de robôs móveis.

Cinemática direta é quando se determina a velocidade de um robô a partir das velocidades angulares de suas rodas. Cinemática inversa, como o próprio nome já diz, é o oposto: determina-se a velocidade dos atuadores a partir da velocidade desejada para o robô. O Anexo I apresenta alguns tipos de rodas que podem ser empregadas em projetos de robôs móveis.

Uma cadeira de rodas motorizada pode ser representada por um veículo de acionamento diferencial (*differential drive vehicle*), pois sua locomoção é feita por dois motores independentes, um acoplado a cada roda fixa, que giram em torno de um eixo comum. No robô de acionamento diferencial da Figura 2.1, tem-se um sistema de coordenadas global com origem em um ponto de referência inercial (I), um sistema de coordenadas com origem fixa na plataforma do robô (R), um com origem fixa no centro da roda direita (W1) e outro na esquerda (W2). A posição da roda direita em relação ao sistema de coordenadas inercial é dada por:

$$r_{IW1} = r_{IR} + r_{RW1}. \quad (2.1)$$

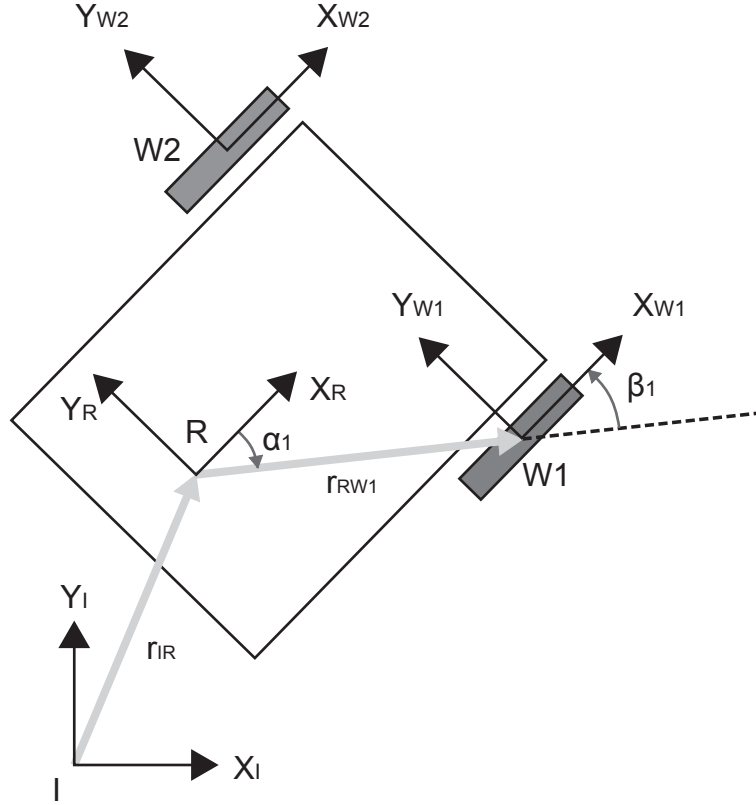


Figura 2.1: Robô de acionamento diferencial com rodas fixas e seus parâmetros.

Como as rodas são fixas na plataforma do robô, a velocidade de translação da roda direita em relação a I é:

$$\frac{d(r_{IW1})}{dt} = v_{IW1} = v_{IR} + \omega_{IR} \times r_{RW1} \quad , \quad (2.2)$$

em que v_{IR} e ω_{IR} são as velocidades linear e angular (em torno de Z_R) do robô, respectivamente. As restrições de movimento de um robô com rodas são impostas pelo tipo de roda utilizada. Uma roda fixa ideal possui as seguintes restrições:

- A velocidade instantânea de translação no eixo X_{W1} é igual à velocidade angular $\dot{\phi}_1$ da roda em torno do eixo Y_{W1} multiplicada pelo seu raio r_1 :

$$\dot{x}_{W1} = \dot{\phi}_1 r_1. \quad (2.3)$$

- A roda não desliza, então não há deslocamento instantâneo no eixo Y_{W1} :

$$\dot{y}_{W1} = 0. \quad (2.4)$$

- Assume-se que o movimento é planar, então não há deslocamento no eixo Z_{W1} :

$$\dot{z}_{W1} = 0. \quad (2.5)$$

Com isso, a velocidade da roda direita v_{IW1} expressa em $W1$ é:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}_{W1} = \begin{bmatrix} \dot{\phi}_1 r_1 \\ 0 \\ 0 \end{bmatrix} = v_{IW1_{W1}}. \quad (2.6)$$

Para poder utilizar a Equação (2.6), a Equação (2.2) também deve ser expressa em relação a $W1$:

$$v_{IW1_{W1}} = v_{IR_{W1}} + \omega_{IR_{W1}} \times r_{RW1_{W1}}. \quad (2.7)$$

Sendo $[\dot{x}, \dot{y}, 0]_I^T$ a velocidade do robô em relação a I (v_{IR}) e R_z a matriz de rotação em torno do eixo Z , a velocidade v_{IR} pode ser expressa no sistema de coordenadas da roda rotacionando desde o sistema inercial. A Figura 2.2 mostra que a diferença angular entre os sistemas de coordenadas R e I é de Θ , e pela Figura 2.1 a diferença angular entre $W1$ e R é de $\alpha_1 + \beta_1$.

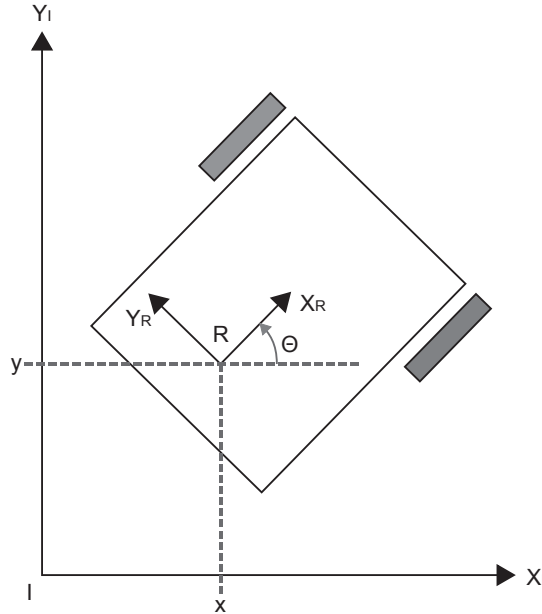


Figura 2.2: Robô de acionamento diferencial representado no sistema inercial.

Desse modo, para se obter v_{IR} em relação ao sistema de coordenadas $W1$ ($v_{IR_{W1}}$), faz-se:

$$v_{IR} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \end{bmatrix}_I, \quad (2.8)$$

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.9)$$

$$v_{IR_{W1}} = R_z(-\beta_1 - \alpha_1) R_z(-\Theta) v_{IR}. \quad (2.10)$$

Sabendo que a velocidade angular do robô é igual a $\dot{\Theta}$ em torno do eixo Z_R , que $\dot{\Theta}_R = \dot{\Theta}_I = \dot{\Theta}_{W1}$, e utilizando a forma matricial do produto vetorial, obtém-se:

$$\omega_{IR_{W1}} \times r_{RW1_{W1}} = \begin{bmatrix} 0 & -\dot{\Theta} & 0 \\ \dot{\Theta} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} |r_{RW1}| \cos(\beta_1) \\ -|r_{RW1}| \sin(\beta_1) \\ 0 \end{bmatrix} = \begin{bmatrix} |r_{RW1}| \sin(\beta_1) \\ |r_{RW1}| \cos(\beta_1) \\ 0 \end{bmatrix} \dot{\Theta}. \quad (2.11)$$

Substituindo as Equações (2.6), (2.9), (2.10) e (2.11) em (2.7):

$$\begin{bmatrix} \cos(\alpha_1 + \beta_1) & \sin(\alpha_1 + \beta_1) & 0 \\ -\sin(\alpha_1 + \beta_1) & \cos(\alpha_1 + \beta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} R_z(-\Theta) \begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \end{bmatrix}_I + \begin{bmatrix} |r_{RW1}| \sin(\beta_1) \\ |r_{RW1}| \cos(\beta_1) \\ 0 \end{bmatrix} \dot{\Theta} = \begin{bmatrix} \dot{\phi}_1 r_1 \\ 0 \\ 0 \end{bmatrix}. \quad (2.12)$$

A Equação (2.12) pode ser reescrita como:

$$\begin{bmatrix} \cos(\alpha_1 + \beta_1) & \sin(\alpha_1 + \beta_1) & |r_{RW1}| \sin(\beta_1) \\ -\sin(\alpha_1 + \beta_1) & \cos(\alpha_1 + \beta_1) & |r_{RW1}| \cos(\beta_1) \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\Theta} \end{bmatrix}_R = \begin{bmatrix} \dot{\phi}_1 r_1 \\ 0 \end{bmatrix}. \quad (2.13)$$

A solução é análoga para a roda esquerda:

$$\begin{bmatrix} \cos(\alpha_2 + \beta_2) & \sin(\alpha_2 + \beta_2) & |r_{RW2}| \sin(\beta_2) \\ -\sin(\alpha_2 + \beta_2) & \cos(\alpha_2 + \beta_2) & |r_{RW2}| \cos(\beta_2) \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\Theta} \end{bmatrix}_R = \begin{bmatrix} \dot{\phi}_2 r_2 \\ 0 \end{bmatrix}. \quad (2.14)$$

Desse modo, a solução da cinemática inversa de um robô de acionamento diferencial é dada por:

$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{r_1} & 0 \\ 0 & \frac{1}{r_2} \end{bmatrix} \begin{bmatrix} \cos(\alpha_1 + \beta_1) & \sin(\alpha_1 + \beta_1) & |r_{RW1}| \sin(\beta_1) \\ \cos(\alpha_2 + \beta_2) & \sin(\alpha_2 + \beta_2) & |r_{RW2}| \sin(\beta_2) \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\Theta} \end{bmatrix}_R. \quad (2.15)$$

Além das rodas fixas, uma cadeira de rodas possui também rodas de apoio, que geralmente são rodas *caster*. Em [16] é feita a dedução das equações para diversos tipos de roda, entre elas a *caster* e a esférica. Chegou-se à conclusão de que as restrições de movimento de rodas de apoio desses tipos não impõem restrições ao movimento do robô, então elas não precisam ser incluídas nas deduções das equações.

Como as rodas fixas são paralelas e não podem deslizar lateralmente, a cadeira de rodas irá sempre percorrer uma trajetória circular, e a reta que liga os eixos das rodas passa pelo centro da trajetória (Figura 2.3). O raio da trajetória pode ser zero (quando o robô gira sem transladar) ou tender ao infinito (quando o robô anda em linha reta). Por causa das limitações de movimento, cadeiras de rodas são classificadas como veículo não-holonômico. Em robótica móvel, são holonômicos os veículos que podem se movimentar instantaneamente para qualquer posição dentro de seu espaço de trabalho, sem nenhuma restrição.

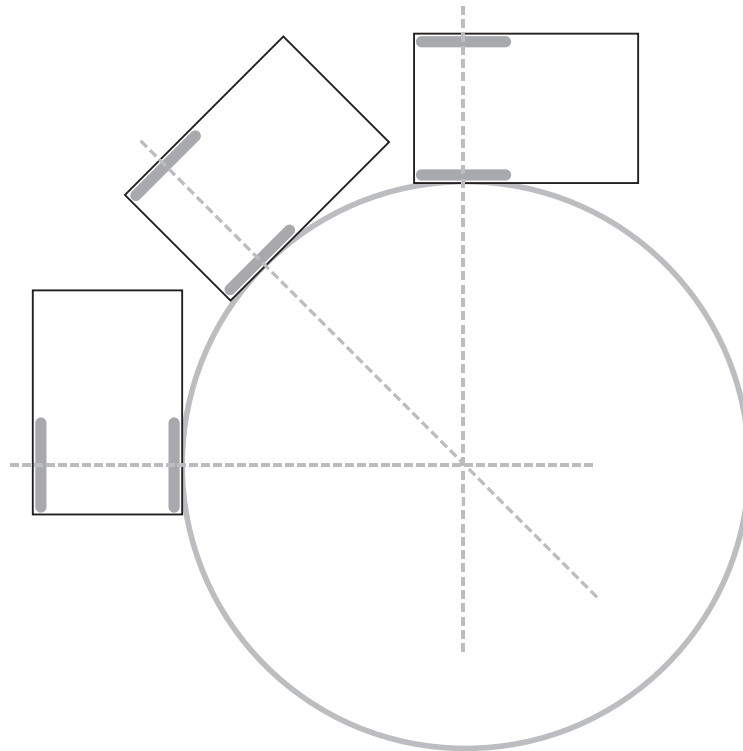


Figura 2.3: Trajetória de um robô de acionamento diferencial.

2.2 Algoritmo para condução de veículo autônomo

Para realizar a condução de robôs móveis, são empregados algoritmos que analisam o robô e o ambiente e determinam qual ação deve ser tomada. Um dos algoritmos existentes é o de campos

potenciais, em que os obstáculos do ambiente são representados por pontos de potencial de repulsão (repelindo o robô), e o destino do robô é um ponto de potencial de atração (atraindo o robô) [17]. A força de atração tem módulo fixo em qualquer região do espaço, enquanto as forças de repulsão são inversamente proporcionais à distância entre o robô e os obstáculos. Para se deslocar até o destino, o veículo percorre uma trajetória tangente aos vetores de força resultante do campo.

O principal problema desse tipo de algoritmo é que nem sempre a posição final obtida é a posição desejada. A Figura 2.4 apresenta um campo vetorial gerado pela presença de um obstáculo (representado pelas linhas pretas) e de um destino para o robô (representado pelo ponto preto). Cada vetor representa a força resultante em seu respectivo ponto no espaço. É possível observar que existe uma região em que a força de atração do destino e as forças de repulsão do obstáculo se anulam, e essa região em que a força resultante é nula corresponde a um mínimo local no módulo da função do campo. Se o robô entrar nessa região, ele não conseguirá sair e ficará preso. Como um dos objetivos do projeto é fazer uma cadeia de rodas seguir um acompanhante que estará sempre próximo, acredita-se que os problemas com mínimo local serão reduzidos.

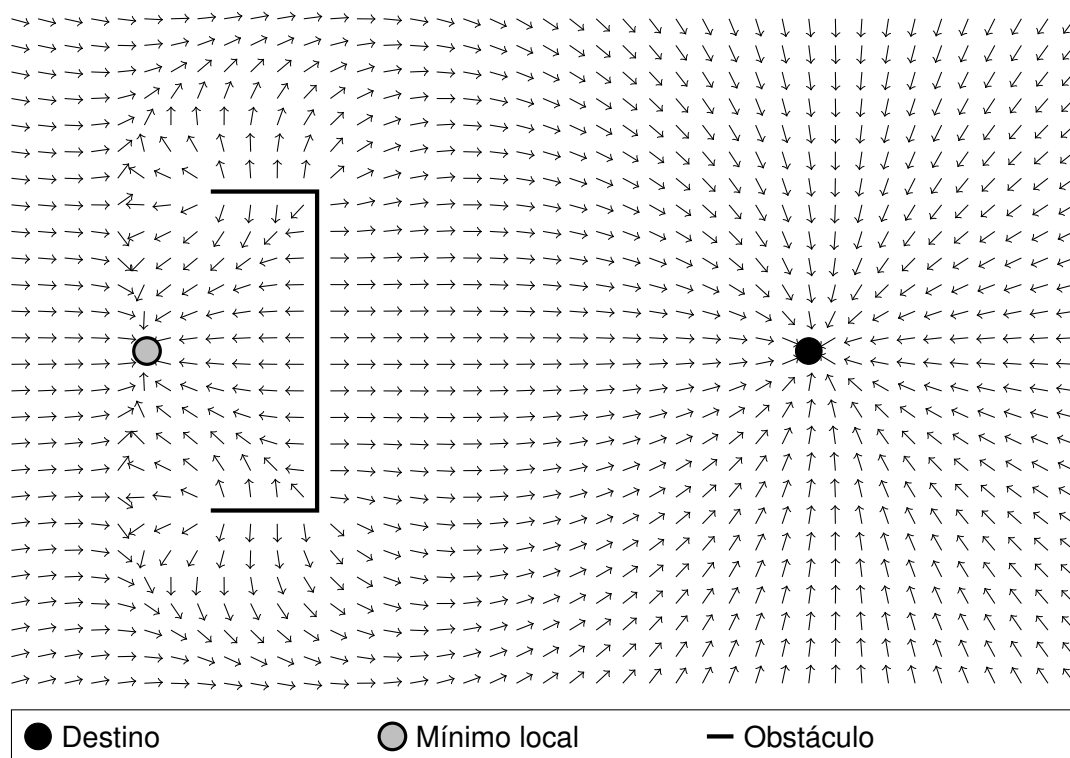


Figura 2.4: Exemplo de campo que apresenta um mínimo local.

Outro problema é que o método não leva em consideração as restrições de movimento específicas de um robô, e precisa ser adaptado para ser aplicado em veículos não-holonômicos. Em [18, 19] é proposto um algoritmo de desvio de obstáculos utilizando campos potenciais para robôs móveis

retangulares de acionamento diferencial, que posteriormente foi implementado em uma cadeira de rodas em [20, 21, 22]. As vantagens do algoritmo de campos potenciais são sua simplicidade e seu baixo custo computacional, que proporcionam uma resposta rápida do robô às alterações no ambiente, e as vantagens dessa implementação em específico é que ela leva em consideração o formato do robô de acionamento diferencial e suas características não-holonômicas, sendo adequada para utilizar em uma cadeira de rodas. A Figura 2.5 explica como o algoritmo funciona.

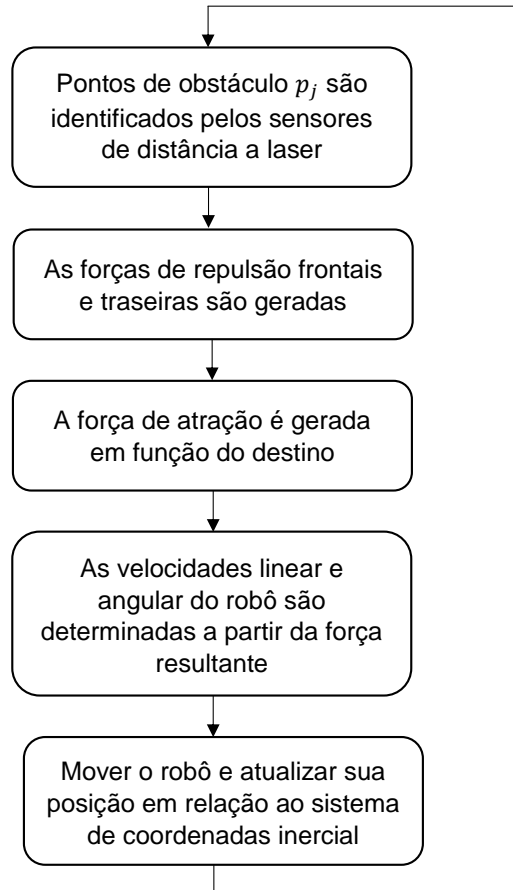


Figura 2.5: Diagrama de blocos do algoritmo para robos retangulares.

Nesse algoritmo, os n_p pontos detectados por um sensor de distância de varredura a laser acoplado ao robô são chamados de pontos de obstáculo p_j , $j = 1, 2, \dots, n_p$. Sabendo que I é o sistema de coordenadas inercial e R o sistema de coordenadas fixo ao robô, a posição atual do veículo é dada por $[X, Y, \Theta]_I^T$, e a posição de destino por $[X_G, Y_G, \Theta_G]_I^T$. O ponto $[X, Y]_I^T$ é o ponto médio entre os eixos das rodas do veículo, e também a origem do sistema de coordenadas fixo no robô. Θ é a diferença angular entre X_R e X_I e representa a orientação do veículo, como mostra a Figura 2.6.

Como veículos de acionamento diferencial não podem se mover lateralmente ($\dot{y}_R = 0$), os pontos de ação onde são aplicadas as forças de atração e repulsão são $rf = [xf, 0]_R^T$ e $rr = [-xr, 0]_R^T$, localizados na frente e atrás do robô. Para levar em consideração o formato do veículo, as forças de

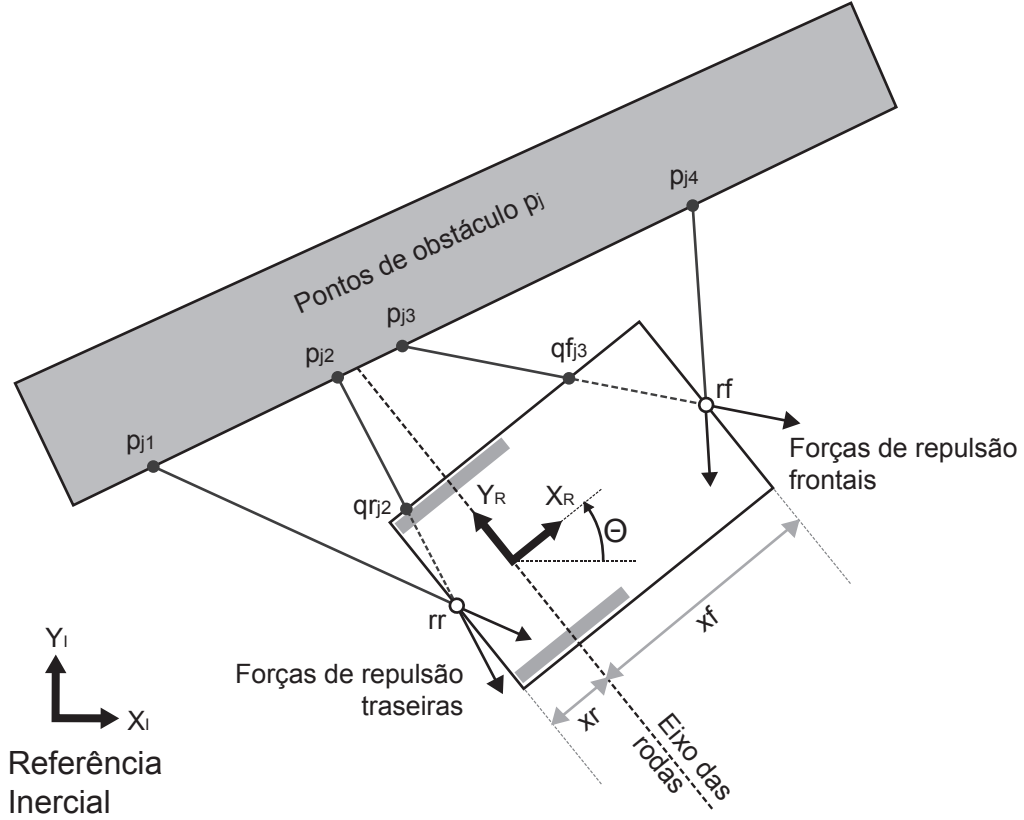


Figura 2.6: Geração de forças de repulsão a partir dos pontos de obstáculos.

repulsão são calculadas em função da distância entre o obstáculo e o corpo ou contorno do veículo.

A força de repulsão de um ponto $p_j = [p_{jx}, p_{jy}]_R^T$, expressada no sistema de coordenadas do robô, é calculada fazendo:

$$F_{fj} = \frac{K}{|q_{fj} - p_j|^2} \frac{rf - p_j}{|rf - p_j|}, \text{ se } p_{jx} > 0, \quad (2.16)$$

$$F_{rj} = \frac{K}{|q_{rj} - p_j|^2} \frac{rr - p_j}{|rr - p_j|}, \text{ se } p_{jx} < 0. \quad (2.17)$$

Se $p_{jx} > 0$, a força é aplicada ao ponto de ação frontal rf , resultando na força de repulsão frontal F_f . Caso contrário, a força é de repulsão traseira F_r , aplicada sobre o ponto de ação traseiro rr .

A constante K determina a intensidade da força de repulsão. q_{fj} e q_{rj} são os pontos de interseção entre o contorno do robô e os segmentos de reta que ligam p_j a rf e rr , respectivamente.

A força de atração F_a tem módulo 1 e puxa o ponto de ação frontal em direção ao destino. Sendo R o sistema de coordenadas do robô na posição atual e R' o sistema na posição de destino, a força F_a é tangente à circunferência que passa pelo ponto de ação frontal do robô na posição atual (rf) e que é tangente ao eixo $X_{R'}$ no ponto rf' (ponto de ação frontal na posição de destino), como pode ser visto na Figura 2.7. A força de atração é calculada fazendo:

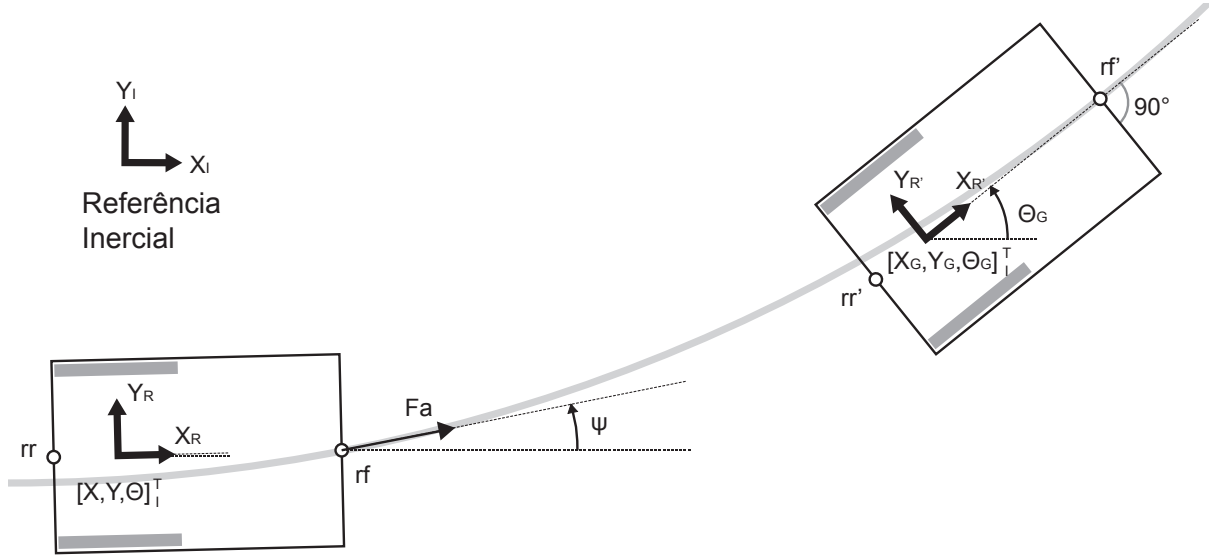


Figura 2.7: Geração da força de atração.

$$\theta_G = \Theta_{G_I} - \Theta_I, \quad (2.18)$$

$$\begin{bmatrix} x'_G \\ y'_G \end{bmatrix}_R = Rot(-\Theta_I) \begin{bmatrix} X_G - X \\ Y_G - Y \end{bmatrix}_I + Rot(\theta_G) rf - rf, \quad (2.19)$$

$$\psi = 2 \arctg\left(\frac{y'_G}{x'_G}\right) - \theta_G, \quad (2.20)$$

$$F_a = \begin{bmatrix} \cos(\psi) \\ \sin(\psi) \end{bmatrix}_R, \quad (2.21)$$

em que $Rot(\alpha)$ é a matrix de rotação:

$$Rot(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}. \quad (2.22)$$

A força resultante F é calculada sobre o ponto de ação frontal. As forças de repulsão frontais e traseiras agem sobre o robô como se este fosse uma alavanca em que o ponto fixo é a origem de R. Se um veículo estiver se deslocando paralelamente a uma parede, as forças no ponto de ação traseiro farão com que o veículo se aproxime da parede, enquanto as forças no ponto de ação frontal farão com que o veículo se afaste. Assim, as forças repulsivas traseiras devem ter sentido invertido para serem somadas no ponto de ação frontal. Os coeficientes k_f e k_r representam a taxa de ação das forças frontais e traseiras, respectivamente. A força resultante F é dada por:

$$F = F_a + kf \sum_{p_{jx} > 0} F_{fj} - kr \sum_{p_{jx} < 0} F_{rj} , \quad (2.23)$$

em que:

$$kf + kr = 1. \quad (2.24)$$

A velocidade angular ω (em torno de Z_R) e linear v (na direção de X_R) do robô podem ser calculadas fazendo:

$$\begin{bmatrix} f_x \\ f_y \end{bmatrix}_R = \frac{F}{|F|} \quad (2.25)$$

e

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = C \begin{bmatrix} f_x \\ f_y \\ xf \end{bmatrix} , \quad (2.26)$$

em que C é um coeficiente para ajuste e limitação de velocidade. Em [18, 19] é possível observar que tipicamente $kr > kf$ quando $xf > xr$, e os coeficientes devem ser ajustados para que as forças frontais e traseiras se anulem quando o veículo estiver paralelo a uma parede a uma determinada distância para que ande em linha reta.

É importante comentar que as forças são expressas em relação ao sistema de coordenadas do robô, e, para eliminar a dependência do sistema de coordenadas inercial, a posição da cadeira e a posição de destino podem ser dadas em relação a R. A posição atual do robô será sempre $[X, Y, \Theta]_R^T = [0, 0, 0]_R^T$, e a posição de destino $[X_G, Y_G, \Theta_G]_R^T$ se altera com o tempo. A Equação (2.19) fica igual a:

$$\begin{aligned} \begin{bmatrix} x'_G \\ y'_G \end{bmatrix}_R &= Rot(-\Theta_R) \begin{bmatrix} X_G - X \\ Y_G - Y \end{bmatrix}_R + Rot(\Theta_G) rf - rf , \\ \begin{bmatrix} x'_G \\ y'_G \end{bmatrix}_R &= \begin{bmatrix} X_G \\ Y_G \end{bmatrix}_R + Rot(\Theta_{G_R}) rf - rf. \end{aligned} \quad (2.27)$$

2.3 Intersecção entre segmentos de reta

Para se obter os pontos q_f e q_r do algoritmo de campos potenciais utilizado, é necessário calcular a intersecção entre o segmento de reta que une o respectivo ponto de ação com o ponto de obstáculo e o contorno do robô. Considera-se dois segmentos de reta, um que liga o ponto P ao ponto Q e outro que liga o ponto R ao ponto S, em que:

$$P = \begin{bmatrix} P_X \\ P_Y \end{bmatrix}, \quad Q = \begin{bmatrix} Q_X \\ Q_Y \end{bmatrix}, \quad R = \begin{bmatrix} R_X \\ R_Y \end{bmatrix}, \quad S = \begin{bmatrix} S_X \\ S_Y \end{bmatrix}, \quad (2.28)$$

e os segmentos são dados por:

$$Segmento1 = P + \alpha (Q - P), \quad \alpha = \{\alpha \in \mathbb{R} \mid 0 \leq \alpha \leq 1\}, \quad (2.29)$$

e

$$Segmento2 = R + \beta (S - R), \quad \beta = \{\beta \in \mathbb{R} \mid 0 \leq \beta \leq 1\}. \quad (2.30)$$

Igualando a Equação (2.29) à Equação (2.30), obtém-se o ponto de intersecção $[x, y]^T$:

$$x = P_X + \alpha (Q_X - P_X) = R_X + \beta (S_X - R_X), \quad (2.31)$$

$$y = P_Y + \alpha (Q_Y - P_Y) = R_Y + \beta (S_Y - R_Y). \quad (2.32)$$

Resolvendo para α :

$$\alpha = \frac{(P_Y - R_Y)(S_X - R_X) - (P_X - R_X)(S_Y - R_Y)}{(Q_X - P_X)(S_Y - R_Y) - (Q_Y - P_Y)(S_X - R_X)}. \quad (2.33)$$

Resolvendo para β :

$$\beta = \frac{(R_X - P_X)(Q_Y - P_Y) - (R_Y - P_Y)(Q_X - P_X)}{(S_Y - R_Y)(Q_X - P_X) - (S_X - R_X)(Q_Y - P_Y)}. \quad (2.34)$$

Para pontos P, Q, R e S quaisquer, a intersecção existe se as equações (2.33) e (2.34) resultam em $0 \leq \alpha \leq 1$ e $0 \leq \beta \leq 1$, e o ponto de intersecção pode ser calculado com a Equação (2.29) ou (2.30).

Capítulo 3

Materiais e Métodos

3.1 Materiais

Aqui serão descritos os componentes escolhidos para o projeto e os *softwares* utilizados para simulação.

3.1.1 Cadeira de rodas motorizada

Para a elaboração do projeto, foram pesquisados os modelos de cadeira de rodas motorizada disponíveis no mercado. Um dos principais fatores utilizados para a escolha foi o peso que a cadeira suporta, que está diretamente relacionado à potência dos motores. Os motores devem ter torque suficiente para serem capazes de mover a cadeira com o cadeirante e os demais equipamentos que serão instalados.

A cadeira de rodas motorizada adquirida para o projeto foi uma Freedom SX fabricada em meados de 2010 (Figura 3.1), capaz de suportar pessoas de até 130kg, segundo o fabricante. A cadeira possui quatro rodas, sendo duas fixas com raio de 20", ligadas à motorização, e duas *caster* com raio de 8", que servem apenas de apoio. A transmissão do movimento dos motores para as rodas fixas é feita por meio de uma correia. Como o raio da roda é maior que o raio do eixo do motor, há uma redução na velocidade angular e aumento do torque nas rodas. A cadeira possui apenas suspensão passiva, provida por molas e pelos pneus infláveis.

São utilizados dois motores de corrente contínua de 24V e 400W com escovas, fornecidos pela Bosch, cada um ligado a uma roda fixa. A cadeira conta com freios eletromagnéticos que podem ser acionados eletricamente ou por uma alavanca. Os freios são de 24V e consomem cerca de 300mA cada. A alimentação é feita por duas baterias de 12V e 24Ah ligadas em série.

A condução da cadeira é realizada por um módulo de comando chamado de Drive S microprocessado (Figura 3.2), situado próximo ao apoio de braço direito. Ele contém um *joystick* para direcionamento da cadeira, um potenciômetro para regular a velocidade, um LED indicador de carga da bateria



Figura 3.1: Cadeira de rodas Freedom SX.

e 3 chaves. Utilizando-se a Figura 3.2 como referência, a chave de baixo liga/desliga o módulo de comando, a chave do meio ativa o sistema turbo e a chave de cima aciona os freios eletromagnéticos. O sistema turbo garante uma aceleração maior aos motores, dando a possibilidade ao cadeirante de empinar a cadeira para sobrepor obstáculos como pequenos desníveis. O Drive S é conectado às baterias, aos motores e aos freios eletromagnéticos, e possui um módulo de potência microprocessado que controla a velocidade dos motores em malha aberta por PWM (*Pulse-Width Modulation*, modulação por largura de pulso).



Figura 3.2: Drive S da cadeira de rodas Freedom SX.

As baterias são carregadas por um carregador de 24V projetado especificamente para as cadeiras de rodas Freedom. Ele possui seleção automática de tensão (de 90V_{AC} a 240V_{AC}) e de frequência (50Hz ou 60Hz). Quando o carregador é conectado ao Drive S, por segurança os motores são automaticamente desligados, evitando que a cadeira se movimente enquanto estiver presa ao carregador. Isso é feito com o uso de um relé, e serve também para impedir que os motores utilizem o carregador como fonte de energia, pois este não é capaz de fornecer a corrente consumida pelos motores. A Figura 3.3 fornece algumas dimensões da cadeira.

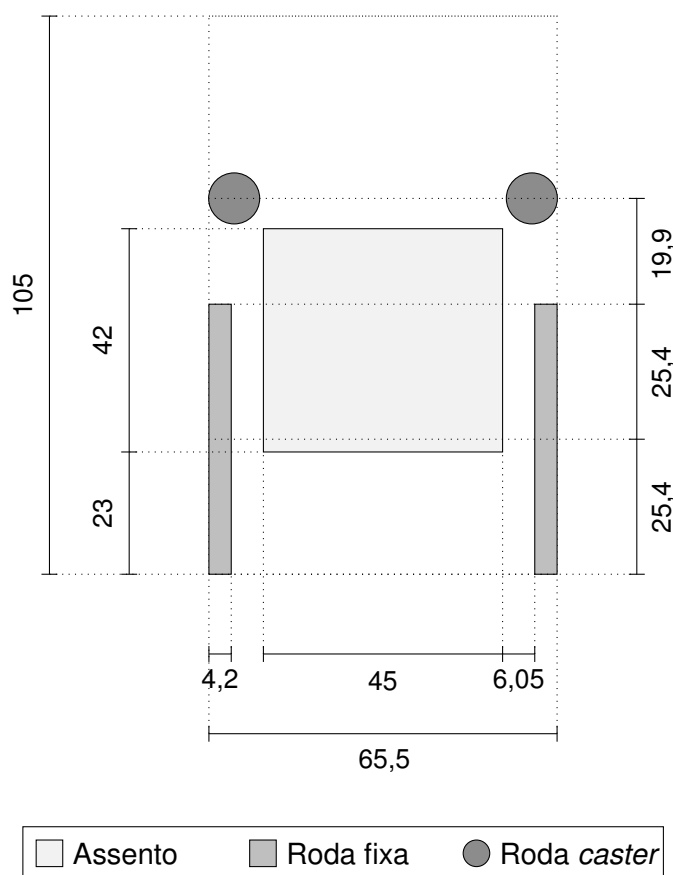


Figura 3.3: Dimensões da cadeira Freedom SX em centímetros.

3.1.2 Controlador

O controlador Roboteq MDC2460 (Figuras 3.4 e 3.5) foi projetado para ser capaz de converter comandos de controle remoto a rádio, *joystick* analógico, PC (via RS-232 ou USB) ou microcomputador para saídas de alta tensão e corrente, alimentando dois motores DC com escovas e controlando suas velocidades por PWM, com operação nos quatro quadrantes [23].

É possível realizar o controle de velocidade dos motores tanto em malha aberta quanto em malha

fechada, pois o controlador possui entrada para dois *encoders* de quadratura e contadores de 32 *bits*. Para um intervalo de tempo entre medições de Δt segundos (período de amostragem), a velocidade de um motor pode ser determinada fazendo:

$$\text{Velocidade angular [rad/s]} = \frac{\text{Contagem} \cdot 2\pi}{\text{Resolução do encoder em CPR} \cdot \Delta t} \quad (3.1)$$



Figura 3.4: Controlador MDC2460: entrada para *encoders*, conectores DB15 e USB.



Figura 3.5: Controlador MDC2460: terminais Faston de 0,25 polegadas.

No projeto da cadeira de rodas autônoma, o controlador será empregado para fazer a interface entre os motores, os *encoders* e o computador (PC). Um controlador PID (*Proportional–Integral–Derivative*,

proporcional integral derivativo) digital com alto tempo de subida pode ser implementado para se obter transições de velocidade mais suaves para o cadeirante. Abaixo são listadas algumas características do controlador:

- Modos de comando por USB, RS-232, 0-5V analógico, ou controle remoto à rádio;
- Aceita tensões de até 60V, com até 60A de corrente em cada canal;
- Opera a partir de uma única fonte de energia;
- Limite de corrente programável, prevenindo a ocorrência de danos aos motores e à bateria;
- Inclui uma linguagem de programação para automação e customização;
- Até 6 entradas analógicas;
- Até 6 entradas de largura de pulso, *duty-cycle* ou frequência;
- Até 6 entradas digitais;
- Entrada para dois *encoders* de quadratura com contadores de 32 *bits*;
- Saída de 5V para alimentar *encoders*;
- LED de diagnóstico;
- Proteção contra sobretensão, subtensão, sobreaquecimento e curto-circuito;
- Detecção de travamento do motor se a corrente ultrapassa o limite estabelecido pelo usuário.

Há dois terminais pelos quais o microcomputador do controlador pode ser energizado. No terminal VMot, a tensão fornecida serve para alimentar tanto os motores quanto o microprocessador. Já o terminal PwrCtrl (*Power Control*) é exclusivo do microcomputador.

A Figura 3.6 ilustra as conexões elétricas recomendadas pelo *datasheet* do componente. Deve-se assegurar que o terra do conector DB15 não esteja ligado ao negativo da bateria. A chave SW1 liga/desliga o controlador: o controlador é ligado quando PwrCtrl está conectado a uma tensão positiva ou quando VMot está ligado a uma tensão positiva e PwrCtrl desconectado (flutuante). O controlador é desligado quando PwrCtrl é conectado ao terra. Sempre que possível, deve-se utilizar uma tensão de 12V ou mais em PwrCtrl para assegurar o funcionamento da lógica do controlador (microcomputador e associados) [24]. A bateria reserva é opcional e serve para garantir a operação do controlador quando a bateria principal está fraca.

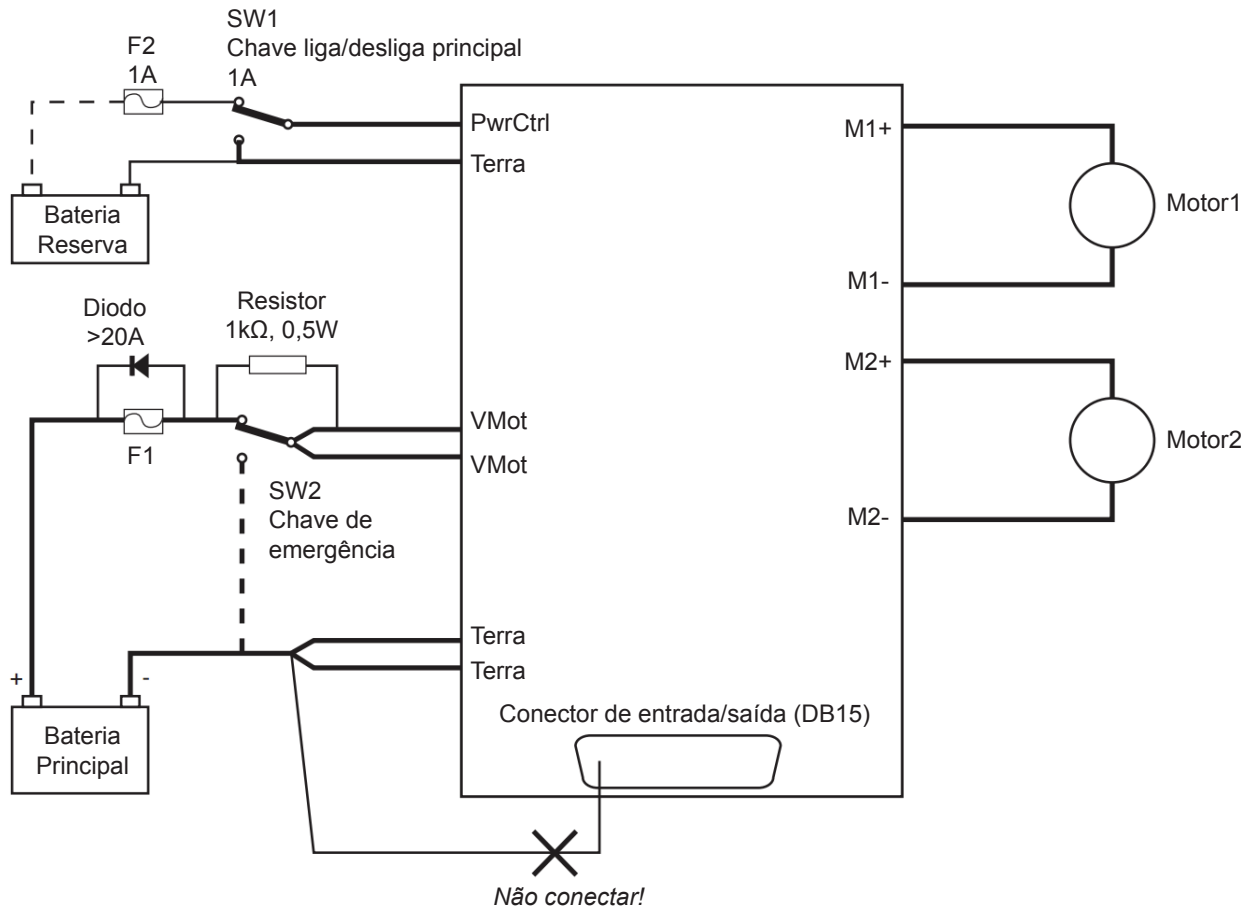


Figura 3.6: Esquema das conexões elétricas do controlador MDC2460. As linhas mais espessas representam conexões obrigatórias. Imagem adaptada de [23].

A chave SW2 serve para cortar a corrente dos motores em caso de emergência, deixando VMot desconectado ou conectado ao terra. O resistor de 1kΩ e 0,5W previne a ocorrência de arco-voltaico. Enquanto o controlador estiver desligado, se houver a possibilidade de um motor girar e produzir uma tensão maior que 60V, o terminal VMot deve ser aterrado.

Caso motores operem como geradores de energia DC (quando o fusível F1 queima ou quando os motores giram a uma velocidade maior do que a tensão aplicada pode proporcionar), um diodo de alta corrente garante um caminho até a bateria para a corrente gerada, recarregando a bateria.

Para aplicações em que uma falha na etapa de saída de energia do controlador, levando os motores ao descontrole, possa ferir pessoas (como é o caso de uma cadeira de rodas autônoma), um contator de segurança (normalmente aberto) deve ser utilizado conforme mostra a Figura 3.7, e PwrCtrl deve ser ligado à bateria para que o controlador possa operar independente da tensão em VMot. O enrolamento do contator deve ser conectado a uma das saídas digitais do conector DB15 (pinos 1 e 9), configurada para ser ativada em caso de "No MOSFET Failure". Internamente, as saídas digitais

são constituídas por transistores MOSFET na configuração de dreno aberto, ou seja, as saídas são aterradas quando ativadas, permitindo a passagem de corrente. As saídas digitais suportam até 40V e 1A e, na configuração padrão, o pino 9 é conectado ao contator e o pino 1 aos freios do motor. Quando uma falha é detectada pelo controlador, ele desativa a saída digital e abre o circuito ligado ao contator. A Figura 3.8 mostra a numeração dos pinos no conector DB15.

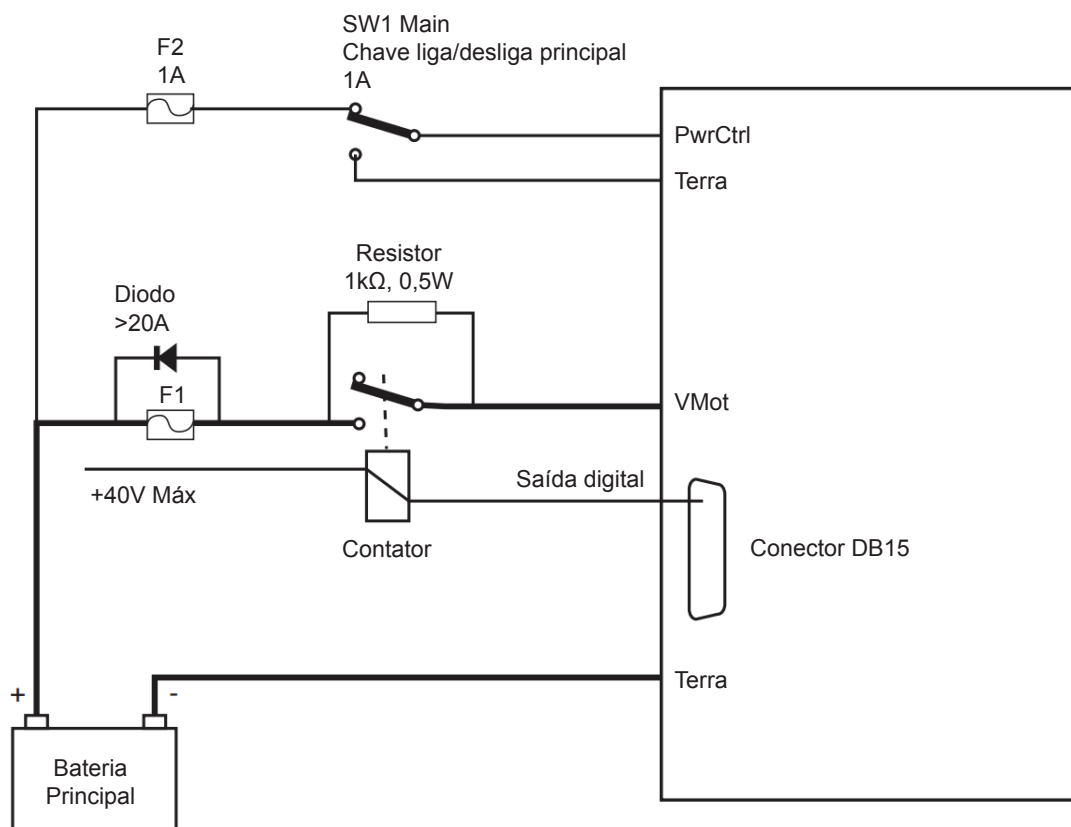


Figura 3.7: Exemplo de esquema das conexões elétricas do controlador MDC2460 com contator de segurança. Imagem adaptada de [23].

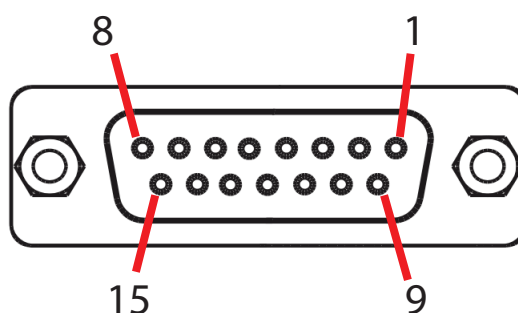


Figura 3.8: Numeração dos pinos do conector DB15. Imagem adaptada de [23].

O adaptador da Figura 3.9 facilita o acesso aos terminais da entrada do controlador para os *enco-*

ders. A Tabela 3.1 explica a função de cada terminal. Segundo o *datasheet*, a corrente máxima total que as saídas de 5V do controlador podem fornecer é de 200mA.

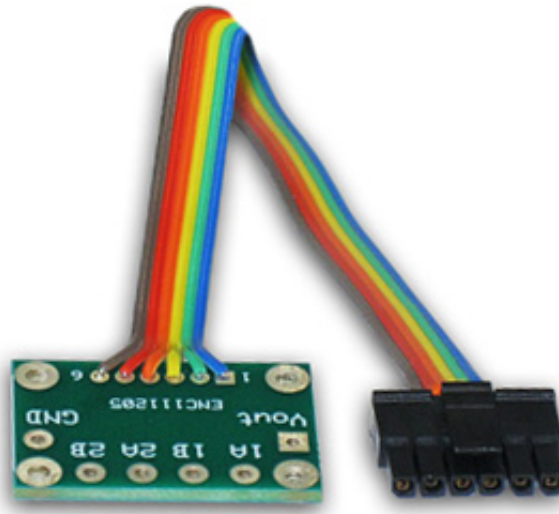


Figura 3.9: Adaptador CABLE-ENC1 para a entrada de *encoder*.¹

Tabela 3.1: Terminais da entrada para *encoder* do controlador MDC2460.

Terminal	Função
5Vout	Saída de 5V para alimentar os <i>encoders</i>
1A	<i>Encoder</i> 1, canal A
1B	<i>Encoder</i> 1, canal B
2A	<i>Encoder</i> 2, canal A
2B	<i>Encoder</i> 2, canal B
GND	Terra

Com o computador, a comunicação RS-232 é preferível em relação à comunicação por USB, pois esta não é confiável em meios ruidosos. O adaptador da Figura 3.10 é utilizado para converter o conector DB15 para um conector de 9 pinos de comunicação RS-232. Ele também permite o acesso individual a 13 dos 15 pinos, e por isso é chamado de *break out board*. Os únicos pinos que não podem ser acessados são o RX e o TX da comunicação serial.

É recomendado que o controlador seja acoplado de modo que sua placa inferior esteja em contato com uma superfície metálica para auxiliar na dissipação de calor. Mais detalhes a respeito do

¹Imagem retirada de <https://www.roboteq.com/index.php/component/virtuemart/201/cable-enc1-detail>. Acesso em 01 de dezembro de 2016.

controlador podem ser encontrados em [24].

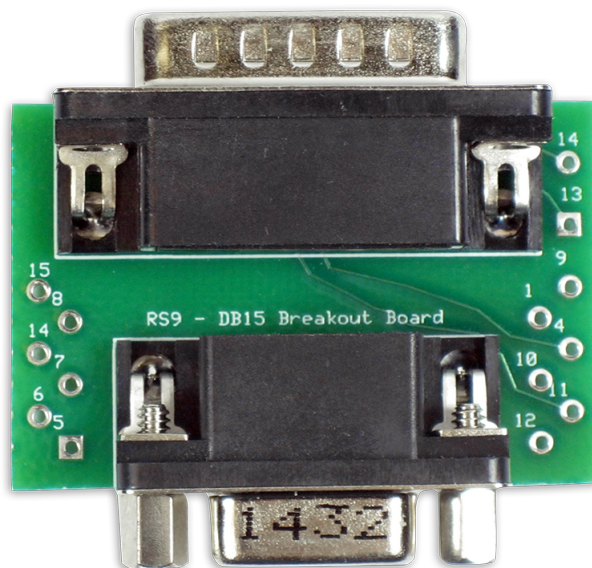


Figura 3.10: Adaptador BOB15.²

3.1.3 Sensores

Para que a cadeira de rodas autônoma obtenha informações sobre si mesma e sobre o ambiente ao seu redor, os sensores descritos abaixo foram empregados no projeto.

Encoder óptico incremental (odômetro óptico)

Encoders são dispositivos cujas funções são medir deslocamento e posição [17]. *Encoders* absolutos fornecem uma medida absoluta de posição em relação a um ponto de referência, enquanto *encoders* incrementais fornecem apenas uma medida de deslocamento, ou seja, de variação de posição. *Encoders* incrementais rotativos podem ser acoplados ao eixo de motores a fim de medir a velocidade angular. Quando aplicados a robôs móveis, é possível estimar o deslocamento do robô a partir do deslocamento angular e do raio de suas rodas. *Encoders* lineares são empregados em equipamentos como impressoras 3D para medir com precisão o deslocamento e a posição do módulo de impressão.

²Imagem retirada de <https://www.roboteq.com/index.php/component/virtuemart/339/bob15-detail>. Acesso em 01 de dezembro de 2016.

Encoders incrementais rotativos ópticos utilizam luz para realizar as medições. Um disco com ranhuras radiais igualmente espaçadas (Figura 3.11) é posicionado entre um fotoemissor (um LED, por exemplo) e um fotorreceptor (um fotodiodo, por exemplo). A saída do fotorreceptor pode ser digitalizada, de modo que se obtenha nível lógico baixo quando o disco bloqueia a passagem de luz, e nível lógico alto quando a luz passa por uma ranhura e alcança o fotorreceptor. Conforme o disco gira, uma onda quadrada é gerada na saída do sistema, e a frequência do sinal é diretamente proporcional à velocidade angular. Alguns *encoders* incrementais possuem um terminal adicional que emite apenas um pulso por revolução, chamado de canal Z ou canal I, e pode ser utilizado como referência para regular as medições. Se acoplado ao eixo de um motor, este tipo de *encoder* permite a medição do valor absoluto da velocidade angular, mas não é possível determinar o sentido de rotação (horário ou anti-horário).

Há também os dispositivos chamados de *encoders* de quadratura, que possuem dois pares fotoemissor/fotorreceptor posicionados de forma específica para gerar os sinais da Figura 3.12. Um fotorreceptor gera o sinal A e o outro o sinal B, e os sinais são defasados de 90° . Analisando ambos os sinais, é possível determinar o sentido de rotação do eixo do motor. Com base na Tabela 3.2, a sequência de estados s1-s2-s3-s4 indica um sentido de rotação oposto à sequência s4-s3-s2-s1. Como o *encoder* de quadratura apresenta 4 estados ao invés de 2 (nível alto e nível baixo), é possível obter uma resolução quatro vezes maior nas medições, considerando uma mesma quantidade de pulsos por revolução.

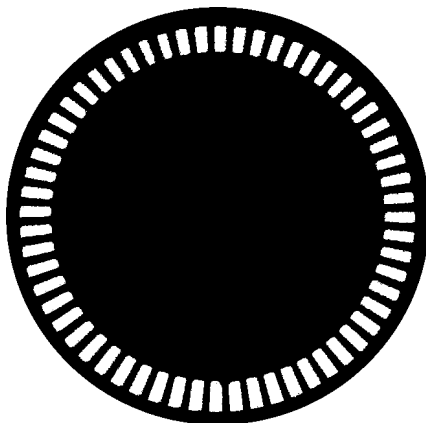
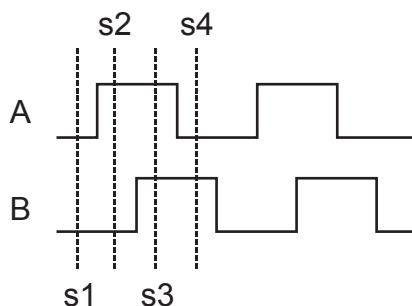


Figura 3.11: Disco de *encoder*.³

Existem diversos tipos de *encoders* disponíveis no mercado. Para o projeto da cadeira de rodas autônoma, foram escolhidos dois *encoders* incrementais modelo HEDS-5540#I06 (Figura 3.13), fabricado pela Avago Technologies, um para cada roda. Este modelo é alimentado com tensão de 5V e

³Imagem retirada de <http://forums.parallax.com/discussion/146228/simple-and-cheap-encoder-idea>. Acesso em 01 de dezembro de 2016.

Figura 3.12: Sinais de um *encoder* de quadratura.Tabela 3.2: Estados de um *encoder* de quadratura.

Estado	Nível do canal A	Nível do canal B
s1	baixo	baixo
s2	alto	baixo
s3	alto	alto
s4	baixo	alto

possui saída digital compatível com circuitos TTL. Além disso, com base na Tabela 3.3, os dois *encoders* juntos consomem uma corrente máxima de 170mA, que é abaixo do máximo que pode ser fornecido pelo controlador (200mA). O dispositivo apresenta eixo oco de 1/4 polegada e resolução de 512 CPR ("Counts Per Revolution", contagens por revolução). Isso significa que cada revolução apresenta 512 mudanças de estado, e o dispositivo tem resolução angular de $360^\circ/512 = 0,703^\circ$. A Tabela 3.4 fornece as condições máximas de operação do dispositivo.

O HEDS-5540#I06 possui 3 canais (A, B e I). Para garantir uma leitura confiável dos valores, o *datasheet* do dispositivo recomenda utilizar um resistor de *pull-up* em cada canal, conforme mostra a Figura 3.14. Nesta configuração, cada canal pode ser ligado à uma carga TTL.

Tabela 3.3: Características elétricas do *encoder* HEDS-5540#I06 para $V_{CC}=5,0V$ a $25^\circ C$.

Parâmetro	Símbolo	Mínimo	Típico	Máximo	Unidade	Notas
Corrente de alimentação	I_{CC}	30	57	85	mA	
Tensão de saída de nível alto	V_{OH}	2,4			V	$I_{OH}=200\mu A$ max
Tensão de saída de nível baixo	V_{OL}			0,4	V	$I_{OL}=3,86mA$



Figura 3.13: Imagem do *encoder* HEDS-5540#I06.⁴

Tabela 3.4: Condições máximas de operação *encoder* HEDS-5540#I06.

Parâmetro	Valor
Tensão de alimentação, V_{CC}	-0,5V até 7V
Tensão de saída	-0,5V até V_{CC}
Corrente de saída por canal	-1,0mA até 5mA
Velocidade	30000 rpm
Aceleração	250000 rad/s ²

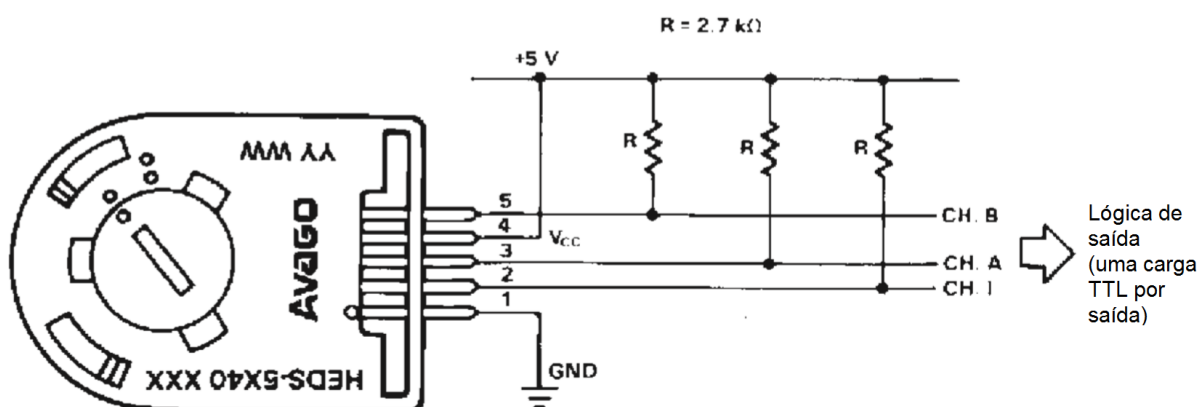


Figura 3.14: Resistores de *pull-up* no *encoder* HEDS-5540#I06. Imagem adaptada do *datasheet* do componente.

⁴Imagem retirada de <http://www.digikey.com/product-search/en?keywords=heds5540%20I06>. Acesso em 01 de dezembro de 2016.

O componente HEDS-8903 (Figura 3.15) será conectado ao *encoder* para permitir o acesso aos terminais por meio de cabos. A Tabela 3.5 informa a função de cada cabo.

Na aplicação de veículos móveis, *encoders* rotativos acoplados às rodas não são suficientes para se determinar com precisão o deslocamento do veículo. Pequenos deslizamentos das rodas, que não são captados pelos sensores, geram incertezas que se acumulam com o passar do tempo, e a estimativa da posição do robô a partir da velocidade das rodas diverge da posição real. Outros tipos de sensores devem ser utilizados em conjunto com os *encoders*, como os sensores LIDAR.

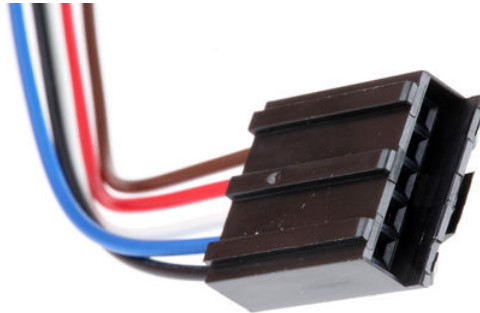


Figura 3.15: Conector HEDS-8903.⁵

Tabela 3.5: Função de cada cabo do conector HEDS-8903.

Pino do <i>encoder</i>	Parâmetro	Cor do cabo
1	Terra (GND)	Preto
2	Canal I	Azul
3	Canal A	Branco
4	Alimentação (VCC)	Vermelho
5	Canal B	Marrom

⁵Imagem retirada de <http://mt.rsdelivers.com/product/broadcom/heds-8903/broadcom-152mm-connector-for-use-with-hedm-5x0x-heds-5xxx/2056904>. Acesso em 01 de dezembro de 2016.

LIDAR

LIDAR (*Light Detection and Ranging*) é uma tecnologia de sensores que utilizam luz para medir distâncias. Um feixe de luz colimada, como um *laser*, é emitido em direção a um objeto alvo, e a luz refletida é captada por um detector sensível às componentes de luz na frequência desejada. A distância do objeto é calculada pela diferença de tempo entre a emissão da luz e a detecção da luz refletida [16].

A Figura 3.16 ilustra o funcionamento deste tipo de sensor. O espelho é utilizado para redirecionar o feixe de luz e pode ser acoplado a um motor, possibilitando a varredura do ambiente e o mapeamento da região ao redor do sensor [17].

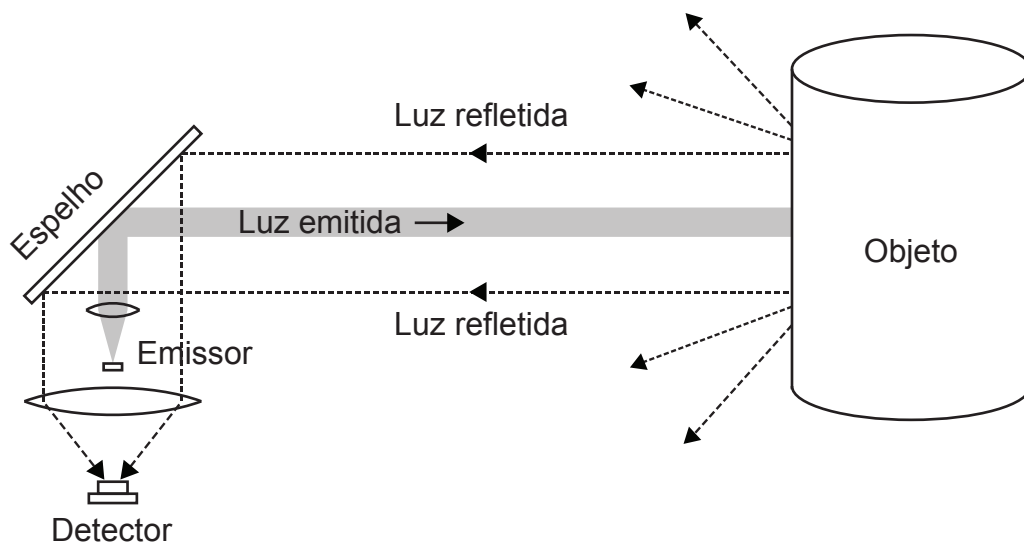


Figura 3.16: Diagrama esquemático do sensor de distância baseado em *laser*.

Por utilizarem luz para realizar as medições, sensores a *laser* são mais rápidos e possuem um custo maior que sensores baseados em vibração sonora, como os sensores de ultrassom. Outras desvantagens dos sensores LIDAR são listadas a seguir:

- Luz externa de alta intensidade (como a luz solar) causa interferência na medição;
- Objetos que refletem pouca luz são mais difíceis de serem detectados. Materiais opticamente transparentes, como o vidro, não são detectados;
- Objetos que possuem reflexão especular, como espelhos, nem sempre refletem a luz em direção ao sensor, e podem não ser detectados.

No projeto, decidiu-se utilizar dois sensores a *laser* que varrem o ambiente em duas dimensões. Estes sensores permitem detectar objetos ao redor da cadeira de rodas e mapear a área, para que a cadeira percorra uma trajetória segura livre de colisões.

Foi feita uma pesquisa dos modelos de sensores disponíveis no mercado, e as informações de alguns deles estão no Apêndice A. O modelo do sensor escolhido foi o UST-10LX (Figura 3.17), fabricado pela Hokuyo, pois possui ângulo de varredura e alcance convenientes, bem como tamanho pequeno e alta velocidade de varredura. Para realizar as medições, é utilizado um *laser* infravermelho classe 1 de comprimento de onda de 905nm, que não apresenta risco à saúde em situação de uso normal [25]. A Tabela 3.6 fornece algumas especificações do sensor.



Figura 3.17: Foto do sensor UST-10LX, fabricado pela Hokuyo.⁶

Tabela 3.6: Características do Hokuyo UST-10LX.

Tensão de alimentação	12V _{DC} / 24V _{DC} (faixa de operação de 10 a 30V, com <i>ripple</i> de 10%)
Corrente de alimentação	Em 24V _{DC} , 150mA ou menos (450mA durante a inicialização)
Fonte de luz	Laser semiconductor (905nm) (classe 1)
Alcance	0,06m até 10m (<i>white Kent sheet</i>) 0,06m até 4m (reflexão difusa de 10%) Máximo: 30m
Ângulo de varredura	270°
Tempo de varredura	25ms/varredura (Velocidade do motor 2400rpm)
Resolução angular	0,25°
Interface	Ethernet 100BASE-TX
Peso	130g (sem o cabo)
Dimensões (largura x profundidade x altura)	50 x 50 x 70 mm (sensor apenas)

⁶Imagem retirada de https://www.hokuyo-aut.jp/02sensor/07scanner/ust_10lx_20lx.html. Acesso em 01 de dezembro de 2016.

Como se pode ver pela Figura 3.18, o sensor realiza 1081 passos de medição em um ângulo de varredura de 270° , o que resulta em uma resolução angular de $0,25^\circ$. Para cada passo de medição, o sensor retorna um valor que corresponde à distância medida em milímetros. Um valor menor que 21 indica erro de medição, e um valor maior que 30000 indica que não foi detectado nenhum objeto ou houve erro de medição. É recomendado que o sensor seja instalado com sua base em contato com um bom dissipador de calor, como uma placa de alumínio de 200mm x 200mm x 2mm, pois o sensor aquece quando está em operação e o calor se concentra em sua base [26].

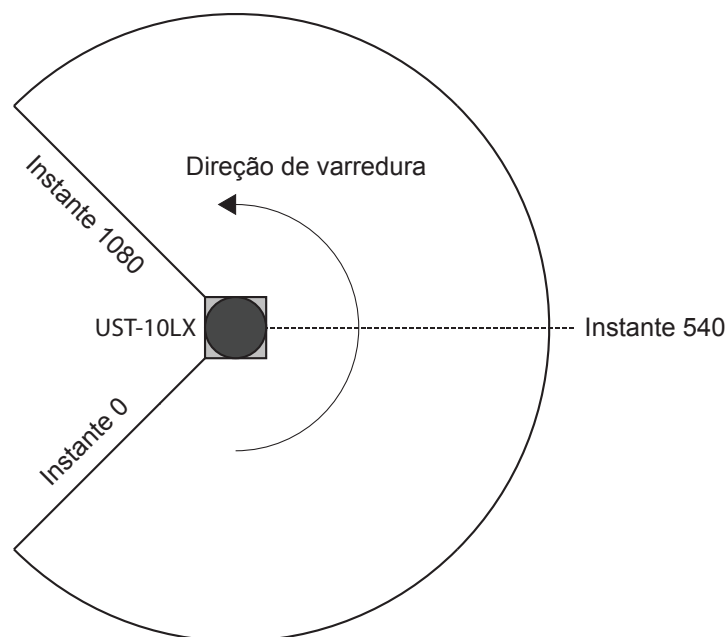


Figura 3.18: Diagrama da varredura do UST-10LX.

Além da interface Ethernet, o dispositivo possui um conector fabricado pela JST, série SH, com 1mm de distância entre os terminais (modelo SHR-06V-S, apresentado na Figura 3.19). São 6 terminais, 4 de entrada/saída e 2 de alimentação, listados na Tabela 3.7. Para utilizar o sensor ligado a um computador, basta fornecer a alimentação contínua e conectar a interface Ethernet. Na comunicação, o computador envia uma mensagem de solicitação ao sensor, e o sensor retorna uma mensagem de resposta. Existem 2 padrões de comunicação: no modo “*handshake*” (“aperto de mãos”), o sensor retorna uma resposta para cada solicitação, enquanto no modo “*continuous*” (“contínuo”), o sensor retorna várias respostas por solicitação [27].



Figura 3.19: O círculo vermelho indica o conector JST do sensor UST-10LX.

Tabela 3.7: Cabos de alimentação e entrada/saída do Hokuyo UST-10LX.

Cor do cabo	Sinal
Vermelho	Entrada COM+
Cinza	Saída COM-
Azul claro	Entrada de IP <i>Reset</i>
Laranja	Saída síncrona
Marrom	+VIN(12VDC/24VDC)
Azul	-VIN

Kinect para Xbox One

O Kinect para Xbox One (Figura 3.20) é um dispositivo fabricado pela Microsoft capaz de capturar imagens em RGB (*Red Green Blue*, imagens coloridas) e imagens de profundidade. Em imagens RGB, cada *pixel* é representado por 3 valores que indicam a intensidade de cor vermelha, verde ou azul naquela região da imagem. Já em imagens de profundidade, cada pixel é representado por um valor que indica a distância entre a câmera e a respectiva região da imagem.



Figura 3.20: Foto do Kinect para Xbox One.⁷

Para capturar imagens de profundidade, o Kinect para Xbox One opera de maneira similar aos sensores LIDAR: diversos feixes de luz infravermelha são emitidos e refletem no ambiente. A imagem de profundidade é criada em função da diferença de tempo entre a emissão da luz e a recepção, pelo detector, da luz refletida. Por este motivo, o Kinect para Xbox One apresenta as mesmas desvantagens dos sensores LIDAR. A Tabela 3.8 contém as especificações do aparelho.

Tabela 3.8: Especificações do Kinect para Xbox One.

Câmera RGB	1920x1080 <i>pixels</i> , 30FPS (15FPS em pouca luz)
Câmera de profundidade	512 x 424 <i>pixels</i> , 30FPS
Distância mínima	0,5m
Distância máxima	4,5m
Campo de visão horizontal	70°
Campo de visão vertical	60°

⁷Imagem retirada de https://www.microsoftstore.com/store/msusa/en_US/pdp/Kinect-Sensor-for-Xbox-One/productID.2267482500. Acesso em 01 de dezembro de 2016.

O equipamento ainda conta com 4 microfones para capturar som e é capaz de identificar não só a fonte do som como também a direção das ondas sonoras [28]. A interface do Kinect para Xbox One é feita por meio de um conector próprio do Kinect e requer alimentação de 12V. Para que possa ser conectado a um computador por uma entrada USB3.0, a Microsoft desenvolveu o Adaptador Kinect para Windows (Figura 3.21).

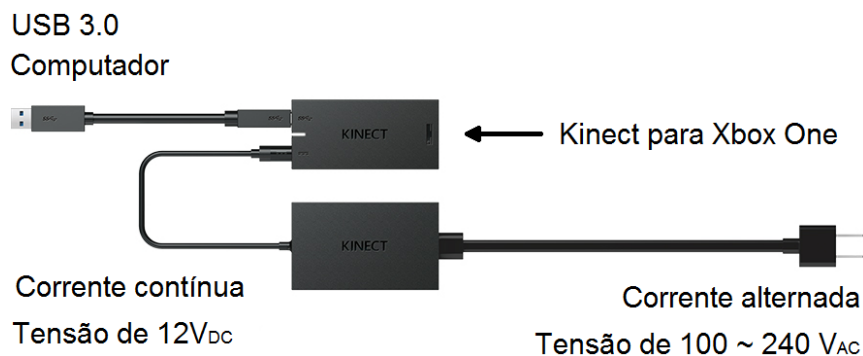


Figura 3.21: Adaptador Kinect para Windows.⁸

Para o projeto da cadeira de rodas autônoma, os 12V de alimentação do Kinect podem ser fornecidos diretamente pela fonte do computador ou por um conversor CC/CC com estabilizador de tensão, ligado às baterias. O fabricante não disponibilizou a informação de consumo de potência, mas a potência pode ser estimada a partir da corrente máxima que o adaptador pode fornecer, que é de 2,67A (potência máxima = $12V \cdot 2,67A = 32,04W$).

Com o SDK ("*Software Development Kit*", kit de desenvolvimento de *software*) disponibilizado pela Microsoft para Windows [29], é possível realizar o rastreamento de até 6 pessoas simultaneamente, cada uma representada por 25 articulações. O Kinect para Xbox One foi escolhido para ser utilizado no projeto da cadeira de rodas autônoma porque ele possibilita a implementação de diversas ações interessantes. Algumas delas são listadas a seguir:

- Reconhecimento facial do acompanhante;
- Reconhecimento de obstáculos no ambiente;
- O acompanhante pode dar comandos de voz para a cadeira ou comandos de sinais com as mãos ou com o corpo;
- Rastreamento do acompanhante, para que a cadeira possa segui-lo.

⁸Imagem adaptada de <http://www.xbox.com/en-US/xbox-one/accessories/kinect/kinect-adapter>. Acesso em 01 de dezembro de 2016.

O *hardware* do Kinect para Xbox One é idêntico ao do Kinect para Windows V2, então os *softwares* desenvolvidos para este são compatíveis com aquele. Existem *drivers* que são *open source* e permitem a utilização do Kinect V2 em sistemas operacionais baseados em Linux [30].

3.1.4 Computador

O computador é responsável por processar as informações dos sensores e determinar a ação da cadeira. A Tabela 3.9 apresenta as especificações do computador escolhido. Optou-se por utilizar um SSD (*Solid-State Drive*) ao invés de um HDD (*Hard Disk Drive*) porque HDDs possuem partes móveis, e o movimento da cadeira poderia ser danoso ao equipamento.

Tabela 3.9: Especificações do computador.

Placa mãe	TF-IMBM-B75A-A10-G2
Processador	Intel i7-3770
Memória RAM	16 GB (2 X 8 GB)
Armazenamento	240 GB SSD 2.5" Sata
Fonte	Fonte automotiva M4-ATX (6-30 V _{DC} , 250 Watts)

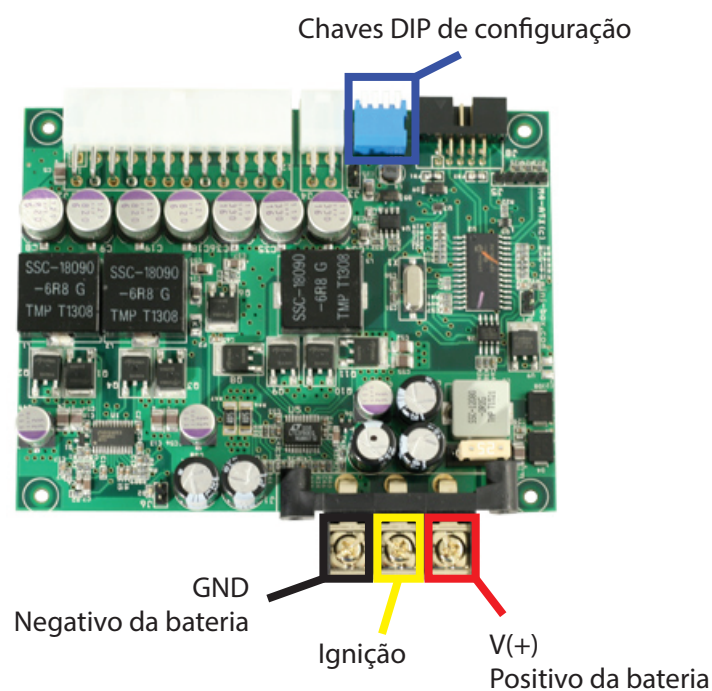
A placa mãe possui duas entradas Ethernet, possibilitando a conexão com os dois sensores a *laser*. Para se comunicar com o controlador, há 5 portas COM que podem ser utilizadas para comunicação serial RS-232. A Tabela 3.10 lista as portas presentes na placa mãe.

Como a cadeira é um veículo móvel que não ficará ligado à rede de distribuição elétrica, a energia para todos os componentes será provida pelas baterias. Desse modo, a fonte M4-ATX (Figura 3.22) do computador é responsável por receber a tensão contínua das baterias e convertê-la para os níveis de tensão aceitos pelos demais componentes do PC.

A fonte M4-ATX foi projetada com o foco na utilização em automóveis com baterias de 12V. No entanto, ela também deve ser capaz de lidar com as variações de tensão que ocorrem na ignição do motor, no descarregamento da bateria e em *load dumps*. Um *load dump* ocorre quando uma carga para qual um gerador está fornecendo corrente é abruptamente desconectada, causando um pico de tensão transitório no gerador [31]. Pensando nisso, a fonte foi projetada de modo que possa operar com alimentação de 6V_{DC} a 30V_{DC}, e circuitos de proteção foram instalados. Para a o projeto da cadeira de rodas autônoma, pretende-se utilizar as duas baterias de 12V da cadeira motorizada ligadas em série como alimentação do computador.

Tabela 3.10: Portas presentes na placa mãe IMBM-B75A.

Armazenamento	SATA x 5 - SATA 6,0 Gb/s x 1 - SATA 3,0 Gb/s x 4
Porta serial	Portas COM (<i>box header</i> x 4, DB-9 externo x 1) COM 1: RS-232/422/485 (DB-9 externo) COM 2-5: RS-232 (<i>box header</i>)
Porta PS/2	Teclado x 1 & Mouse x 1
USB	USB2.0 X 6, USB3.0 X 4
Ethernet	RJ-45 com LED x 2
Interface de expansão	PCI-E [x16] x 1, PCI-E [x4] x 1, PCI x 2, TPM 1.2 <i>onboard</i> (Opcional)

Figura 3.22: Foto da fonte M4-ATX.⁹

A Tabela 3.11 informa quanto de corrente a fonte pode fornecer com uma alimentação de 24V. O nível de tensão 5VSB (5V *StandBy*) está sempre ligado, consumindo energia, até mesmo quando o computador está desligado. Por ele são alimentados os circuitos de disparo da placa-mãe. Por causa da corrente consumida, a bateria irá descarregar mesmo com o computador desligado.

⁹Imagem adaptada de <http://www.mini-box.com/M4-ATX>. Acesso em 01 de dezembro de 2016.

Tabela 3.11: Valores máximos de corrente que podem ser fornecidos pela fonte M4-ATX para cada nível de tensão.

Nível de tensão	Corrente máxima (A)	Corrente de pico (<30 segundos) (A)
5V	15	20
3,3V	15	20
5VSB	1,5	2
-12V	0,15	0,2
12V	12	14

As chaves DIP (ver Figura 3.22) selecionam o modo de operação da fonte. No modo recomendado (chave 1 = *ON*, chave 2 = *OFF*, chave 3 = *OFF*) a linha 5VSB permanece desligada enquanto a ignição está desconectada. Quando a ignição é conectada ao positivo da bateria por alguns segundos, a fonte liga a linha 5VSB e manda um pulso para ligar a placa mãe. O sistema começará a inicializar e o computador permanecerá ligado por pelo menos 60 segundos, para permitir que o sistema operacional seja carregado por completo e prevenir corrupção de disco. Caso a ignição seja desconectada por alguns segundos, a fonte manda um pulso para desligar a placa mãe e, após o computador desligar por completo, todos os níveis de tensão são desligados. Neste modo, a fonte M4-ATX consome menos de 0,5mA [32]. A Figura 3.23 apresenta um diagrama de blocos da fonte.

Caso todas as chaves DIP estejam desligadas, a fonte funciona como uma fonte ATX comum, e o conector de ignição não deve ser utilizado. Para não gastar bateria quando o computador estiver desligado, o contato da fonte com o positivo da bateria pode ser aberto.

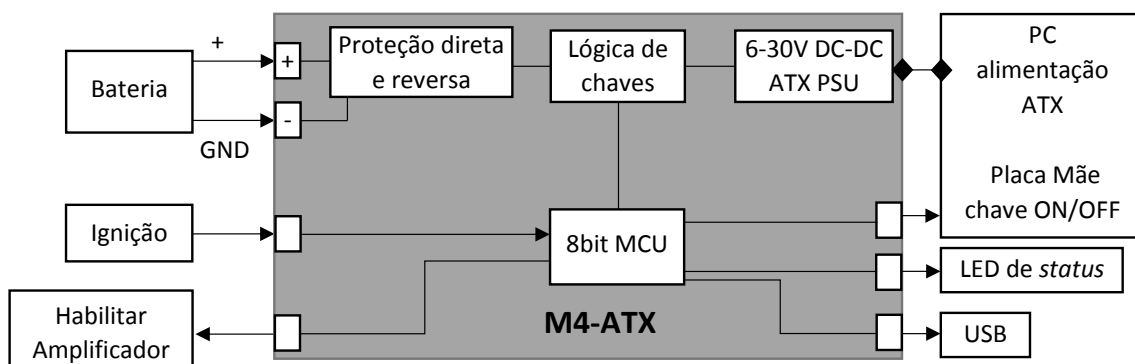


Figura 3.23: Diagrama esquemático da fonte M4-ATX. Imagem adaptada de [32].

3.1.5 Softwares

V-REP

V-REP é um simulador de robôs com diversas funções e APIs (*Application Programming Interfaces*, Interfaces de Programação de Aplicações) elaboradas. Ele é baseado em arquitetura distribuída de controle: cada objeto/modelo pode ser controlado individualmente por um *script*, um *plugin*, um nó do ROS (*Robot Operating System*), um programa utilizando sua API remota, ou uma solução personalizada. Os programas podem ser escritos em C, C++, Python, Java, Lua, Matlab, Octave ou Urbi. [33]

Blender

Blender é um *software* grátis e *open source* para todas as etapas de criação 3D, entre elas a modelagem, a animação e a edição de vídeo. O Blender é multi-plataforma e está disponível para computadores Linux, Windows e Mac. [34]

SFML

A biblioteca SFML (*Simple and Fast Multimedia Library*, Biblioteca Multimídia Simples e Rápida) fornece uma interface simples aos vários componentes do computador para facilitar o desenvolvimento de jogos e aplicações multimídia. SFML é multi-plataforma, o que significa que as aplicações desenvolvidas com ele podem ser executadas em Windows, Linux e Mac OS X. A biblioteca está disponível para diversas linguagens de programação, entre elas C, C++ e C#. [35]

3.2 Métodos

3.2.1 Projeto

O sistema de controle da cadeira de rodas autônoma atuará em níveis. No nível mais alto, o computador capta os comandos do usuário e as informações dos sensores de distância e executa o algoritmo de controle. O computador calcula qual deve ser a velocidade de cada roda e envia os valores para o controlador. No nível intermediário, o controlador serve de interface entre o computador e os motores. O controlador é encarregado de fazer os motores girarem conforme manda o computador, utilizando a leitura dos *encoders* como realimentação para o controle em malha fechada. No nível

mais inferior estão os motores, que apenas recebem a tensão aplicada pelo controlador. A divisão em níveis pode ser vista na Figura 3.24.

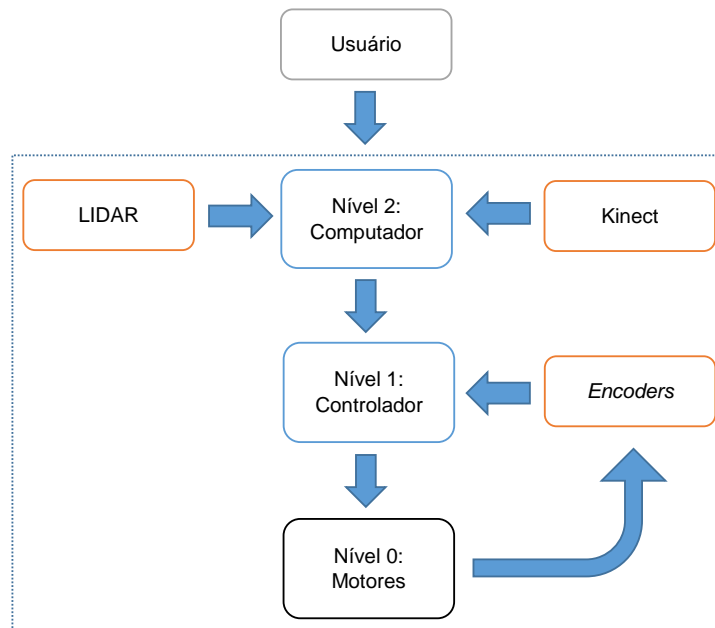


Figura 3.24: Diagrama do sistema de controle.

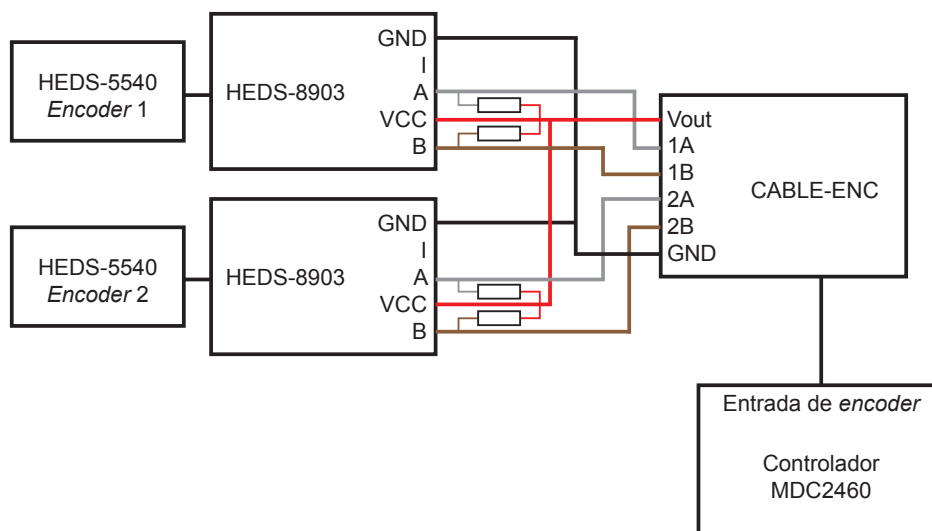


Figura 3.25: Conexão dos *encoders* com o controlador. As cores representam as cores dos cabos dos HEDS-8903.

A conexão dos *encoders* HEDS-5540 com o controlador MDC2460, por intermédio do adaptador CABLE-ENC, é feita conforme mostra a Figura 3.25. Os resistores de *pull-up* de 2,7k Ω são colocados entre os canais A/B e VCC de cada *encoder*, e deve-se verificar se a leitura de ambos os *encoders* é

positiva quando a cadeira anda para frente. O *encoder* que apresentar leitura negativa deve ter seus canais A e B trocados no adaptador CABLE-ENC.

Apesar dos sensores LIDAR serem capazes de operar com tensões de $10V_{DC}$ até $30V_{DC}$, não é recomendado que eles sejam conectados diretamente na bateria. Um circuito de proteção deve ser utilizado para garantir que os sensores não sejam danificados caso ocorram anomalias na bateria (como *load dump*). A fonte M4-ATX possui uma linha de tensão de 12V que pode ser utilizada para alimentar os sensores LIDAR e o Kinect, no entanto deve-se observar o limite de corrente da fonte, que não pode ultrapassar 12A na linha de 12V. A Figura 3.26 mostra a conexão do computador com os sensores caso a fonte M4-ATX seja capaz de alimentá-los. Caso contrário, um conversor CC/CC com estabilizador de tensão deverá ser utilizado para converter a tensão da bateria para os sensores (outra fonte M4-ATX pode ser empregada). Sabe-se que, durante a inicialização, cada sensor LIDAR consome cerca de 1A em 12V, e que o Kinect consome uma corrente máxima de 2,67A, ou seja, a fonte deve ter em torno de 5A disponíveis na linha de 12V para ser capaz de alimentar estes sensores.

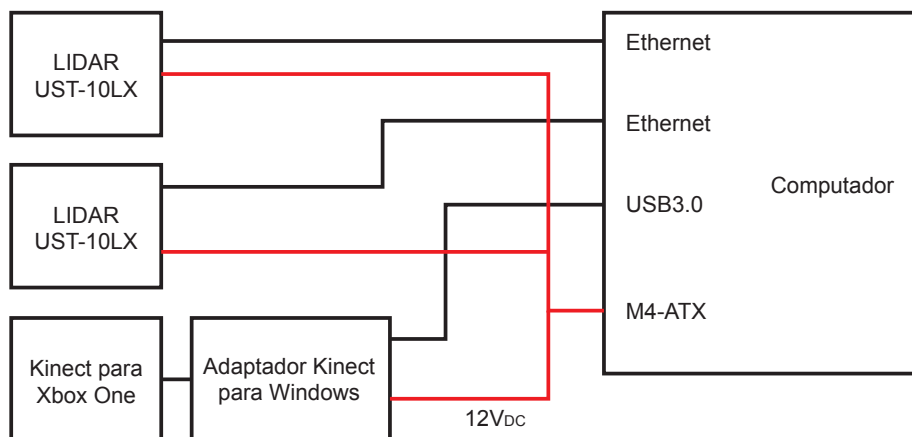


Figura 3.26: Esquema das ligações do computador com os sensores. As ligações vermelhas representam a alimentação, enquanto as pretas são conexões entre os componentes.

As ligações do controlador são mostradas na Figura 3.27. A corrente nominal consumida por cada motor é de 16,7A ($400W/24V$), então é esperada uma corrente máxima de 33,4A no terminal VMot. O contator a ser utilizado deve ser de corrente contínua e ser capaz de suportar os 33,4A de VMot com uma margem de segurança, a fim de evitar a fusão de seus terminais, e a corrente em seu enrolamento de acionamento não deve ser superior a 1A por causa da limitação da saída digital do controlador. A chave de emergência serve para cortar a corrente do enrolamento do contator, fazendo com que o circuito de VMot seja aberto. Quando uma carga indutiva (como os enrolamentos do contator ou dos freios eletromagnéticos) é conectada a uma saída digital, é recomendado utilizar um diodo de roda livre em paralelo à carga [24], e por isso foram empregados os diodos D2 e D3. A tensão de ruptura

dos diodos deve ser maior do que as baterias podem fornecer. Para o esquema apresentado na Figura 3.27, a corrente máxima estimada nos fios que se ligam aos terminais positivo e negativo do conjunto de baterias é de 36,4A (33,4A de VMot + 1A de PwrCtrl + 2A das saídas digitais). A saída digital do pino 1 do conector DB15 é responsável por ativar os freios eletromagnéticos, e a corrente consumida não ultrapassa 1A. A espessura dos fios foi determinada com base nas normas em [36] para eletrodutos com 2 condutores de cobre carregados.

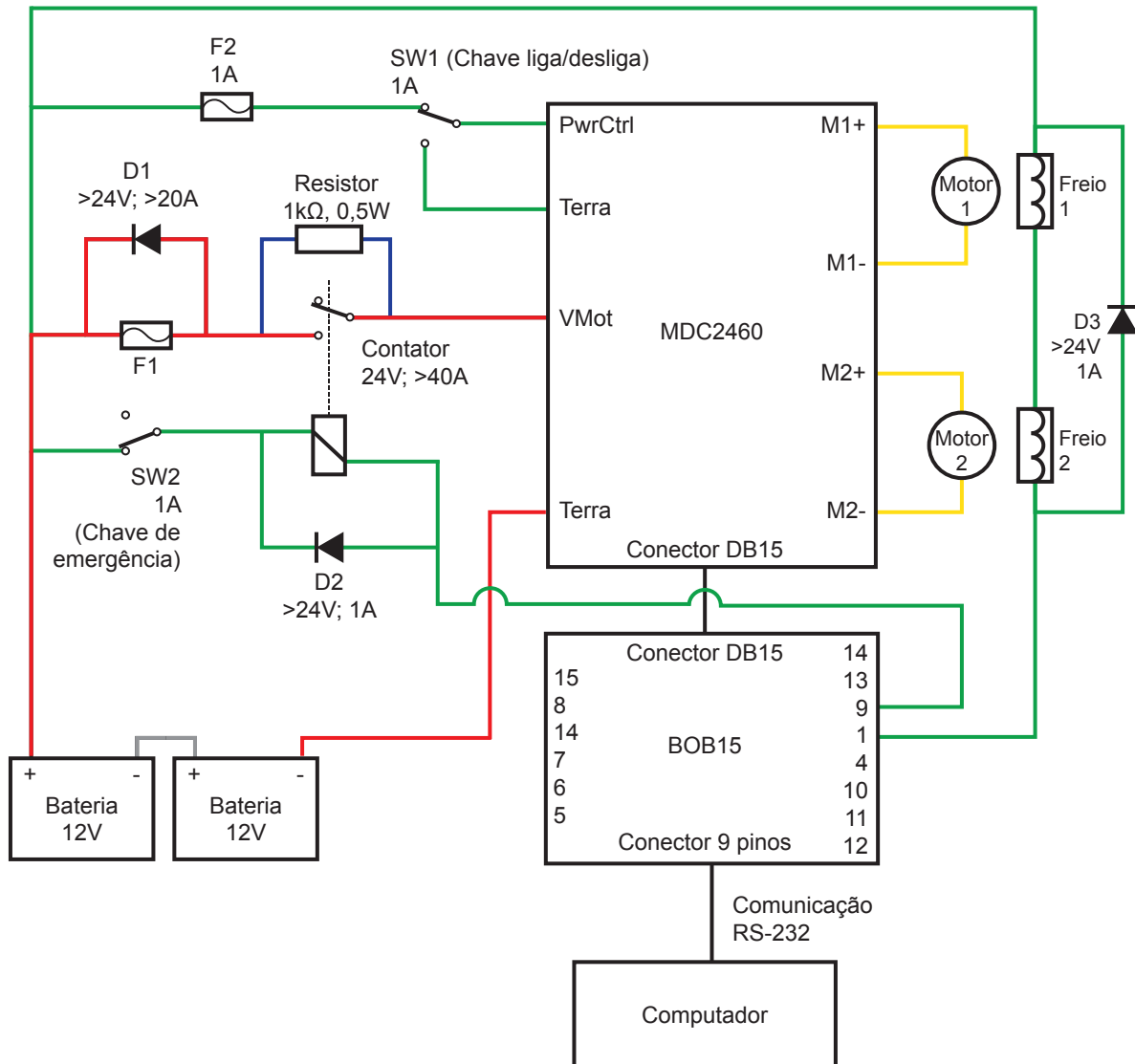


Figura 3.27: Esquema das ligações do controlador MDC2460. Cada cor representa uma espessura diferente de fio. Cinza = 16mm^2 (76A), vermelho = 10mm^2 (57A), amarelo = 4mm^2 (32A), verde = azul = $1,5\text{mm}^2$ (17,5A, menor espessura disponível em [36]). Em preto são as conexões entre componentes.

O transitório da corrente de partida dos motores não foi levado em consideração na escolha dos fusíveis. O controlador possui filtros de ruído e, por isso, não se sabe o quanto o regime transitório

dos motores afeta a corrente de entrada do controlador. Caso seja necessário, o fusível F1 pode ser substituído.

Estima-se que a fonte M4-ATX consome no máximo cerca de 10,4A (250W/24V). Como ainda não há certeza se apenas uma fonte M4-ATX será suficiente, a espessura do fio que faz a ligação em série das baterias deve ser determinada de modo que seja possível conduzir 57,2A (2 fontes M4-ATX + controlador e associados). A Figura 3.28 ilustra as ligações do computador com a bateria.

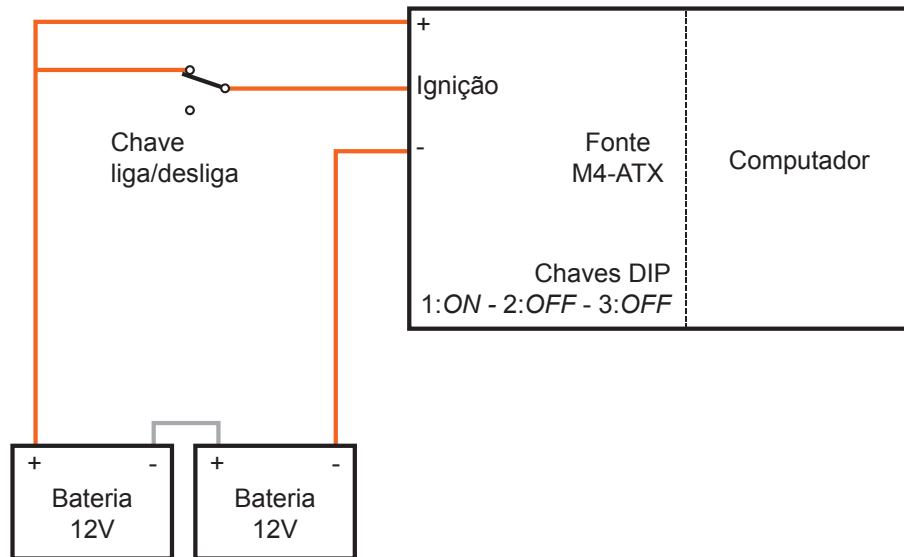


Figura 3.28: Esquema das ligações do computador com a bateria. Cada cor representa uma espessura diferente de fio. Cinza = 16mm² (76A), laranja = 2,5mm² (24A).

Semelhante ao que é feito no Drive S, um relé pode ser utilizado no terminal positivo do conjunto de baterias a fim de abrir o circuito quando o carregador for conectado. O posicionamento dos sensores LIDAR foi decidido conforme os resultados das simulações.

3.2.2 Simulações

Equação final de cinemática inversa

Para poder realizar as simulações, inicialmente foi obtida a equação de cinemática inversa da cadeira de rodas. Fixando a origem do sistema de coordenadas do robô no ponto médio entre as rodas, como mostra a Figura 3.29, e utilizando as informações da Figura 3.3, é possível atribuir os valores apresentados na Tabela 3.12 às constantes da Equação (2.15) e obter sua forma final:

$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix} = \begin{bmatrix} \frac{\dot{x}_R + 0,3065 \dot{\Theta}}{0,254} \\ \frac{\dot{x}_R - 0,3065 \dot{\Theta}}{0,254} \end{bmatrix}. \quad (3.2)$$

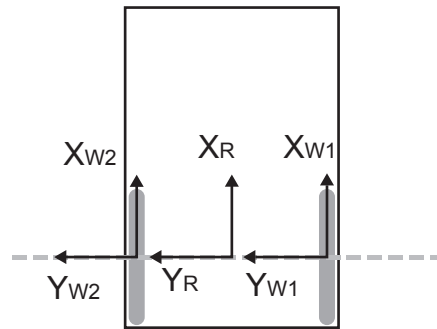


Figura 3.29: Posição conveniente para os sistemas de coordenadas.

Tabela 3.12: Constantes da equação de cinemática inversa.

Constante	Valor
r_1	0,254 m
r_2	0,254 m
$ r_{RW1} $	0,3065 m
$ r_{RW2} $	0,3065 m
α_1	$-\pi/2$
β_1	$\pi/2$
α_2	$\pi/2$
β_2	$-\pi/2$

A velocidade \dot{y}_R não influencia na velocidade das rodas, evidenciando que o robô não possui movimento lateral instantâneo. O algoritmo de campos potenciais apresentado na Seção 2.2 fica en-

carregado de calcular as velocidades \dot{x}_R e $\dot{\Theta}$ desejadas, e a Equação (3.2) determina qual a velocidade que cada roda deve girar.

Modelo da cadeira de rodas no V-REP

Um modelo simplificado em três dimensões da cadeira de rodas foi criado com o *software* Blender para ser importado pelo V-REP. Como mostra a Figura 3.30, as rodas *caster* de apoio foram substituídas por rodas esféricas, pois estas são mais fáceis de implementar na simulação e não alteram a modelagem do veículo.

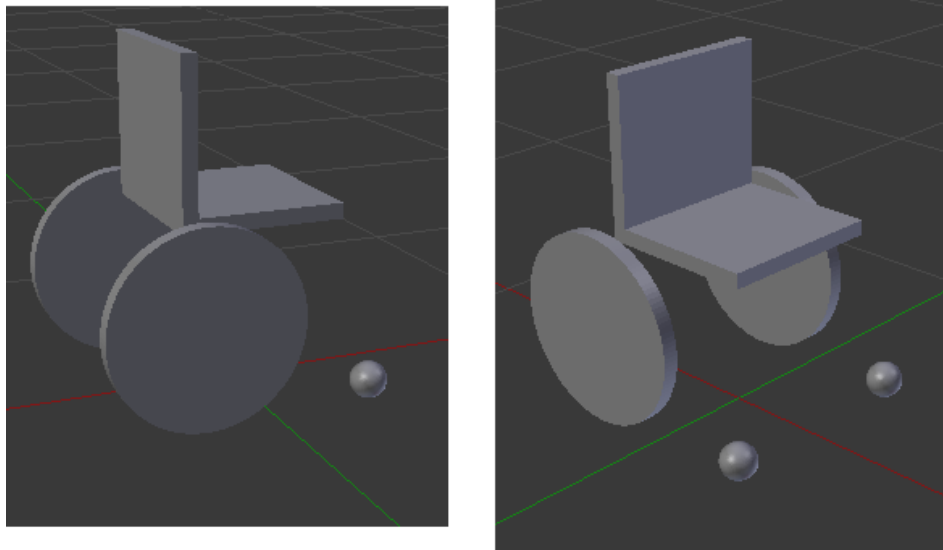


Figura 3.30: Modelo simplificado da cadeira de rodas modelado no Blender.

A importação do modelo no V-REP foi realizada seguindo as instruções em [37]. Quatro *joints* (articulações ou atuadores) foram empregadas, duas de revolução (uma para cada roda fixa) e duas esféricas (uma para cada roda de apoio). A hierarquia dos componentes do modelo ficou estabelecida como mostra a Figura 3.31, e a Tabela 3.13 explica a função de cada componente.

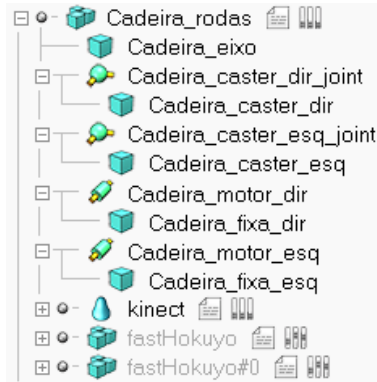


Figura 3.31: Hierarquia dos componentes do modelo da cadeira de rodas no V-REP.

Tabela 3.13: Componentes do modelo da cadeira de rodas no V-REP.

Objeto	Função
Cadeira_rodas	Componente base do modelo. Representa o assento da acadeira com o encosto.
Cadeira_eixo	Cubo de tamanho insignificante posicionado no ponto médio entre os eixos das rodas. Representa o eixo de coordenadas fixo no robô, e é utilizado para fornecer a posição e orientação da cadeira em relação ao eixo de coordenadas global.
Cadeira_caster_dir_joint	Articulação da roda de apoio direita.
Cadeira_caster_dir	Roda de apoio direita, representada por uma roda esférica.
Cadeira_caster_esq_joint	Articulação da roda de apoio esquerda.
Cadeira_caster_esq	Roda de apoio esquerda, representada por uma roda esférica.
Cadeira_motor_dir	Atuador da roda direita.
Cadeira_fixa_dir	Roda fixa direita.
Cadeira_motor_esq	Atuador da roda esquerda.
Cadeira_fixa_esq	Roda fixa esquerda.
kinect	Modelo do Kinect disponibilizado pelo V-REP. Permite capturar imagens RGB e de profundidade.
fastHokuyo	Modelo de sensor LIDAR fornecido pelo V-REP, utilizado para representar o sensor UST-10LX. É possível configurar o alcance e o ângulo de varredura.
fastHokuyo#0	O mesmo que <code>fastHokuyo</code> .

Por padrão, o V-REP tem à disposição modelos de sensores LIDAR e do Kinect. O Kinect foi posicionado no topo da cadeira, direcionado para frente, como mostra a Figura 3.32. As simulações foram realizadas testando duas posições diferentes para os dois sensores a *laser*: um na frente e outro

atrás (Figura 3.33), e um em cada vértice frontal da cadeira (Figura 3.34). Pela janela de parâmetros do *script* (*script parameters*), os sensores foram configurados para apresentar as mesmas características do Hokuyo UST10-LX, isto é, alcance de 10 metros e ângulo de varredura de 270° .

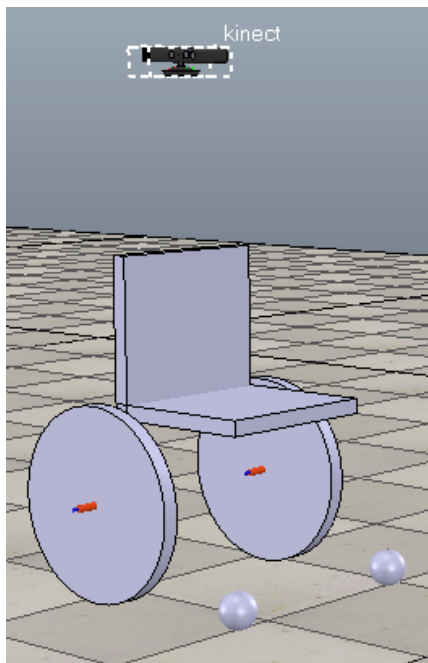


Figura 3.32: Posicionamento do Kinect no modelo.

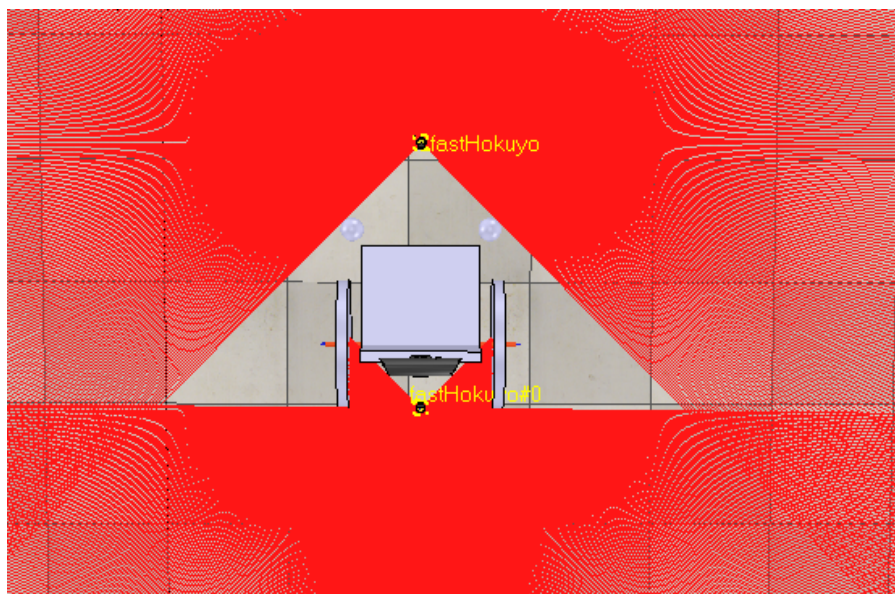


Figura 3.33: *fastHokuyo* posicionado na frente e *fastHokuyo#0* atrás da cadeira. Em vermelho são os feixes de *laser* emitidos pelos sensores.

Para que o V-REP possa se conectar à outro programa utilizando sua API remota, o *script* abaixo

foi adicionado no objeto `Cadeira_rodas`:

```
if (sim_call_type==sim_childscriptcall_initialization) then
    simExtRemoteApiStart(19999)
end
```

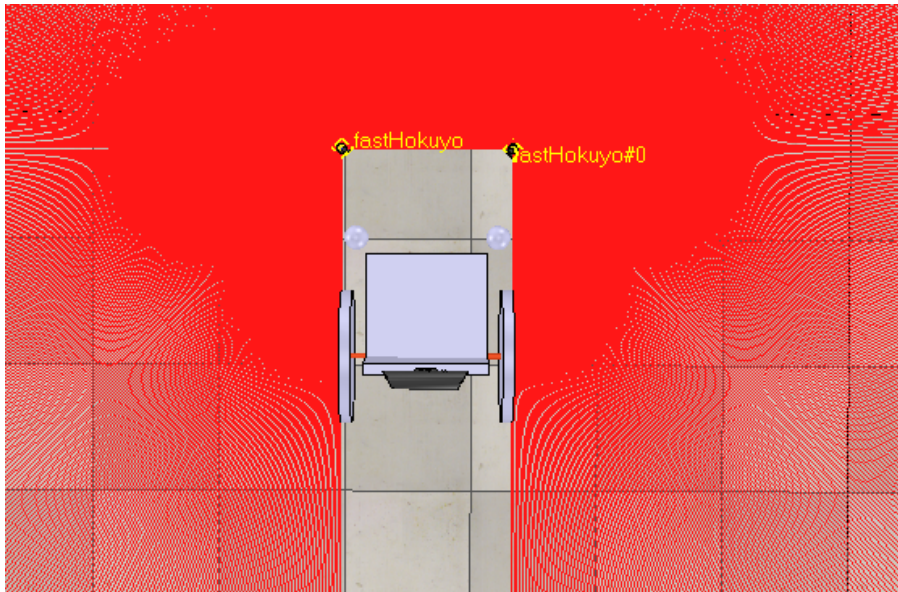


Figura 3.34: `fastHokuyo` posicionado no vértice frontal esquerdo e `fastHokuyo#0` no vértice frontal direito da cadeira. Em vermelho são os feixes de *laser* emitidos pelos sensores.

O modelo `fastHokuyo` disponibilizado pelo V-REP retorna a posição de cada ponto de obstáculo por meio de 3 variáveis *float*, e cada variável representa uma dimensão (X, Y e Z). As componentes em X e em Y são fornecidas considerando o sensor como a origem do sistema de coordenadas, então os dados devem ser manipulados para se obter a posição dos pontos de obstáculo em relação ao sistema de coordenadas fixo no robô. Para que as leituras dos sensores possam ser acessadas por um programa externo, os dados são empacotados e transmitidos por um sinal de *string*. Isso é feito adicionando duas linhas de código no *script* do `fastHokuyo`:

```
-- [...] código já existente
if (sim_call_type==sim_childscriptcall_sensing) then
    -- [...] código já existente
    data=simPackFloats(measuredData)
```

```
simSetStringSignal("measuredDataHokuyo", data)
end
```

Dessa forma, o sinal chamado `measuredDataHokuyo` contém a leitura do `fastHokuyo`. Analogamente, o sinal `measuredDataHokuyo0` contém a leitura do `fastHokuyo#0`.

Na janela de propriedades dinâmicas das articulações (*joint dynamic properties*), `Cadeira_motor_esq` e `Cadeira_motor_dir` devem estar com a opção “*Motor enabled*” selecionada, conforme mostra a Figura 3.35. É possível alterar características específicas dos atuadores clicando em “*Edit engine specific properties*”, mas isso não foi realizado.

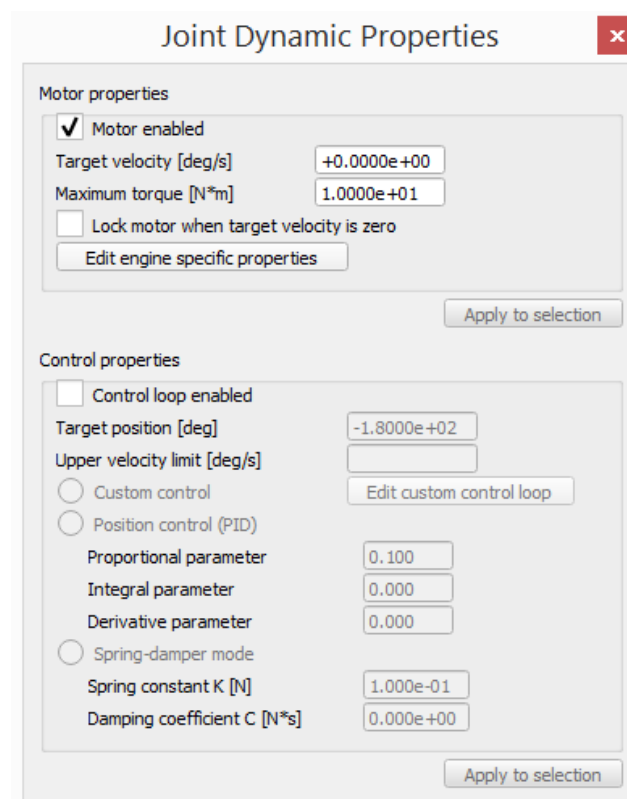


Figura 3.35: Propriedades dinâmicas dos atuadores.

O algoritmo de controle da cadeira foi testado para as duas formas geométricas de corpo (ou contorno) do robô descritas a seguir.

Contorno retangular

O algoritmo de condução funciona exatamente como descrito na Seção 2.2. Para adicionar uma margem de segurança, considerou-se que o robô tem um tamanho maior que o apresentado na Figura 3.3. O tamanho escolhido foi de 120x80cm, o mesmo do módulo de referência utilizado para o estabelecimento das normas em [38].

Na Figura 3.36, os pontos rR_1, \dots, rR_4 são os vértices do contorno real da cadeira, e os pontos r_1, \dots, r_4 são os vértices do contorno virtual utilizado pelo algoritmo de condução, sendo:

$$r_1 = \begin{bmatrix} 0,9 \\ 0,4 \end{bmatrix}_R, \quad r_2 = \begin{bmatrix} -0,3 \\ 0,4 \end{bmatrix}_R, \quad r_3 = \begin{bmatrix} -0,3 \\ -0,4 \end{bmatrix}_R, \quad r_4 = \begin{bmatrix} 0,9 \\ -0,4 \end{bmatrix}_R, \quad (3.3)$$

$$r_f = \begin{bmatrix} 0,9 \\ 0 \end{bmatrix}_R, \quad r_r = \begin{bmatrix} -0,3 \\ 0 \end{bmatrix}_R. \quad (3.4)$$

Os pontos q_f e q_r são calculados pelo algoritmo abaixo, utilizando o método da Seção 2.3.

```

pj = {pjx, pjy}; rf = {xf, 0}; rr = {-xr, 0};
se pjx > 0, então
    se pjx < xf, então
        se pjy > 0, então
            calcular intersecção qf entre os segmentos (r1--r2) e (rf--pj)
        senão
            calcular intersecção qf entre os segmentos (r3--r4) e (rf--pj)
    senão
        qf=rf
senão
    se pjx > -xr, então
        se pjy > 0, então
            calcular intersecção qr entre os segmentos (r1--r2) e (rr--pj)
        senão
            calcular intersecção qr entre os segmentos (r3--r4) e (rr--pj)
    senão
        qr=rr

```

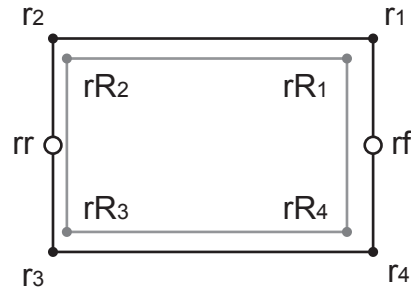


Figura 3.36: Contorno de forma retangular.

Contorno de pentágono irregular

Essa forma é semelhante à anterior, mas há uma protuberância na frente da cadeira a fim de aumentar a margem de segurança para colisões frontais (Figura 3.37). Para isso, adicionou-se o ponto r_5 , dado por:

$$r_5 = \begin{bmatrix} 1,3 \\ 0 \end{bmatrix}_R. \quad (3.5)$$

Os pontos q_f e q_r são calculados pelo algoritmo abaixo.

```

pj = {pjx, pjy}; rf = {xf, 0}; rr = {-xr, 0};
se pjx > 0, então
    se pjx < xf, então
        se pjy > 0, então
            calcular intersecção qf entre os segmentos (r1--r2) e (rf--pj)
        senão
            calcular intersecção qf entre os segmentos (r3--r4) e (rf--pj)
    senão
        se pjy > 0, então
            calcular intersecção qf entre os segmentos (r1--r5) e (rf--pj)
        senão
            calcular intersecção qf entre os segmentos (r5--r4) e (rf--pj)
senão
    se pjx > -xr, então
        se pjy > 0, então

```

```

    calcular intersecção qr entre os segmentos (r1--r2) e (rr--pj)
senão
    calcular intersecção qr entre os segmentos (r3--r4) e (rr--pj)
senão
    qr=rr

```

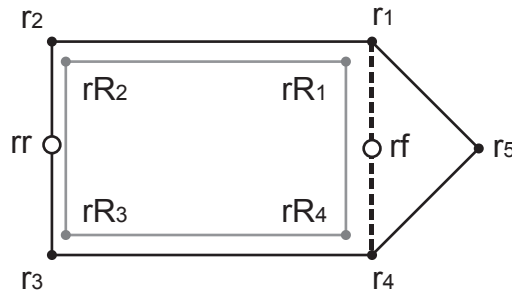


Figura 3.37: Contorno de forma de pentágono irregular.

Modos de operação

Dois tipos de modo de operação foram implementados: autônomo e semi-autônomo. O modo autônomo funciona exatamente como o algoritmo descrito na Seção 2.2, levando a cadeira a um destino enquanto evita obstáculos em função do contorno do robô. No modo semi-autônomo, o usuário controla a cadeira e o algoritmo apenas auxilia na condução para evitar colisões. A intensidade e a orientação da força de atração são controlados pelo cadeirante, enquanto as forças de repulsão são calculadas pelo algoritmo normalmente. Na simulação, as setas do teclado foram utilizadas para indicar a direção e o sentido da força de atração, como especificado abaixo:

- Seta para cima: $Fa = [1, 0]_R^T$;
- Seta para baixo: $Fa = [-1, 0]_R^T$;
- Seta para a direita: $Fa = [0, -1]_R^T$;
- Seta para a esquerda: $Fa = [0, 1]_R^T$.

Quando mais de uma tecla é pressionada ao mesmo tempo, a força de atração resultante é dada pela normalização da soma vetorial das forças individuais, de modo que a norma resultante seja igual a 1.

Como um dos objetivos do projeto é fazer com que a cadeira siga um acompanhante, foi feita uma implementação do método autônomo para que a cadeira siga uma pessoa. Um sistema simples de rastreamento foi utilizado. Neste sistema, inicialmente a cadeira identifica como destino o obstáculo mais próximo de uma região pré-estabelecida (região alvo), nesse caso $[1, 8, 0]^T_R$. Nas iterações seguintes, o novo destino é o ponto de obstáculo mais próximo do destino anterior. Com essa condição, espera-se que no primeiro instante de tempo o acompanhante se encontre próximo a essa região alvo e se torne o destino da cadeira. Desse modo, o acompanhante será seguido automaticamente pela cadeira de rodas. Para isso, tem-se:

$$\begin{bmatrix} X \\ Y \\ \Theta \end{bmatrix}_R = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad (3.6)$$

$$\begin{bmatrix} X_G \\ Y_G \\ \Theta_G \end{bmatrix}_R = \begin{bmatrix} p_{jx_DESTINO} \\ p_{jy_DESTINO} \\ \arctg\left(\frac{p_{jy_DESTINO}}{p_{jx_DESTINO}}\right) \end{bmatrix}. \quad (3.7)$$

A força de atração é calculada com o uso da Equação (2.27).

O V-REP fornece diversos tipos diferentes de modelos de pessoas. O modelo utilizado foi *Bill on path*, no qual a pessoa segue um caminho pré-determinado. A cadeira deve acompanhá-la desde que a condição inicial seja satisfeita e a pessoa seja captada pelos sensores da cadeira. Para não ficar muito próxima do acompanhante, mas também não perdê-lo de vista, a velocidade da cadeira é proporcional ao cubo da distância entre ela e a pessoa.

Ambientes da simulação

As simulações foram realizadas em dois ambientes diferentes. Para o Ambiente 1 (Figura 3.38), realizou-se três tipos de testes:

- Modo autônomo com posições de destino pré-estabelecidas. Inicialmente a cadeira se move até o destino intermediário, e depois para o destino final;
- Modo autônomo seguindo uma pessoa;
- Modo semi-autônomo.

O Ambiente 1 possui um corredor com duas curvas de 90° e, ao final, uma abertura de 1m de largura para uma sala. Esta abertura estreita, que pode ser interpretada como uma porta, serve para

avaliar o comportamento da cadeira em uma situação na qual o cadeirante tem que atravessar uma porta para entrar em um recinto.

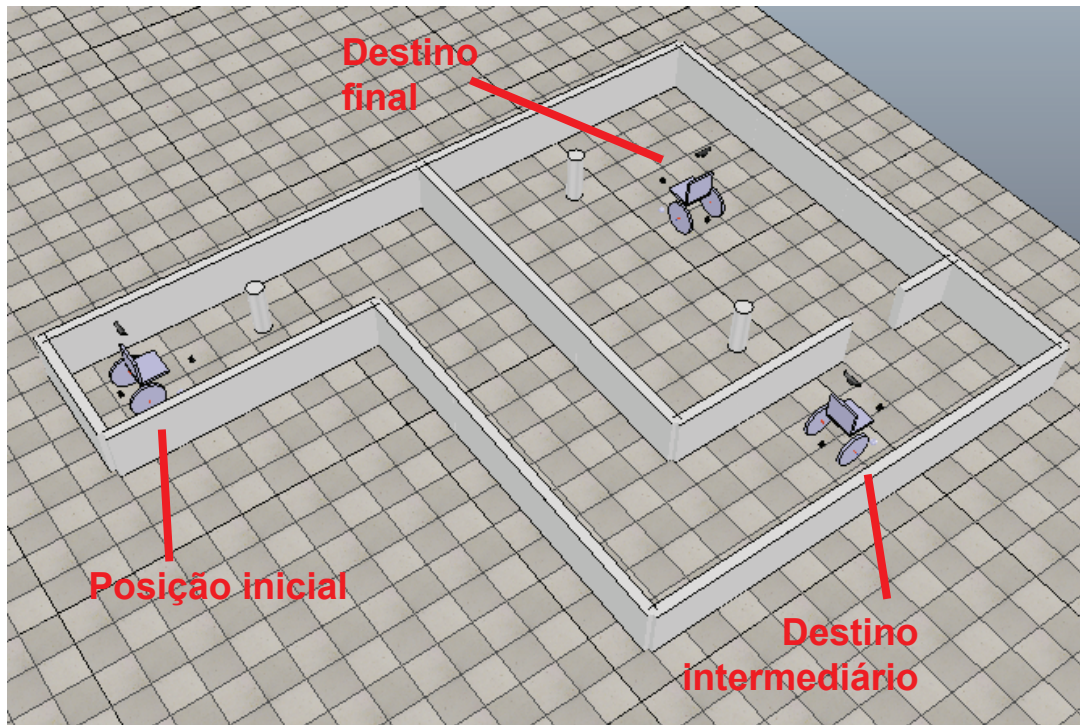


Figura 3.38: Ambiente 1 da simulação.

O Ambiente 2 (Figura 3.39) é composto por uma sala com diversos obstáculos fixos (pilares) e três móveis, representando pessoas andando aleatoriamente. Neste ambiente, apenas o modo autônomo da cadeira de rodas que visa seguir um acompanhante foi testado. O objetivo aqui é verificar se, em um ambiente espaçoso com muitos obstáculos fixos e móveis, a cadeira não falha em rastrear e seguir o acompanhante correto.

Programa em C++ e visualização dos resultados

O programa para controlar a cadeira de rodas no V-REP por meio da API remota foi feito em linguagem C++. A biblioteca SFML foi utilizada para exibir uma janela com uma imagem do mapa 2D das leituras dos sensores de distância e da posição atual da cadeira, para a visualização da evolução da simulação no tempo. A biblioteca também foi utilizada para captar comandos do teclado e transferi-los ao V-REP, como é feito no modo semi-autônomo. A fim de se analisar o comportamento da cadeira, diversos valores para os coeficientes K , k_f e k_r foram testados. A Figura 3.40 ilustra como a manipulação da cadeira de rodas é feita pelo programa em C++.

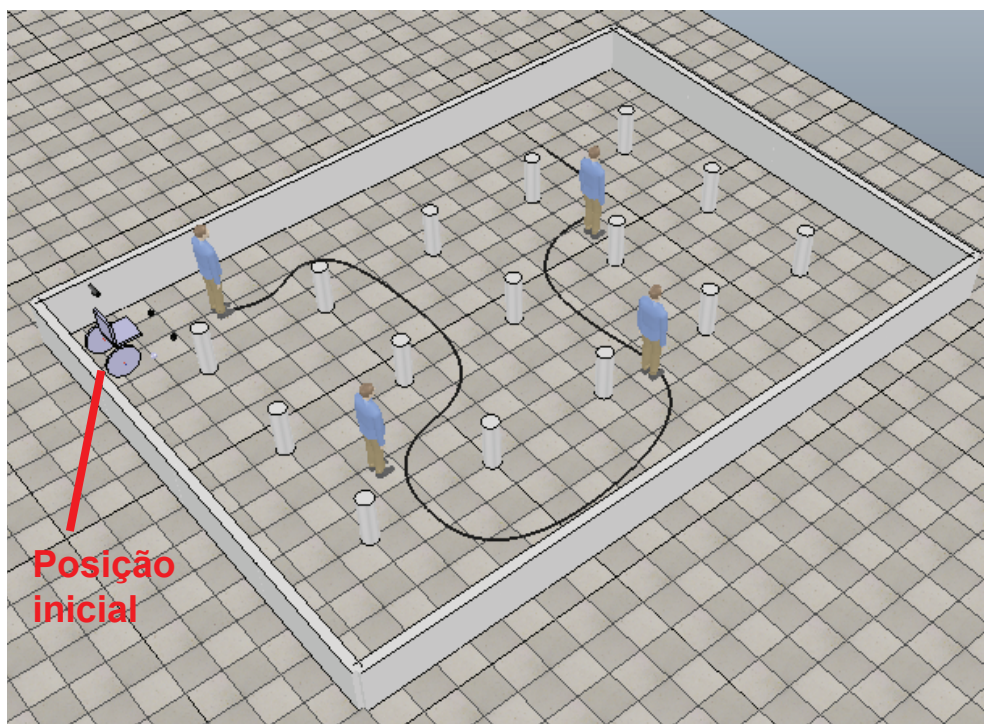


Figura 3.39: Ambiente 2 da simulação.

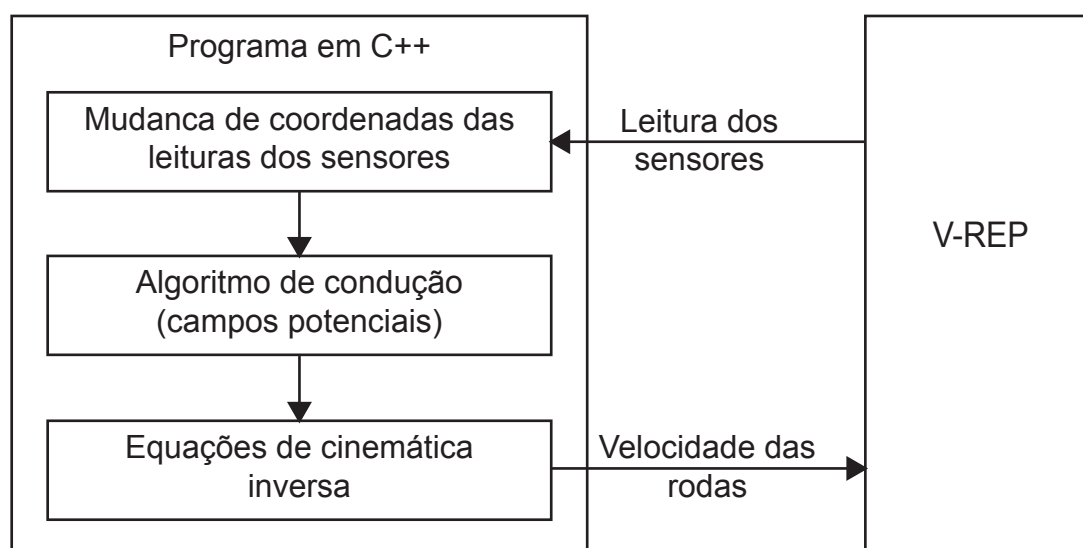


Figura 3.40: Diagrama esquemático da interação do programa em C++ com o V-REP.

Capítulo 4

Resultados

4.1 Simulações

Nos mapas 2D obtidos, a cadeira de rodas é representada por um retângulo preto com seu tamanho real (105x65,5cm). Os quadrados verdes são os destinos, os pontos azuis são obstáculos (obtidos pelas leituras dos sensores) e as setas vermelhas indicam onde ocorreram colisões.

4.1.1 Ambiente 1

Um sensor na frente e outro atrás

- Contorno retangular - modo autônomo com destino pré-estabelecido

Como os sensores não são capazes de detectar obstáculos próximos às laterais da cadeira (ver Figura 3.33), houve colisões nos momentos de realização de curvas (Figura 4.1). Com o algoritmo utilizado, esse posicionamento dos sensores se mostrou perigoso para o cadeirante. Por este motivo, esta configuração dos sensores não foi utilizada nas nas simulações seguintes.

Um sensor em cada vértice frontal

- Contorno retangular - modo autônomo com destino pré-estabelecido

Três testes foram realizados com um sensor em cada vértice frontal da cadeira (ver Figura 3.34) nos quais os coeficientes K , k_f e k_r foram variados. Na simulação da Figura 4.2, a cadeira conseguiu alcançar o destino mas houve uma colisão com a parede ao tentar atravessar a porta. Para evitar que isso ocorra, duas soluções possíveis são aumentar o coeficiente de repulsão K ou alterar o contorno virtual da cadeira.

A simulação da Figura 4.3 foi realizada com um coeficiente de repulsão K maior. A cadeira entrou em um mínimo local e não se moveu mais. O aumento de K faz com que a cadeira se mova em uma

trajetória mais afastada dos obstáculos, no entanto esse aumento também provoca uma ocorrência mais frequente de mínimos locais, o que impede a adoção de um valor elevado de K .

Na simulação da Figura 4.4 foi utilizado o mesmo K da simulação da Figura 4.2, mas com uma proporção maior de kr/kf . As forças de repulsão sobre o ponto de ação traseiro (que dependem de kr) foram mais intensas, o que em determinado momento anulou a força de atração e levou a cadeira a um mínimo local. Com isso, pode-se concluir que mínimos locais ocorrem com mais frequência com valores altos de K e kr .

- Contorno de pentágono irregular - modo autônomo com destino pré-estabelecido

O contorno de pentágono irregular (ver Figura 3.37) garantiu que não houvesse colisões frontais, mas a cadeira não foi capaz de passar pela porta (Figura 4.5). Pelo fato de ser mais seguro, as simulações seguintes foram realizadas utilizando apenas este contorno.

- Contorno de pentágono irregular - modo semi-autônomo

Durante a simulação, o usuário tentou ativamente colidir com as paredes do corredor. Como mostra a Figura 4.6, no modo semi-autônomo o algoritmo foi eficiente para evitar colisões e a cadeira alcançou o destino com sucesso. Uma desvantagem do contorno de pentágono irregular em relação ao contorno retangular é que ele necessita de uma área livre maior para poder girar a cadeira, dificultando a execução de manobras e impossibilitando o movimento em áreas estreitas. Como existe uma região que não é captada pela varredura dos sensores LIDAR, localizada atrás da cadeira (ver Figura 3.34), andar para trás é perigoso e deve ser evitado. No entanto, existem situações em que a única opção disponível para o cadeirante é mover-se para trás (sair de um elevador estreito, por exemplo), então futuramente algum outro sensor deverá ser implementado para cobrir a região traseira da cadeira.

- Contorno de pentágono irregular - modo autônomo seguindo uma pessoa

A linha verde na Figura 4.7 representa a trajetória da pessoa que foi seguida pela cadeira, indicando que o rastreamento do acompanhante foi feito com sucesso enquanto ele se encontrava dentro dos limites de varredura dos sensores. Novamente, a cadeira não foi capaz de atravessar a porta.

4.1.2 Ambiente 2

Um sensor em cada vértice frontal

- Contorno de pentágono irregular - modo autônomo seguindo uma pessoa

O resultado apresentado pela Figura 4.8 indica que a cadeira conseguiu seguir o acompanhante (cuja trajetória é indicada pela linha verde) com sucesso. O rastreamento não foi afetado pelas outras pessoas se movimentando no ambiente (com trajetórias representadas pelos rastros azuis), e não houve colisões. Como os pilares são espaçados, a cadeira não teve dificuldade em passar por entre eles.

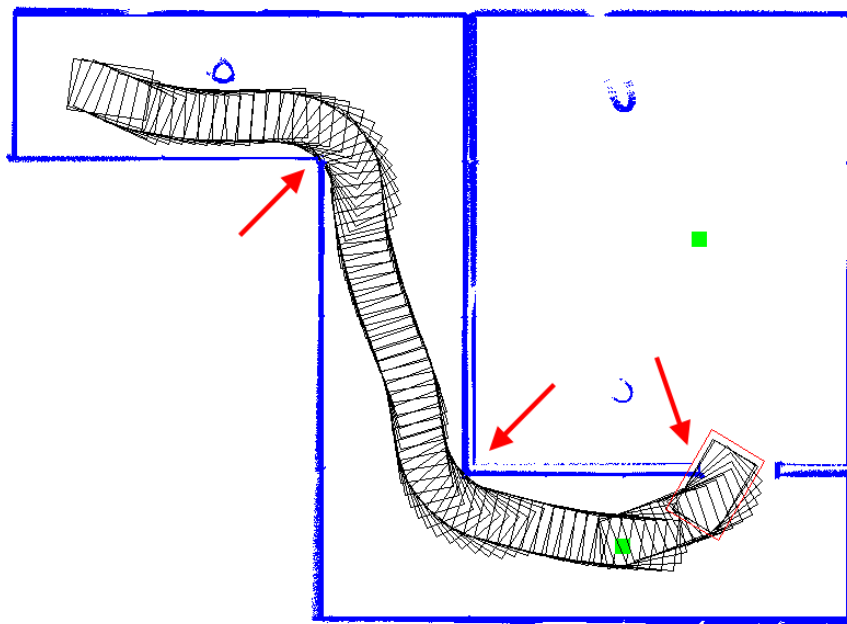


Figura 4.1: Ambiente 1, contorno retangular, um sensor na frente e outro atrás, modo autônomo. $K = 0,00025$, $kf = 0,8$ e $kr = 0,2$. Houve várias colisões.

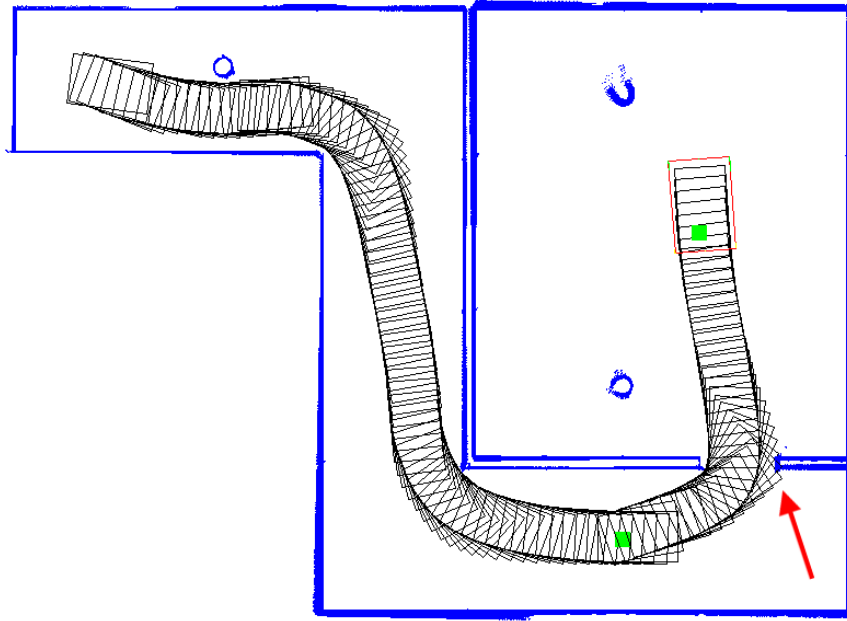


Figura 4.2: Ambiente 1, contorno retangular, um sensor em cada vértice frontal, modo autônomo. $K = 0,00025$, $k_f = 0,8$ e $kr = 0,2$. Foram alteradas as posições dos sensores LIDAR em relação à simulação da Figura 4.1. Houve uma colisão.

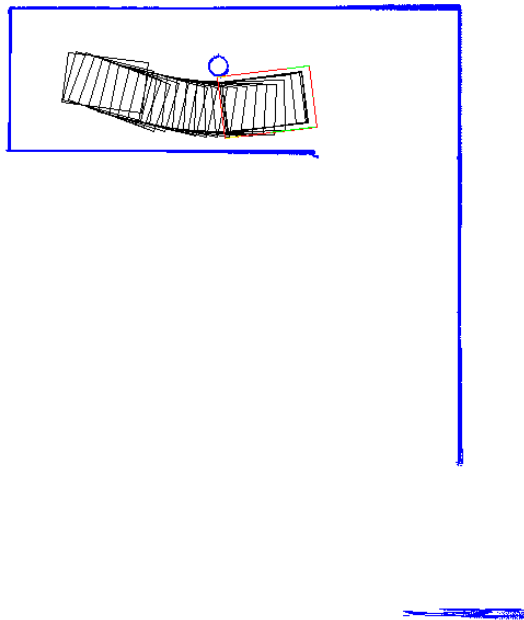


Figura 4.3: Ambiente 1, contorno retangular, um sensor em cada vértice frontal, modo autônomo. $K = 0,0003$, $k_f = 0,8$ e $kr = 0,2$. Foi utilizado um coeficiente de repulsão K maior que o da simulação da Figura 4.2. A cadeira entrou em um mínimo local e não completou o trajeto.

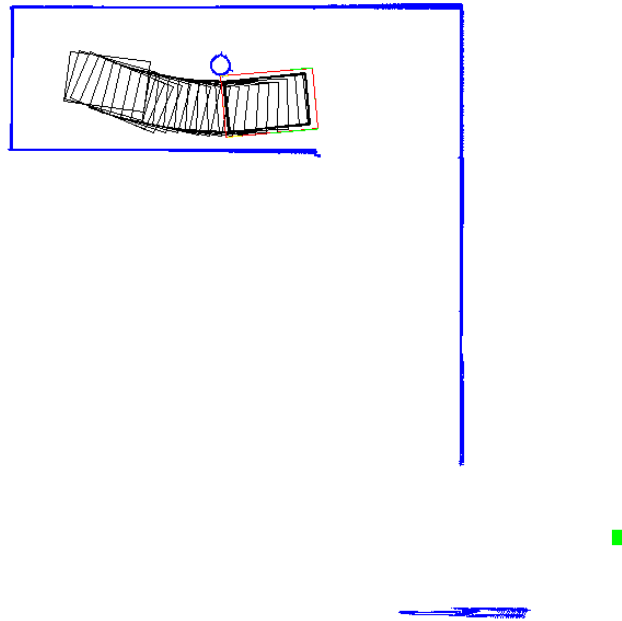


Figura 4.4: Ambiente 1, contorno retangular, um sensor em cada vértice frontal, modo autônomo. $K = 0,00025$, $k_f = 0,6$ e $kr = 0,4$. Foi utilizada uma proporção kr/k_f maior que a da simulação da Figura 4.2. A cadeira entrou em um mínimo local e não completou o trajeto.

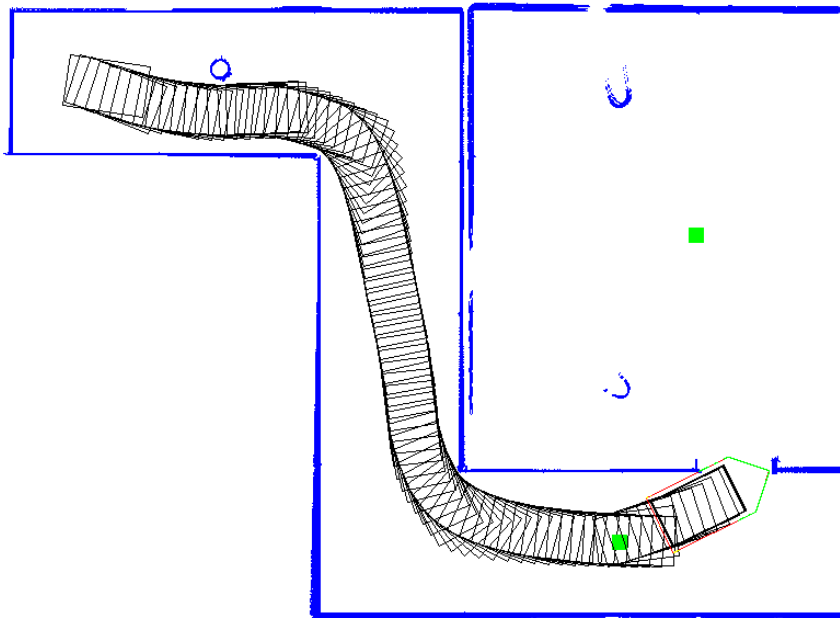


Figura 4.5: Ambiente 1, contorno de pentágono irregular, um sensor em cada vértice frontal, modo autônomo. $K = 0,00015$, $k_f = 0,85$ e $kr = 0,15$. Diferentemente das simulações anteriores, foi empregado o contorno de pentágono irregular. A cadeira entrou em um mínimo local no momento de cruzar a porta.

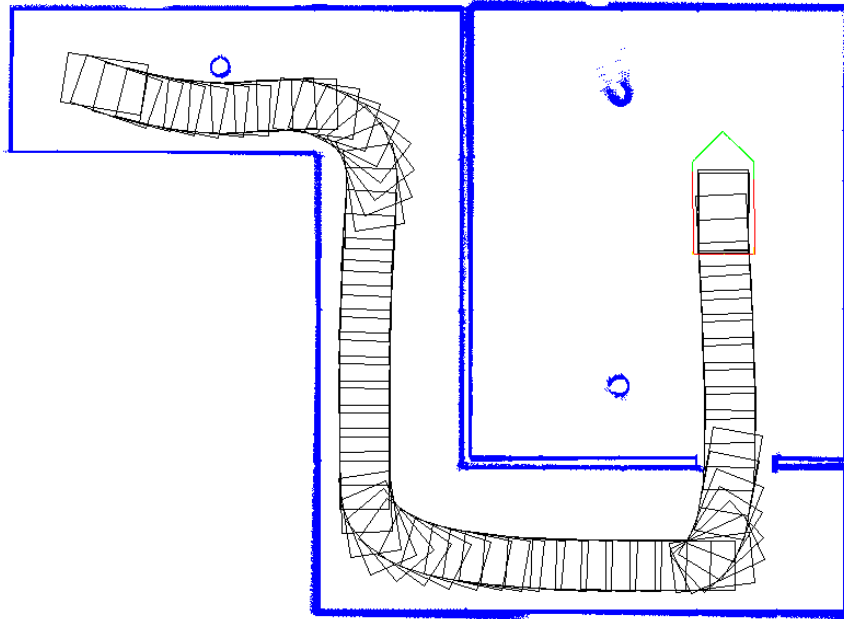


Figura 4.6: Ambiente 1, contorno de pentágono irregular, um sensor em cada vértice frontal, modo semi-autônomo. $K = 0,00015$, $k_f = 0,85$ e $k_r = 0,15$. Com o modo semi-autônomo foi possível cumprir o trajeto.

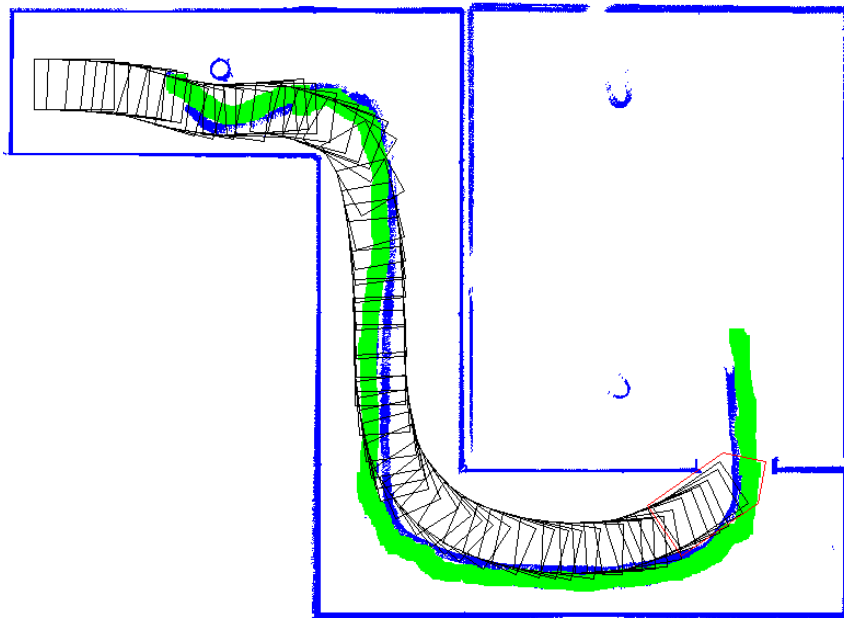


Figura 4.7: Ambiente 1, contorno de pentágono irregular, um sensor em cada vértice frontal, modo autônomo. $K = 0,00015$, $k_f = 0,85$ e $k_r = 0,15$. A cadeira foi capaz de seguir o acompanhante até certo ponto, entrando em um mínimo local no momento de cruzar a porta.

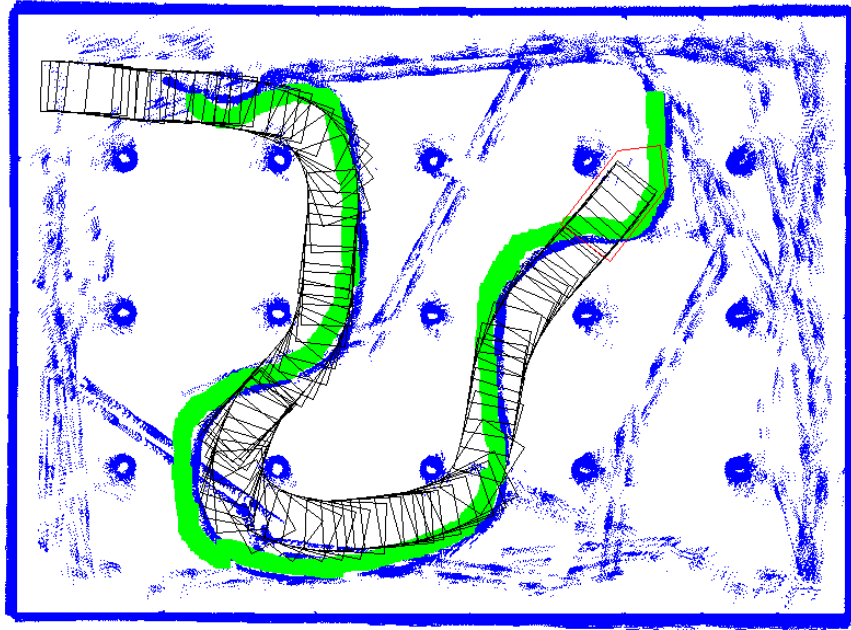


Figura 4.8: Ambiente 1, contorno de pentágono irregular, um sensor em cada vértice frontal, modo autônomo. $K = 0,00015$, $k_f = 0,85$ e $kr = 0,15$. Em um ambiente espaçoso, a cadeira foi capaz de seguir o acompanhante sem problemas.

4.2 Implementação prática

Houve um atraso no procedimento de importação dos componentes, o que impossibilitou a implementação prática do projeto dentro do prazo estabelecido.

Capítulo 5

Conclusão e Perspectivas

5.1 Conclusão

Neste trabalho, elaborou-se uma lista de componentes que devem ser adquiridos para tornar autônoma uma cadeira de rodas motorizada. Para tal, foram pesquisadas as características dos diferentes modelos disponíveis no mercado para cada tipo de componente desejado, e aqueles que se adequaram aos propósitos do projeto foram escolhidos. Levou-se em consideração as especificações e características elétricas de cada componente, a fim de garantir a compatibilidade entre eles.

Para modelar o movimento da cadeira de rodas, estudou-se a cinemática de veículos de acionamento diferencial. Este tipo de veículo possui dois motores independentes, um ligado em cada roda fixa. Por causa das restrições de movimento impostas pelas rodas fixas, o movimento da cadeira é não-holonômico e a trajetória que ela desempenha no espaço é sempre circular com raio variável. Esta restrição de movimentos requer o desenvolvimento de algoritmos específicos para controlar o deslocamento automático da cadeira.

Um desses algoritmos trata-se do algoritmo de campos potenciais para veículos retangulares de acionamento diferencial implementado nas simulações deste projeto. Nesse algoritmo, um robô retangular é repelido por obstáculos e atraído por uma posição de destino, sendo que as forças de atração e repulsão são aplicadas sobre pontos específicos do veículo. A vantagem desse algoritmo é a sua simplicidade, versatilidade de implementação e baixo custo computacional. No entanto, em função da posição do robô, dos obstáculos no ambiente e dos coeficientes do algoritmo, regiões de mínimos locais podem surgir, travando o movimento da cadeira em posições não desejadas.

Para a realização das simulações computacionais, adotou-se um contorno virtual de dimensão maior que o contorno da cadeira real. A implementação dessa “cadeira virtual” se faz necessária para investigar com mais facilidade qual deve ser a melhor disposição dos sensores para que os movimentos sejam realizados com segurança. Observou-se, por exemplo, que uma configuração na qual um sensor LIDAR é posicionado na frente e outro sensor LIDAR atrás da cadeira não é satisfatória

porque as laterais próximas à cadeira não são captadas por esses sensores. Como resultado, a cadeira colide ao realizar curvas em torno de um objeto.

De acordo com o modelo estudado, é mais seguro posicionar os dois sensores LIDAR na frente da cadeira, de modo que um fique à direita e outro à esquerda. Nessa configuração de sensores, as regiões frontais e laterais são amplamente captadas, de modo que é possível não-colidir com obstáculos desde que a cadeira se mova para frente. Contudo, movimentos de ré não são seguros porque, nesta disposição, a varredura dos sensores não capta a região traseira da cadeira, e possíveis obstáculos não são detectados.

O contorno virtual de pentágono irregular adiciona uma protuberância extra na modelagem da cadeira, que serve de margem de segurança adicional contra colisões frontais. A desvantagem deste tipo de contorno é que a área da cadeira é maior e, assim, ela precisa de mais espaço para realizar manobras. O oposto é observado quando o contorno virtual é retangular. Nesta situação, a cadeira consegue manobrar em espaços mais estreitos porque sua área é menor, mas o preço por essa flexibilidade de manobra é a maior exposição a colisões frontais em relação ao contorno de pentágono irregular.

O algoritmo de campos potenciais foi aplicado em duas situações: modo semi-autônomo e modo autônomo. No modo semi-autônomo o usuário toma parte na determinação da trajetória da cadeira. Aqui, o algoritmo se encarrega de auxiliá-lo na locomoção ao identificar obstáculos e evitar que a cadeira se aproxime demasiadamente deles. Consequentemente, o modo semi-autônomo aumenta a segurança do cadeirante.

No modo autônomo, o objetivo principal da cadeira é atingir um ponto específico ou seguir um acompanhante. Nessa situação, o cadeirante se abstém de qualquer controle da cadeira e o algoritmo decide a melhor trajetória a ser percorrida. É interessante notar que, para seguir um acompanhante, a implementação do rastreamento de pessoas possibilitou a distinção do acompanhante correto mesmo em situações em que diversas pessoas diferentes se encontravam andando aleatoriamente em um ambiente com vários obstáculos. Este resultado permite inferir que é possível desenvolver um sistema autônomo com rastreamento de pessoas com baixo custo computacional.

Em suma, o modo semi-autônomo teve um desempenho satisfatório devido à possibilidade de aumentar a segurança de um cadeirante. Embora o modo autônomo tenha cumprido algumas tarefas com sucesso, ele precisa de aperfeiçoamentos para ser utilizado no mundo real.

5.2 Perspectivas

Uma das características do modo autônomo para seguir uma pessoa é a condição inicial, que no mundo real nem sempre é possível de ser cumprida. O Kinect pode ser utilizado para dar comandos à cadeira, seja por voz, sinais de mão ou de corpo. Com ele, o acompanhante tem a possibilidade de informar a cadeira quando o rastreamento pode ser inicializado ou finalizado, garantindo o cumprimento da condição inicial.

Não houve ocorrências de mínimos locais em lugares espaçosos quando a cadeira estava seguindo o acompanhante. Entretanto, o problema recorrente de não ser capaz de atravessar regiões estreitas (como portas) persistiu. Em [39, 40, 41] são propostas alternativas para tratar este tipo de problema, que podem ser implementadas no futuro.

A principal perspectiva, no entanto, é a implementação prática deste projeto. Para isso, é necessário aguardar a entrega de todos os componentes.

Referências Bibliográficas

- [1] IBGE. Censo demográfico 2010. <http://www.ibge.gov.br/home/estatistica/populacao/censo2010/>, Acesso em: 06 de outubro de 2016.
- [2] BRASIL. Lei no 10.098, de 19 de dezembro de 2000. Estabelece normas gerais e critérios básicos para a promoção da acessibilidade das pessoas portadoras de deficiência ou com mobilidade reduzida, e dá outras providências.
- [3] Secretaria especial dos direitos da pessoa com deficiência. <http://www.pessoacomdeficiencia.gov.br/app/>, Acesso em: 16 de outubro de 2016.
- [4] Michael L. Boninger, Rory A. Cooper, Mark A. Baldwin, Sean D. Shimada, and Alicia Koontz. Wheelchair pushrim kinetics: Body weight and median nerve function. *Archives of Physical Medicine and Rehabilitation*, 80(8):910 – 915, 1999.
- [5] Jennifer L. Mercer, Michael Boninger, Alicia Koontz, Dianxu Ren, Trevor Dyson-Hudson, and Rory Cooper. Shoulder joint kinetics and pathology in manual wheelchair users. *Clinical Biomechanics*, 21(8):781 – 789, 2006.
- [6] Michael L. Boninger, Rory A. Cooper, Rick N. Robertson, and Thomas E. Rudy. Wrist biomechanics during two speeds of wheelchair propulsion: An analysis using a local coordinate system. *Archives of Physical Medicine and Rehabilitation*, 78(4):364 – 372, 1997.
- [7] Yoshimasa Sagawa Júnior, Alessandro Haupenthal, Noé Gomes Borges Junior, Daniela Pacheco dos Santos, and Eric Watelain. Análise da propulsão em cadeira de rodas manual: revisão de literatura. *Fisioterapia em Movimento*, 25:185 – 194, 03 2012.
- [8] Matthew A Turk, David G. Morgenthaler, Keith D. Gremban, and Martin Marra. Vits-a vision system for autonomous land vehicle navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):342–361, 1988.
- [9] Dieter Zobel. Trajectory segmentation for the autonomous control of backward motion for truck and trailer. *IEEE Transactions on intelligent transportation systems*, 4(2):59–66, 2003.

- [10] Th Röfer and Axel Lankenau. Architecture and applications of the bremen autonomous wheelchair. *Information Sciences*, 126(1):1–20, 2000.
- [11] Guy Bourhis, Odile Horn, Olivier Habert, and Alain Pruski. An autonomous vehicle for people with motor disabilities. *IEEE Robotics & Automation Magazine*, 8(1):20–28, 2001.
- [12] Bernd Krieg-Brückner, Thomas Röfer, Hans-Otto Carmesin, and Rolf Müller. A taxonomy of spatial knowledge for navigation and its application to the bremen autonomous wheelchair. In *Spatial Cognition*, pages 373–397. Springer, 1998.
- [13] Ulrich Borgolte, Helmut Hoyer, Christian Bühler, Helmut Heck, and Ralf Hoelper. Architectural concepts of a semi-autonomous wheelchair. *Journal of Intelligent and Robotic Systems*, 22(3-4):233–253, 1998.
- [14] Holly A Yanco and James Gips. Preliminary investigation of a semi-autonomous robotic wheelchair directed through electrodes. In *Proceedings of the Rehabilitation Engineering Society of North America 1997 Annual Conference*, pages 414–416, 1997.
- [15] Sarangi P Parikh, Valdir Grassi Jr, Vijay Kumar, and Jun Okamoto Jr. Integrating human inputs with autonomous behaviors on an intelligent wheelchair platform. *IEEE Intelligent Systems*, 22(2):33–41, 2007.
- [16] Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to Autonomous Mobile Robots*. Intelligent Robotics and Autonomous Agents series. The MIT Press, 2nd edition, 2011.
- [17] Roseli Aparecida F. Romero, Edson Prestes, Fernando Osório, and Denis Wolf. *Robótica Móvel*. LTC, 2014.
- [18] Hiroaki Seki, Satoshi Shibayama, Yoshitsugu Kamiya, and Masatoshi Hikizu. Practical obstacle avoidance using potential field for a nonholonomic mobile robot with rectangular body. In *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, pages 326–332. IEEE, 2008.
- [19] Hiroaki Seki, Masatoshi Hikizu, and Yoshitsugu Kamiya. *Real-time obstacle avoidance using potential field for a nonholonomic vehicle*. INTECH Open Access Publisher, 2010.
- [20] Yuki Kinpara, Elly Takano, Yoshinori Kobayashi, and Yoshinori Kuno. Situation-driven control of a robotic wheelchair to follow a caregiver. In *Frontiers of Computer Vision (FCV), 2011 17th Korea-Japan Joint Workshop on*, pages 1–6. IEEE, 2011.

- [21] Yoshinori Kobayashi, Ryota Suzuki, and Yoshinori Kuno. Robotic wheelchair with omni-directional vision for moving alongside a caregiver. In *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*, pages 4177–4182. IEEE, 2012.
- [22] Yoshihisa Sato, Masaya Arai, Ryota Suzuki, Yoshinori Kobayashi, Yoshinori Kuno, Keiichi Yamazaki, and Akiko Yamazaki. A maneuverable robotic wheelchair able to move adaptively with a caregiver by considering the situation. In *2013 IEEE RO-MAN*, pages 282–287. IEEE, 2013.
- [23] Roboteq. Mdc2xxx datasheet. <https://www.roboteq.com/index.php/component/virtuemart/313/mdc2460-274-detail?Itemid=970>, Acesso em: 24 de outubro de 2016.
- [24] Roboteq. Advanced brushed and brushless digital motor controllers user manual. <http://www.roboteq.com/index.php/support/documentation>, Acesso em: 24 de outubro de 2016.
- [25] International Electrotechnical Commission. *Safety of laser products - Part 1: Equipment classification and requirements*. 2007. IEC 60825-1:2007.
- [26] Hokuyo. Scanning laser range finder smart-urg mini ust-10lx (uust003) specification. http://www.hokuyo-aut.jp/02sensor/07scanner/ust_10lx_20lx.html, Acesso em: 20 de outubro de 2016.
- [27] Hokuyo. Communication protocol specification for ust series 10lx/20lx. http://www.hokuyo-aut.jp/02sensor/07scanner/download/products/ust-10lx_20lx/, Acesso em: 22 de outubro de 2016.
- [28] Microsoft. Kinect hardware. <https://developer.microsoft.com/en-us/windows/kinect/hardware>, Acesso em: 23 de outubro de 2016.
- [29] Microsoft. Developing with kinect for windows. <https://developer.microsoft.com/en-us/windows/kinect/develop>, Acesso em: 23 de outubro de 2016.
- [30] Open source drivers for the kinect for windows v2 device. <https://github.com/OpenKinect/libfreenect2>, Acesso em: 23 de outubro de 2016.
- [31] Texas Instruments. Load dump and cranking protection for automotive backlight led power supply, Março 2015. *Application Report SNVA681A*.
- [32] M4-atx 6-30v intelligent atx power supply - installation guide. <http://resources.mini-box.com/online/PWR-M4-ATX/PWR-M4-ATX-manual.pdf>, Acesso em: 20 de outubro de 2016.

- [33] Coppelia Robotics. Coppelia robotics v-rep: Create. compose. simulate. any robot. <http://www.coppeliarobotics.com/>, Acesso em: 31 de outubro de 2016.
- [34] Blender Foundation. blender.org - home of the blender project - free and open 3d creation software. <https://www.blender.org/>, Acesso em: 31 de outubro de 2016.
- [35] Laurent Gomila. Simple and fast multimedia library. <http://www.sfml-dev.org/index.php>, Acesso em: 31 de outubro de 2016.
- [36] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. Nbr 14639:2001. Posto de serviço - Instalações elétricas. Rio de Janeiro: ABNT, 2001.
- [37] Coppelia Robotics. Building a clean model tutorial. <http://www.coppeliarobotics.com/helpFiles/en/buildingAModelTutorial.htm>, Acesso em: 30 de outubro de 2016.
- [38] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. Nbr 9050:2015. Acessibilidade a edificações, mobiliário, espaços e equipamentos urbanos. Rio de Janeiro: ABNT, 2015.
- [39] Liu Chengqing, Marcelo H Ang, Hariharan Krishnan, and Lim Ser Yong. Virtual obstacle concept for local-minimum-recovery in potential-field based navigation. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 983–988. IEEE, 2000.
- [40] Sarangi Patel, Sang-Hack Jung, James P Ostrowski, Rahul Rao, and Camillo J Taylor. Sensor based door navigation for a nonholonomic vehicle. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 3, pages 3081–3086. IEEE, 2002.
- [41] Fernando Auat Cheein, Celso De La Cruz, Ricardo Carelli, and Teodiano F Bastos-Filho. Solution to a door crossing problem for an autonomous wheelchair. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4931–4936. IEEE, 2009.

Apêndice A

Modelos de sensores LIDAR

Tabela A.1: Modelos de sensores LIDAR

Modelo	Alcance	Ângulo de varredura	Resolução	Tempo de varredura	Tensão de alimentação	Interface
Hokuyo URG-04LX-UG01	4m	240°	0,36°	100ms	5V _{DC}	USB
Hokuyo URG-04LX	4m	240°	0,36°	100ms	5V _{DC}	USB, RS-232C
Hokuyo UBG-04LX-F01	4m	240°	0,36°	28ms	12V _{DC}	USB, RS-232C
Hokuyo UTM-30LX	30m	270°	0,25°	25ms	12V _{DC}	USB
Hokuyo UTM-30LX-EW	30m	270°	0,25°	25ms	12V _{DC}	Ethernet
Hokuyo UXM-30LX-EW	30m	190°	0,25°	50ms	10 a 30V _{DC}	Ethernet
Hokuyo UST-10LX	10m	270°	0,25°	25ms	12/24V _{DC}	Ethernet
Hokuyo UST-20LX	20m	270°	0,25°	25ms	12/24V _{DC}	Ethernet
Hokuyo UXM-30LXH-EWA	30m	190°	0,125°	50ms	10 a 30V _{DC}	Ethernet
Sick TIM310-1030000	4m	270°	1°	67ms	9 a 28V _{DC}	USB
Sick TIM351-2134001	10m	270°	1°	67ms	9 a 28V _{DC}	Ethernet, USB
Sick TiM571-2050101	25m	270°	0,33°	67ms	9 a 28V _{DC}	Ethernet, USB
Sick TiM571-2050101	4m	270°	3°	67ms	9 a 28V _{DC}	Ethernet, USB
Sick S10B-9011DA	10m	270°	1°	40ms	16,8 a 30V _{DC}	Ethernet, CAN
Sick LD-OEM1501	250m	360°	0,125°	50ms	24V _{DC}	RS-232, Ethernet

Anexo I

Tipos de rodas

Em [16] são apresentados os tipos de roda da Figura I.1, que podem ser empregados em projetos de robôs móveis.

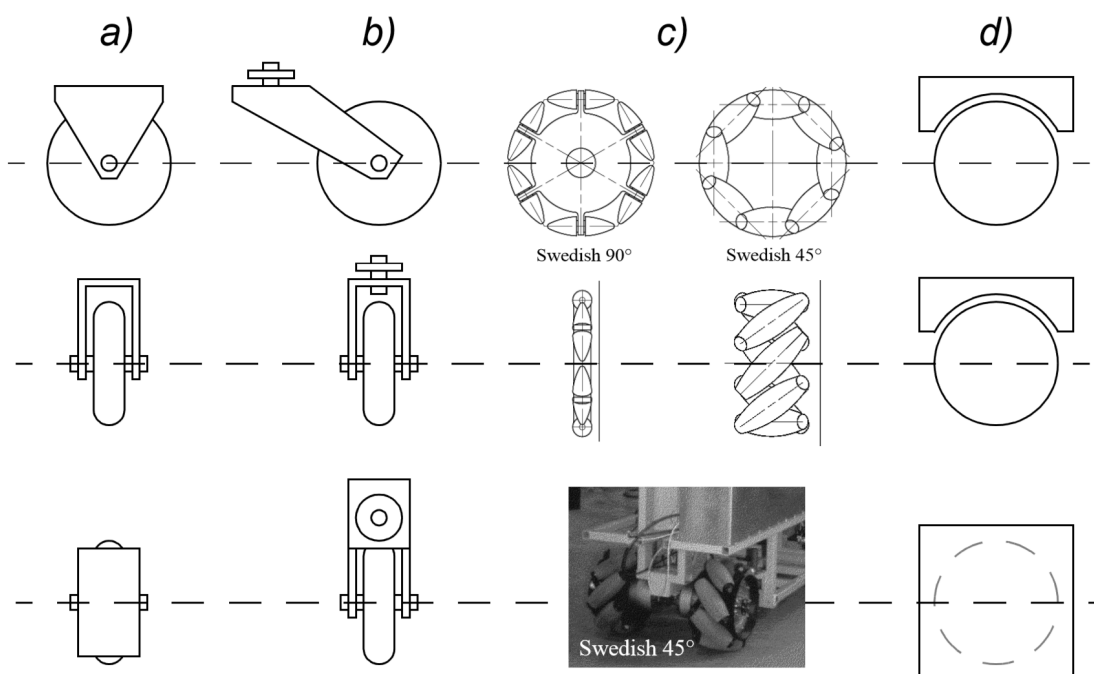


Figura I.1: Tipos de rodas.

- a Roda padrão. Possui dois graus de liberdade: pode girar em torno do eixo da roda e em torno do ponto de contato.
- b Roda *caster*. Possui três graus de liberdade: pode girar em torno do eixo da roda, em torno do ponto de contato e em torno de um eixo perpendicular ao seu eixo de rotação.
- c Roda sueca. É uma roda com rodas ao seu redor. Possui três graus de liberdade: pode girar em torno do eixo da roda, em torno do ponto de contato e em torno dos rolamentos.
- d Roda esférica. Pode se mover de qualquer forma, mas sua fabricação é complicada.