UNIVERSIDADE FEDERAL DE OURO PRETO - UFOP
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO E TÉCNICAS FUNDAMENTAIS

ALEXANDRE SOUZA SANTOS

SISTEMA DE NAVEGAÇÃO PARA UMA CADEIRA DE RODAS ELÉTRICA

Ouro Preto, 2017

ALEXANDRE SOUZA SANTOS

SISTEMA DE NAVEGAÇÃO PARA UMA CADEIRA DE RODAS ELÉTRICA

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para obtenção de Grau em Engenheiro de Controle e Automação.

Orientador: Professor Dr. Gustavo Medeiros Freitas

Co-orientador: Professor Dr. Thiago Antonio Melo Euzébio

Ouro Preto, 2017

S237s Santos, Alexandre Souza .
Sistema de navegação para uma cadeira de rodas elétrica [manuscrito] / Alexandre Souza Santos. - 2017.

58f.: il.: color; grafs.

Orientador: Prof. Dr. Gustavo Medeiro Freitas.
Coorientador: Prof. Dr. Thiago Antonio Melo Euzébio.

Monografia (Graduação). Universidade Federal de Ouro Preto. Escola de Minas. Departamento de Engenharia de Controle e Automação e Técnicas Fundamentais.

1. Sistemas de controle - ROS (sistema operacional de robôs). 2. Cadeiras de rodas - Controle automático. 3. Cadeiras de roda Elétrica. I. Freitas, Gustavo Medeiro. II. Euzébio, Thiago Antonio Melo. III. Universidade Federal de Ouro Preto. IV. Titulo.

Catalogação: ficha@sisbin.ufop.br CDU: 681.5

Monografia defendida e aprovada, em 23 de fevereiro de 2017, pela comissão avaliadora constituída pelos professores:

Prof. Dr. Gustavo Medeiros Freitas - Orientador

Prof. M. Sc. João Carlos Vilela de Castro – Professor Convidado

Prof. Dr. Agnaldo José da Rocha Reis - Professor Convidado

AGRADECIMENTOS

Agradeço primeiramente aos meus pais, Luiz e Magda, pelo amor, carinho e apoio incondicional durante essa minha jornada que não foi fácil. Sem o suporte de vocês nada disso seria possível.

Agradeço aos meus irmãos, Luiz e Marina, tios, tias, avós e todos os familiares que sempre me apoiaram desde o início com palavras amigas, conselhos e também críticas construtivas.

Agradeço a todos os meus amigos, em especial, Jéssica, Kelvin, Eduardo, Arthur, Wallace, Rafael, Halef, Ramon e Carlos, por todos esses anos de amizade e parceria.

Agradeço à UFOP e aos mestres pelo comprometimento com o ensino de qualidade durante toda minha formação acadêmica.

Agradeço aos meus orientadores Gustavo, Thiago e Cocota por todo conhecimento passado e pela paciência.

Agradeço aos meus amigos Rodrigo, Eduardo, Seguchi, Sandro e Allan, grupo que iniciou os trabalhos com a cadeira de rodas na disciplina de Robótica do curso de Engenharia de Controle e Automação.

Agradeço aos amigos do ITV, em especial, Marcos, Jhonny, Aline, Diego, Felipe, Cristiano, Eduardo e Ênio, que tiveram participação fundamental no desenvolvimento deste trabalho.

Por fim, agradeço a todos os amigos que fiz durante minha graduação, em especial os da turma de automação 10.2.

RESUMO

A utilização de equipamentos para facilitar a acessibilidade, como cadeiras de rodas motorizadas, proporciona maior independência a muitas pessoas com limitações de locomoção. No entanto, conforme o grau de limitação, algumas pessoas não estão aptas a controlar esses equipamentos, por exemplo cadeirantes com baixa capacidade de manipulação ou deficiência visual. Com o intuito de amenizar o problema supracitado, promovendo a inclusão de pessoas a esses equipamentos, neste trabalho é apresentado um sistema de controle de navegação para uma cadeira de rodas elétrica. O sistema utiliza a plataforma *Robot Operating System* (ROS), executada no sistema operacional Linux, junto com os componentes de *hardware* Arduino Mega 2560, Kinect do Xbox 360, *laser* Hokuyo UTM-30LX e IMU MTi-G710. O sistema proposto tem como objetivo seguir uma pessoa e evitar colisões. Nesse estudo é feita a identificação dos modelos cinemáticos da cadeira e sintonia de controladores proporcional (P) e proporcional-integral (PI). As estratégias de controle foram validadas por meio de testes em ambientes com e sem obstáculos, e o sistema de navegação apresentou bom desempenho.

Palavras chave: *Sistema de Navegação, Cadeira de Rodas Elétrica, Estratégias de Controle, ROS.*

ABSTRACT

Nowadays, equipment like motorized wheelchair are very useful for people with motor disability, providing greater autonomy to them. However, there are individuals not capable of controlling this kind of equipment, e.g. people with low manipulating skills or visual disability. In order to minimize this problem, this work presents a navigation control system for an electric wheelchair. This system uses the Robot Operating System (ROS) running on Linux and the hardware Kinect Xbox 360, Inertial Measurement Unit (IMU) MTi-G710, laser rangefinder Hokuyo UTM-30LX and Arduino Mega 2560. The goal of the system is to follow a person and avoid collisions. This paper describes the approach to the kinematics modeling of the wheelchair and the tuning of proportional (P) and proportional-integral (PI) controllers. The control techniques were validated by means of tests in surroundings with and without obstacles, and the navigation system showed good performance.

Keywords: *Navigation System, Electric Wheelchair, Control Techniques, ROS.*

LISTA DE SIGLAS E ABREVIATURAS

ROS – *Robot Operating System*

P – Proporcional

PI – Proporcional-Integral

GNSS – *Global Navigation Satellite System*

TA – Tecnologia Assistiva

IBGE – Instituto Brasileiro de Geografia e Estatística

CRIs – Cadeira de Rodas Inteligentes

ICC – Interface Cérebro-Computador

GPS – *Global Positioning System*

MMQ – Método dos Mínimos Quadrados

IMU – *Inertial Measurement Unit*

IHM – Interface Homem Máquina

LISTA DE FIGURAS

Figura 1: Percentual da população do Brasil segundo o tipo de deficiência investigada. 11

Figura 2: Cadeira de rodas modelo Jazzy 600 do fabricante *Pride Mobility Products® Corporation*. 12

Figura 3: Sequência de uma exploração real do robô SmartROB-2. 18

Figura 4: Cadeira de rodas comercial equipada com computador embarcado, câmera, GPS, sensores ópticos, *lasers* e sonares. 18

Figura 5: Teste com usuário controlando a cadeira de rodas elétrica. 19

Figura 6: Arquitetura do sistema da cadeira de rodas elétrica controlada pelo Kinect. 20

Figura 7: Cadeira de rodas *JiaoLong* equipada com as câmeras Kinect e 360º, e seus posicionamentos em relação ao solo. 21

Figura 8: Comandos através dos dedos para controlar a cadeira de rodas: (a) mover para frente, (b) virar à direita, (c) virar à esquerda, (d) mover para trás, (e) mover para a esquerda e (f) mover para a direita. 21

Figura 9: Gestos para controle de navegação do robô (1 – parar; 2 – mover para trás; 3 – mover para frente; 4 – virar à direita; 5 – virar à esquerda). 22

Figura 10: Estimação da posição da cadeira utilizando geometria epipolar e triangulação. 23

Figura 11: Sensor Kinect do Xbox 360. 24

Figura 12: IMU MTi-G710 do fabricante XSENS. 25

Figura 13: Sensor *laser* Hokuyo UTM-30LX. 25

Figura 14: Arduino Mega 2560. 26

Figura 15: Arduino IDE. 27

Figura 16: MATLAB: ambiente de computação numérica utilizado em diversas aplicações. 28

Figura 17: *Joystick* original da cadeira de rodas Jazzy 600. 29

Figura 18: Detalhes da conexão do *joystick* da cadeira de rodas. 29

Figura 19: Circuito elétrico projetado para condicionamentos de sinais e controle da cadeira de rodas. 30

Figura 20: Placa de circuito impresso desenvolvida para condicionamentos de sinais e controle da cadeira de rodas. 31

Figura 21: Cadeira de rodas elétrica com sistema de instrumentação e controle embarcados para realizar o rastreamento de uma pessoa e evitar obstáculos. 31

Figura 22: Diagrama do sistema controle da cadeira de rodas. 32

Figura 23: Funcionamento do sensor Kinect: sistema de coordenadas e identificação da pessoa por juntas. 32

Figura 24: Representação das coordenadas do indivíduo em relação ao Kinect. 33

Figura 25: Grafo de comunicação entre os nós do ROS implementados para o controle da cadeira de rodas. 33

Figura 26: Malha de controle da distância da pessoa em relação ao Kinect. 34
Figura 27: Malha de controle da orientação da pessoa em relação ao Kinect. 34
Figura 28: Curva do comportamento da distância percorrida da cadeira em função do tempo utilizando o sensor Kinect, aplicando uma tensão de 3.5V nos terminais que movimentam a cadeira para a direita. 35
Figura 29: Comparação entre os comportamentos da distância adquiridos pelos sensores Kinect e *laser*. 36
Figura 30: Curva do comportamento angular da cadeira de rodas utilizando o Kinect, aplicando uma tensão de 3.5V nos terminais que movimentam a cadeira para a direita. 36
Figura 31: Comparação entre os comportamentos angulares para a direita e para a esquerda da cadeira de rodas utilizando o Kinect. 37
Figura 32: Comparação entre os comportamentos angulares da cadeira de rodas adquiridos pelos sensores Kinect e IMU. 37
Figura 33: Resposta de um sistema integrador em malha aberta para obtenção dos valores k' e θ. 38
Figura 34: Análise gráfica dos modelos real e estimado de distância. 39
Figura 35: Análise gráfica dos modelos real e estimado de orientação. 39
Figura 36: Integral do erro absoluto (IAE). 41
Figura 37: Curva de Nyquist L (sólido) quando as restrições MS(círculo tracejado) e MT(círculo traço-ponto) de robustez são cumpridas. 42
Figura 38: Posicionamento do sensor *laser* em relação à cadeira de rodas. 42
Figura 39: Pista construída como referência para a pessoa a ser seguida pela cadeira de rodas. 44
Figura 40: Representação 2D da pista onde foram realizados os testes da cadeira de todas. 44
Figura 41: Ambiente com obstáculos para validação do algoritmo de anti-colisão. 45
Figura 42: Comportamento das variáveis de processo (a, c) e das ações de controle (b, d) em relação aos valores do parâmetro KP dos controladores referentes aos modelos de distância e orientação da cadeira de rodas. 46
Figura 43: Diagrama de Nyquist da solução do problema de otimização utilizando o método de sintonia ótima. 46
Figura 44: Ação de controle da cadeira de rodas (b, d) em relação às variáveis de processo (a, c) utilizando os parâmetros calculados pelo método de sintonia ótima. 47

SUMÁRIO

1 INTRODUÇÃO 10
1.1 Considerações iniciais 10
1.2 Motivação 11
1.3 Objetivos 12
1.4 Plataforma disponível 12
1.5 Estrutura do trabalho 13

2 REVISÃO BIBLIOGRAFICA 14
2.1 Fundamentos da navegação 14
2.2 Instrumentação 15
2.2.1 Odômetro 15
2.2.2 Magnetômetro 15
2.2.3 Acelerômetro 15
2.2.4 Giroscópio 16
2.2.5 Dispositivo GPS 16
2.2.6 Sensor de contato 16
2.2.7 Sensor ultrassônico 16
2.2.8 Sensor infravermelho 17
2.2.9 Sensor a *laser* 17
2.3 Sistemas de navegação autônomos e semiautônomos aplicados à robótica 17

3 METODOLOGIA 24
3.1 *Hardware* 24
3.1.1 Sensor Kinect 24
3.1.2 *Inertial Measurement Unit* (IMU) MTi-G710 24
3.1.3 Sensor *laser rangefinder* Hokuyo UTM-30LX 25
3.1.4 Arduino Mega 2560 25
3.2 *Software* 26
3.2.1 ROS 26
3.2.2 Arduino IDE 27
3.2.3 MATLAB 27

4 CONTROLE DA CADEIRA 29
4.1 Funcionamento da cadeira de rodas Jazzy 600 29

4.2 Circuito de condicionamento de sinais 30
4.3 Sistema de controle da cadeira de rodas 31
4.4 Modelagem da cadeira de rodas 35
4.4.1 Modelagem da cadeira em relação à sua velocidade linear 35
4.4.2 Modelagem da cadeira em relação à sua velocidade angular 36
4.4.3 Funções de transferências dos modelos de distância e orientação da cadeira de rodas 38
4.5 Sintonia do controlador 40
4.6 Detecção de obstáculos 42

5 RESULTADOS EXPERIMENTAIS E AJUSTE DE SINTONIA 44
5.1 Descrição dos testes 44
5.2 Resultados 45
5.3 Análise dos resultados 47

6 CONCLUSÃO 49

7 REFERÊNCIA BIBLIOGRÁFICA 50

8 ANEXOS 55

1 INTRODUÇÃO

Neste capítulo é apresentada uma breve introdução sobre sistemas de navegação, assim como a motivação, plataforma disponível e os objetivos.

1.1 Considerações iniciais

Desde a antiguidade, sistemas de navegação vêm sendo utilizados para a locomoção do homem no espaço. Com o enorme avanço tecnólogo, instrumentos como mapas e bússolas deram espaço ao *Global Navigation Satellite System* (GNSS), um sistema de navegação por satélite. Nos dias atuais, novos sistemas de navegação têm sido desenvolvidos em diversas áreas como, transporte, aviação, militar, civil e outras. Isso devido às particularidades de cada cenário e aos atributos que esses sistemas proporcionam, que variam desde o conforto à execução de tarefas específicas.

Outra área na qual sistemas de navegação têm se destacado é a de Tecnologia Assistiva (TA). Esse campo é subdividido em diversas categorias que possuem recursos como talheres modificados, roupas de fácil manuseio, barras de apoio, próteses, sistemas de controle de ambiente, muletas, andadores, cadeira de rodas manuais e elétricas, rampas de acesso, elevadores, acessórios para mobilidade em veículos, equipamentos adaptados para prática de esportes, dentre muitos outros. O objetivo da TA é desenvolver recursos e serviços para auxiliar pessoas com deficiência em relação às suas habilidades funcionais, proporcionando autonomia, maior independência, melhor qualidade de vida e inclusão social (BERSCH; TONOLLI, 2006 apud BERSCH, 2008).

De acordo com o censo de 2010 do Instituto Brasileiro de Geografia e Estatística (IBGE), a população no Brasil é de 190.755.799, onde 23,9% (45.590.634) dela são portadores de alguma deficiência, sendo ela visual, auditiva, motora ou mental (Figura 1). Essas estatísticas só reforçam a grande responsabilidade da comunidade científica em desenvolver soluções que busquem melhorar a qualidade de vida desses cidadãos. No entanto, este trabalho abrange de forma mais específica pessoas portadoras de deficiência que possuem mobilidade reduzida e necessitam de equipamentos como cadeira de rodas para se locomoverem.

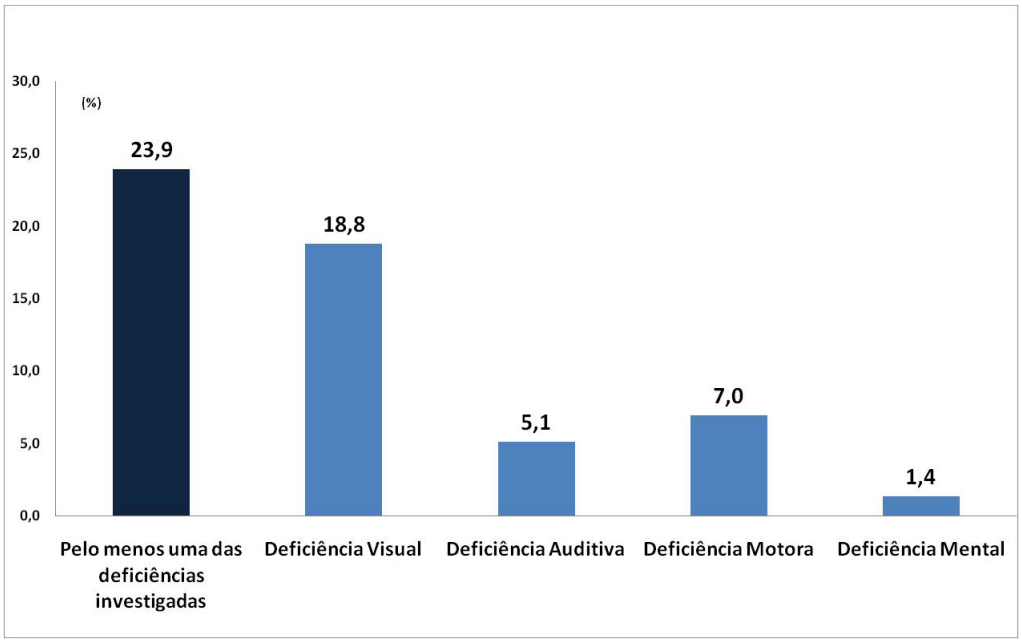

Figura 1: Percentual da população do Brasil segundo o tipo de deficiência investigada.
Fonte: (IBGE, 2010).

1.2 Motivação

Na atualidade, há muito o que se melhorar em relação à acessibilidade em ambientes públicos e domésticos para pessoas com deficiência. A utilização de equipamentos como cadeiras de rodas motorizadas proporciona maior independência a muitos indivíduos com limitações de locomoção. Os que possuem capacidade de manipulação intacta podem utilizar um *joystick* para o controle da cadeira de rodas elétrica. No entanto, conforme o grau de limitação, alguns usuários não estão aptos a controlar esse equipamento. Indivíduos com baixa capacidade de manipulação, deficientes visuais, crianças e idosos, dependem de outras pessoas para conduzir cadeiras convencionais, o que a longo prazo pode ser desgastante (FERREIRA et al., 2007).

Segundo Radabaugh (1993, apud BERSCH, 2008), “para as pessoas sem deficiência, a tecnologia torna as coisas mais fácies. Para as pessoas com deficiência, a tecnologia torna as coisas possíveis”. Seguindo essa linha de pensamento e tendo em vista que vivemos na era da revolução tecnológica, nada mais natural do que usar a tecnologia para resolver os problemas do nosso cotidiano. Com o intuito de amenizar o cenário supracitado, permitindo que pessoas com limitações possam fazer uso de cadeiras de rodas elétricas, este trabalho visa o aumento da acessibilidade, proporcionando inclusão social, autonomia, maior conforto e melhor qualidade de vida paras os usuários.

1.3 Objetivos

Este trabalho tem como objetivo propor um sistema de navegação semiautônomo para uma cadeira de rodas elétrica afim de seguir uma pessoa e evitar colisões.

Neste estudo é feita a identificação do modelo cinemático da cadeira através da análise das variáveis de entrada e saída dos sistemas que representam os movimentos lineares e angulares, e sintonia de controladores P e PI. O trabalho também conta com o desenvolvimento de um algoritmo que realiza a detecção de obstáculos e atua na cadeira de modo a evitar colisões.

1.4 Plataforma disponível

Para o desenvolvimento deste trabalho, a plataforma utilizada foi uma cadeira de rodas elétrica modelo Jazzy 600, ilustrada na Figura 2. Esse equipamento é apropriado para usuários com capacidade limitada de impulsionar uma cadeira de rodas convencional, proporcionando maior conforto e comodidade. A cadeira Jazzy 600 contém 2 motores de 24V, 2 baterias de 12V ligadas em série e um *driver* de acionamento conectado a um *joystick (Hall effect)*. Ela tem capacidade de transportar uma pessoa de até 135kg.

Figura 2: Cadeira de rodas modelo Jazzy 600 do fabricante *Pride Mobility Products® Corporation*.

1.5 Estrutura do trabalho

Este trabalho é composto por 6 capítulos.

No capítulo 1, Introdução, é apresentada uma breve introdução sobre sistemas de navegação, assim como a motivação, os objetivos e plataforma disponível.

No Capítulo 2, Revisão Bibliográfica, são expostos os conceitos teóricos como fundamentação básica para o estudo, além dos componentes necessários para o desenvolvimento do sistema.

No Capítulo 3, Metodologia, são apresentados os instrumentos utilizados para o desenvolvimento do sistema navegação.

No Capítulo 4, Controle da Cadeira, são descritas a abordagem do sistema controle da cadeira de rodas, a modelagem cinemática e a sintonia de controladores.

No Capítulo 5, Resultados Experimentais, são apontados os resultados e as análises do sistema de navegação da cadeira de rodas.

Por fim, no Capítulo 6, Conclusão, são realizadas as considerações finais.

2 REVISÃO BIBLIOGRAFICA

Neste capítulo são expostos os conceitos teóricos como fundamentação básica para o estudo, além dos componentes necessários para o desenvolvimento do sistema. Ainda, é feito um breve resumo sobre os fundamentos da navegação, instrumentação e trabalhos relacionados a sistemas de navegação autônomo e semiautônomo aplicados à robótica.

2.1 Fundamentos da navegação

Desde o início o homem vem buscando meios de otimizar seu trabalho. Isto pode ser visto através da criação de ferramentas na idade da pedra às invenções de máquinas nos tempos modernos (NGELÉTRICA, 2017). A partir daí, processos de automação têm se tornado cada vez mais presentes em aplicações industriais, visando melhorar a qualidade de processos e produtos, aumentando a produção e diminuindo custos (MARCH, 2001). Já em aplicações residenciais, o alvo é o conforto e a segurança.

Em relação à navegação, a automação vendo sendo utilizada como uma poderosa ferramenta no desenvolvimento de sistemas, que podem ser classificados em três tipos:

- Manuais - o operador controla toda a navegação.
- Semiautônomos - o operador indica as referências da navegação.
- Autônomos – a navegação ocorre sem auxílio do operador.

Navegar consiste em planejar e executar uma viagem de um corpo no espaço tendo como referência um ponto destino. Para garantir que o processo de navegação seja executado, é preciso ter conhecimento sobre o corpo que se deseja movimentar e sobre o ambiente no qual o movimento ocorrerá. Quando o ambiente é conhecido, o problema pode ser resumido ao planejamento de trajetória. Caso contrário, a navegação ocorre de acordo com a atualização das informações do ambiente (MARCH, 2001).

Como este trabalho tem como objetivo propor um sistema de navegação semiautônomo para uma cadeira de rodas elétrica afim de seguir uma pessoa e evitar colisões, o problema se resume ao seguimento de trajetória e detecção de obstáculos. Em relação à trajetória, a cadeira deve permanecer sempre alinhada com o indivíduo condutor, mantendo uma distância fixa. Já em relação à detecção de obstáculos, o sistema deve monitorar constantemente a presença de obstáculos no ambiente e, quando necessário, atuar na cadeira de forma a evitar colisões.

2.2 Instrumentação

Em robótica móvel, navegação está relacionada a robôs realizarem tarefas específicas. Além do conhecimento das características do robô, que limitam o espaço físico e o modo que ele movimentará, é necessário também ter conhecimento sobre seu posicionamento e o ambiente ao seu redor. Essas informações são basicamente obtidas através de sensores como odômetros, magnetômetros, acelerômetros, giroscópios, GPSs, sensores de contato, ultrassônicos, infravermelhos, *lasers* e câmeras de vídeo.

2.2.1 Odômetro

Os odômetros são dispositivos que geralmente utilizam *encoders* para medir o giro das rodas de um robô, e assim, determinar sua localização relativa no ambiente através de estimativas. A principal fonte de erros desses sensores está relacionada à sua utilização em ambientes irregulares e/ou escorregadios, onde o escorregamento das rodas do robô não é percebido pelo mesmo (RIBEIRO, 1999).

2.2.2 Magnetômetro

O magnetômetro é um instrumento de navegação e orientação baseado em propriedades magnéticas dos materiais ferromagnéticos e do campo magnético terrestre. É um objeto com uma agulha magnética que indica o sentido que leva ao polo norte magnético terrestre. Com esse dispositivo, é possível identificar a orientação do robô no ambiente sem precisar ter informações do seu estado anterior.

2.2.3 Acelerômetro

Os acelerômetros, como o nome próprio sugere, são dispositivos utilizados para medir as acelerações de um corpo que está ligado a ele. Além disso, esses sensores podem ser empregados para medir velocidades e distâncias percorridas através da integração. Dentre as suas diversas aplicações, os acelerômetros são principalmente usados em sistemas de posicionamento e como sensores de inclinação e vibração.

2.2.4 Giroscópio

Os giroscópios são equipamentos baseados no princípio da inércia, que permite medir a velocidade de rotação do robô em torno do seu eixo. A velocidade angular pode ser integrada em relação ao tempo, determinando assim a posição angular do robô no ambiente.

2.2.5 Dispositivo GPS

O GPS é um dispositivo de navegação por satélite capaz de retornar a localização do robô em termos de latitude, longitude e altitude. Esses dados são calculados a partir das distâncias do robô em relação a satélites de referência. Para que a posição seja determinada corretamente, são necessários no mínimo 4 satélites (MASUMOTO, 1993).

2.2.6 Sensor de contato

Os sensores de contato são dispositivos eletromecânicos que detectam mudança através do contato físico direto com o objeto alvo. Geralmente são utilizados como recurso de emergência quando o sistema não identifica os obstáculos, ou por não serem reflexivos à luz vermelha ou por absorverem os pulsos de alta frequência dos ultrassônicos (MARCH, 2001).

2.2.7 Sensor ultrassônico

Esses dispositivos utilizam a propagação de ondas acústicas de frequências superiores ao espectro audível (acima de 20 kHz) para extrair informações do ambiente. Eles possuem um emissor e um receptor, geralmente encontrados no mesmo encapsulamento, que são capazes de medir a distância de um determinado objeto a partir do eco. Essa informação é obtida através do intervalo de tempo entre a emissão e o retorno da onda sonora. Como a velocidade de som é conhecida (c = 343 m/s), a distância D entre o sensor e objeto pode ser calculada através da seguinte equação:

$$D = \frac{c.To}{2}, \tag{1}$$

onde To é o tempo de viagem da onda. O fator 2 converte a distância total da viagem na distância real entre o objeto e o sensor (SICILIANO; KHATIB, 2008).

Os sensores ultrassônicos apresentam algumas características que os tornam muito populares como, baixo custo, baixo consumo energético, baixo processamento computacional, alta resistência em ambientes com sujeira e umidade, alta durabilidade, entre outras (PEPPERL+FUCHS, 2010).

2.2.8 Sensor infravermelho

Os sensores infravermelhos são um dos dispositivos mais utilizados no mundo da automação devido a sua grande aplicabilidade, baixo custo e funcionamento simples. A maioria utiliza o método da triangulação para calcular a distância de um determinado obstáculo. Um pulso de luz infravermelha é emitido pelo emissor do sensor, que quando atinge um objeto, é refletido formando um ângulo entre o emissor, o ponto de reflexão e o detector do sensor. Sendo assim, é possível determinar a distância do objeto, dado que o ângulo formado varia com a distância (ALCANTRA et al., 2013).

2.2.9 Sensor a *laser*

Os sensores a *laser* são caracterizados pela sua alta precisão, confiabilidade e durabilidade. Os que são utilizados para medir distâncias se baseiam no princípio de tempo de voo. O sensor dispara um feixe de luz unidirecional, que ao atingir um objeto, é refletido e chega até o receptor. Como a velocidade da luz é constante, a distância pode ser calculada com base no tempo de voo (IFM, 2017).

- Câmeras de vídeo

As câmeras de vídeo são ferramentas muito utilizadas em robótica devido ao grande potencial da informação visual. Com esses dispositivos é possível fazer o mapeamento do ambiente, detectando padrões e obstáculos. As principais limitações desses sensores dizem respeito à sensibilidade a luz e ao alto grau de complexidade no processamento de imagens (MARCH, 2001).

2.3 Sistemas de navegação autônomos e semiautônomos aplicados à robótica

Muitos estudos têm sido realizados em relação ao desenvolvimento de aplicações envolvendo cadeira de rodas. Por exemplo, na tese de doutorado desenvolvida por Becker (2000) são apresentados estudos sobre aplicação de tecnologias assistivas e técnicas de controle em Cadeira de Rodas Inteligentes (CRIs). Nesse trabalho é feita uma análise biomecânica do processo de acionamento de cadeiras de rodas manuais em pistas planas e rampas, e um algoritmo em MATLAB foi desenvolvido para simulação. Ainda, foram testadas técnicas de mapeamento e um controlador *fuzzy* foi implementado para navegação e desvio de obstáculos. Os testes foram realizados utilizando o robô SmartROB-2 equipado com dois sensores *laser* do tipo SICK LSM200 e o sistema operacional XOberon. A sequência ilustrada na Figura 3 mostra

a exploração real do robô em um ambiente construído, onde as linhas que saem dos sensores SICK representam as medições que encontraram as paredes do mapa.

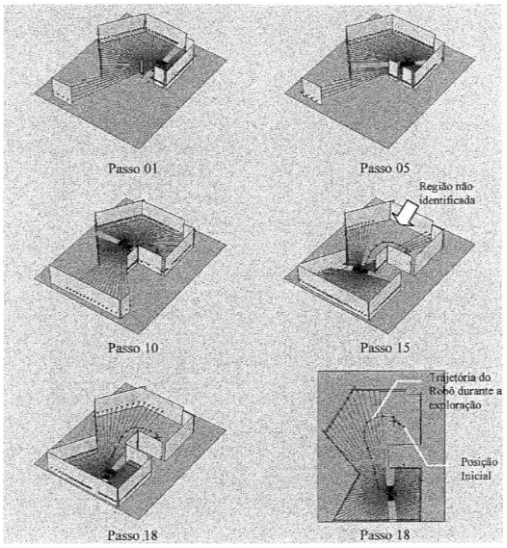

Figura 3: Sequência de uma exploração real do robô SmartROB-2.
Fonte: (BECKER, 2000).

No trabalho desenvolvido por Li et al. (2013) é proposto um sistema de navegação autônomo para uma cadeira de rodas elétrica comercial utilizando a plataforma ROS e múltiplos sensores. A cadeira é equipada com computador embarcado, câmera, GPS, sensores ópticos, *lasers* e sonares (Figura 4). Os sensores e os motores da cadeira são conectados ao computador, onde são feitos o tratamento de dados, o mapeamento do ambiente, a localização da cadeira e o planejamento de trajetórias. Para que a navegação fosse realizada, todos esses elementos tiveram que ser integrados. Ainda, o sistema utiliza um controlador PI para controlar a velocidade das rodas da cadeira, melhorando a precisão dos seus movimentos. O sistema também conta com uma interface-homem-máquina (IHM) para que o usuário possa controlar a navegação conforme sua preferência.

Figura 4: Cadeira de rodas comercial equipada com computador embarcado, câmera, GPS, sensores ópticos, *lasers* e sonares.
Fonte: (LI et al., 2013).

No trabalho descrito em Barbosa et al. (2013) é feita a implementação de um eletroencefalógrafo (EGG), dispositivo capaz de identificar atividades cerebrais, e uma interface cérebro-computador (ICC) para controle de uma cadeira de rodas elétrica. A interface se baseia na análise de sinais eletroencefalográficos, que interpretam atividades mentais e as associam aos comandos de movimento da cadeira. Para tal, são propostos diferentes modelos de classificação baseados em redes neurais. A calibração do sistema é realizada em duas etapas. Na primeira o usuário precisa imaginar estar movendo os pés por 11 segundos. Após esse intervalo o usuário descansa por cinco segundos. O mesmo procedimento ocorre para os movimentos da língua e dos braços. Esse processo é repetido até que o sistema armazene 700 amostras. Na segunda etapa, parte do banco de dados obtido (400 amostras) é utilizada para o treinamento dos modelos. Em seguida, o sistema está pronto para identificar atividades mentais. Como resultado, os modelos apresentaram uma taxa de acerto de 90% e cada atividade mental foi identificada na média de cinco amostras, que pode ser traduzida em cinco segundos ou menos, dependendo do tempo de amostra estabelecido. Um dos testes realizados pode ser pode ser observada pela Figura 5.

Figura 5: Teste com usuário controlando a cadeira de rodas elétrica.
Fonte: (BARBOSA et al., 2013).

Baseado no envelhecimento da sociedade e na importância dos cuidados que devem ser tomados com os anciãos, o projeto descrito em Chang et al. (2013) tem como foco o desenvolvimento de um sistema de controle para cadeira de rodas utilizando o Kinect integrado

com sistemas embarcados, comunicação *Zigbee*, reconhecimento de gestos, posicionamento com infravermelho e processamento através de imagens (Figura 6). No sistema desenvolvido, o ancião usa um gesto com uma das mãos para chamar a cadeira de rodas. Uma vez que o Kinect reconhece o gesto, é requerido que a cadeira se mova até o ancião, um caminho é indicado por infravermelho. Durante o movimento da cadeira de rodas, um sensor CMOS captura as imagens iluminadas pelo infravermelho, calcula o erro para o percurso do movimento e o fornece para o controlador com o intuito de ajustar o percurso. Com a cadeira de rodas já ao alcance do ancião, ele seleciona o local de destino em um painel sensível ao toque. Quando o ancião não precisa mais da cadeira de rodas, ela retorna automaticamente ao seu local padrão de estacionamento. As análises dos resultados mostraram que o sistema apresentou um erro do percurso planejado de valor máximo de 20cm e um erro do ponto final desejado de valor máximo de 10cm.

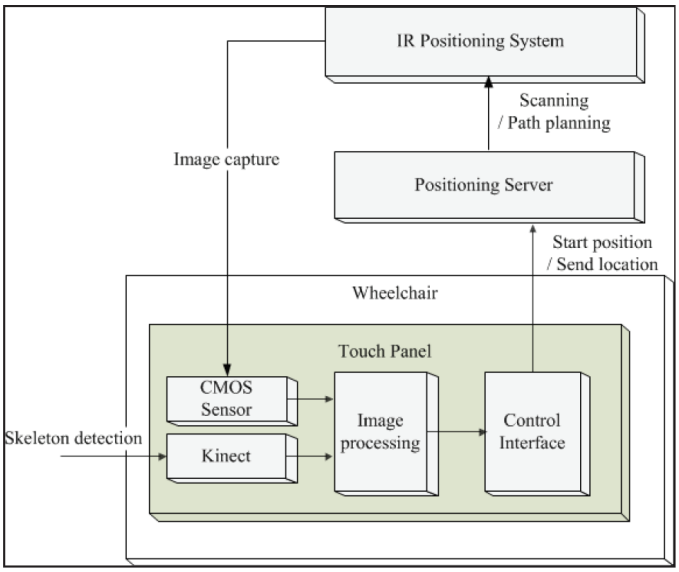

Figura 6: Arquitetura do sistema da cadeira de rodas elétrica controlada pelo Kinect.
Fonte: (CHANG et al., 2013).

Na dissertação de mestrado do Zanetello (2014) é implementado um sistema de controle para a cadeira de rodas inteligente JiaoLong, que permite ao usuário conversar normalmente com uma outra pessoa enquanto ambos se movimentam pelo ambiente. Para o desenvolvimento desse sistema, foram utilizadas a câmera RGB-D do Kinect e uma câmera 360º desenvolvida pela *ARI of Shanghai Jiao Tong University*. A estrutura elaborada está ilustrada na Figura 7. Inicialmente, a câmera RGB-D é utilizada para detectar a pessoa com quem o cadeirante deseja conversar. Após a detecção, a cadeira de rodas inteligente posiciona-se ao lado do indivíduo e acompanha seus movimentos. Para a segurança do usuário, o sistema também faz o reconhecimento e desvio de obstáculos.

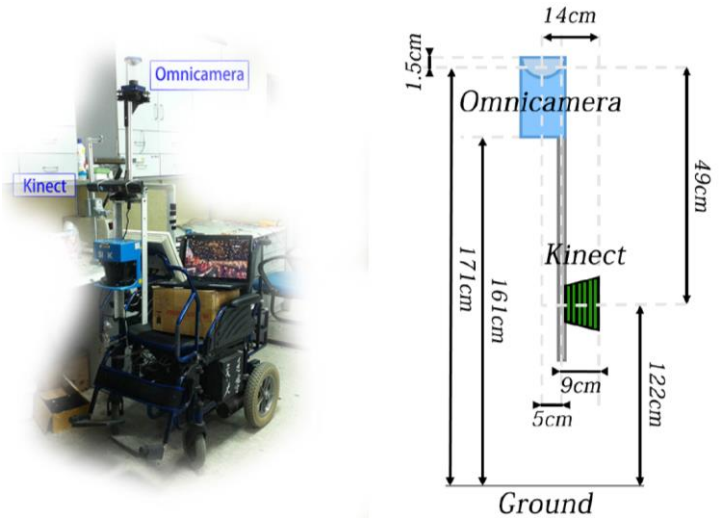

Figura 7: Cadeira de rodas *JiaoLong* equipada com as câmeras Kinect e 360º, e seus posicionamentos em relação ao solo.
Fonte: (ZANETELLO, 2014).

No trabalho descrito em Trigueiros e Ribeiro (2015) é proposto um sistema de visão computacional capaz de guiar uma cadeira de rodas através de gestos manuais. A mão do usuário é detectada pelo Kinect e um algoritmo é utilizado para identificar os movimentos. O dedo indicador é usado para controlar os movimentos para frente, para virar à esquerda e virar à direita (Figura 8 - a, b, c). O movimento para trás é acionado com dois dedos em formato de 'V' (Figura 8 - d). Já os polegares controlam os deslocamentos laterais; para à esquerda e para à direita (Figura 8 – e, f). Caso o usuário deseje parar a cadeira, basta fechar a mão. O sistema apresentou bom comportamento durante os testes efetuados em diferentes cenários, com e sem obstáculos. As informações das mãos demoram certa de 4ms para serem extraídas e 2ms para serem classificadas, possibilitando que a navegação ocorra de uma forma bem natural.

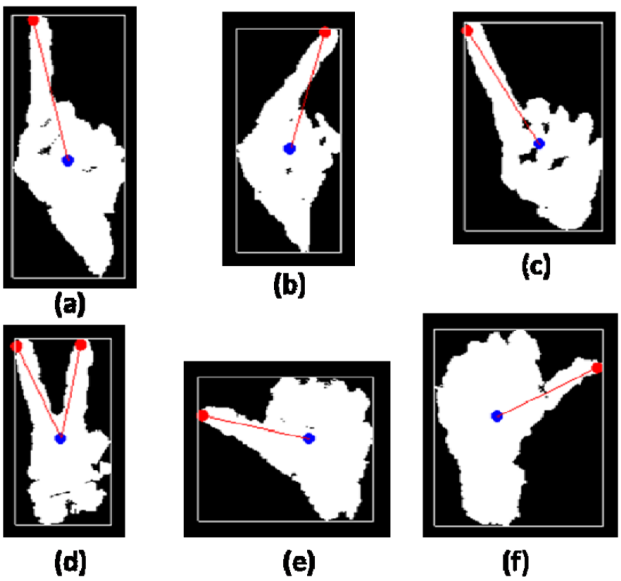

Figura 8: Comandos através dos dedos para controlar a cadeira de rodas: (a) mover para frente, (b) virar à direita, (c) virar à esquerda, (d) mover para trás, (e) mover para a esquerda e (f) mover para a direita.
Fonte: (TRIGUEIROS; RIBEIRO, 2015).

No artigo Schneider et al. (2015) é elaborado um controle de navegação do robô AmigoBot através dos movimentos dos braços do usuário. Para tal, o Kinect é utilizado para a detecção dos movimentos dos braços, que são processados através de uma biblioteca de lógica *fuzzy* para controlar a direção e a aceleração do robô. Os gestos para navegação do robô estão representados pela Figura 9. Como o controle das velocidades angulares e lineares são independentes, os gestos ‘2’ e‘3’ da Figura 9 podem ser combinados com os gestos ‘4’ e‘5’. Após a implementação e análise dos resultados obtidos, os autores concluíram que o controle desenvolvido foi viável para a aplicação desejada. Apesar de satisfeitos com os resultados, foi constatado que para a implementação do protótipo seriam necessários mais testes e uma calibração para todos os usuários que utilizassem o controle.

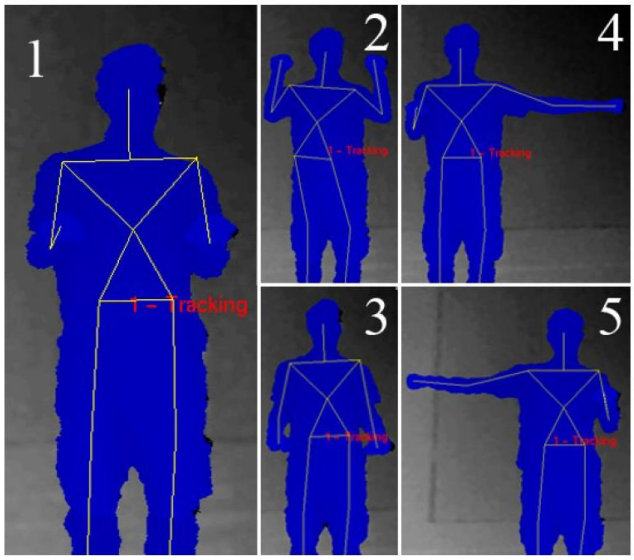

Figura 9: Gestos para controle de navegação do robô (1 – parar; 2 – mover para trás; 3 – mover para frente; 4 – virar à direita; 5 – virar à esquerda).
Fonte: (SCHNEIDER et al., 2015).

Por fim, no estudo realizado por Nasri et al. (2016) é apresentado um algoritmo de navegação autônoma para controlar uma cadeira de rodas elétrica utilizando o ROS. O algoritmo de geolocalização recebe dados de uma plataforma composta por uma câmera 360º, 11 sensores ultrassônicos e 12 sensores infravermelhos que, juntos, conseguem obter informações sobre a posição da cadeira no ambiente e as distâncias dos objetos ao seu redor. A posição da cadeira é estimada utilizando geometria epipolar e triangulação (Figura 10). O algoritmo de prevenção de colisões consiste na fusão dos dados dos sensores ultrassônicos e sensores infravermelhos que, após coletados, são tratados e enviados para ROS. Ainda, uma interface de visualização foi desenvolvida para o monitoramento dos obstáculos ao redor da cadeira. Vários experimentos foram realizados em trajetórias retilíneas e em forma de Z, e o sistema apresentou pequenos erros nos eixos X e Y. No entanto, os erros foram maiores em

trajetórias em forma de Z do que em trajetórias retilíneas, mostrando que a complexidade da trajetória influencia na precisão do algoritmo.

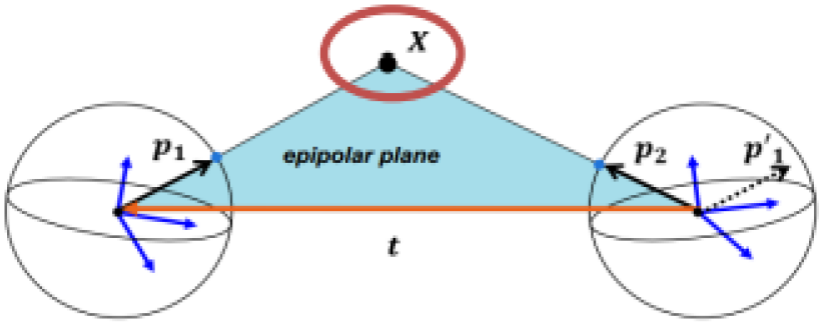

Figura 10: Estimação da posição da cadeira utilizando geometria epipolar e triangulação.
Fonte: (NASRI et al., 2016).

3 METODOLOGIA

Neste capítulo, são apresentados os instrumentos e metodologias adotados para o desenvolvimento do sistema navegação, tais como *hardware* e *software*.

3.1 *Hardware*

No desenvolvimento do sistema de navegação proposto, foram utilizados equipamentos como sensor Kinect do Xbox 360, sensor de movimento MTi-G710, sensor laser Hokuyo UTM-30LX e a plataforma microcontrolada Arduino Mega 2560.

3.1.1 Sensor Kinect

O sensor Kinect (Figura 11) foi elaborado para consoles de jogos eletrônicos a fim de proporcionar uma experiência diferenciada para os jogadores. Ele é capaz de detectar movimentos, possibilitando que os jogadores interajam com os jogos sem a necessidade de um *joystick*. Esse sensor possui uma câmera RGB, 2 sensores de profundidade infravermelhos e 4 microfones, que são controlados por um processador e *software* embutidos. O Kinect possui um campo de visão estreito e foi projetado para funcionar indoor na ausência de luz solar.

Figura 11: Sensor Kinect do Xbox 360.
Fonte: (BOHRER, 2013).

3.1.2 *Inertial Measurement Unit* (IMU) MTi-G710

A IMU MTI-G-710 (Figura 12) é um dispositivo robusto composto por giroscópios, acelerômetros, magnetômetros 3D e um receptor GNSS *(Global Navigation Satellite System)*. Ele possui um poderoso processador capaz de processar com muita exatidão dados de orientação, posição, velocidade e aceleração linear 3D, taxa de giro, campo magnético e pressão estática. Seu driver contém uma interface gráfica de fácil uso e pode comunicar via porta COM através das linguagens C e C++, além de ter suporte para Windows e Linux (XSENS, 2016).

Figura 12: IMU MTi-G710 do fabricante XSENS.

3.1.3 Sensor *laser rangefinder* Hokuyo UTM-30LX

O sensor *laser rangefinger* Hokuyo UTM-30LX (Figura 13) é um equipamento robusto e de alta velocidade muito utilizado em aplicações robóticas. Os pontos de medição são calculados através de incrementos angulares, que podem ser configurados com uma resolução máxima de 0,25º. Esse sensor consegue captar distâncias de até 30m na faixa de 270º com velocidade de leitura de 25ms. Seu filtro interno e circuito de proteção o tornam menos sensível à luminosidade, possibilitando sua utilização em ambientes externos. Outra característica desse dispositivo é seu baixo consumo de energia, que facilita sua integração em plataformas operadas à bateria (HOKUYO, 2012; ROBOSHOP, 2017).

Figura 13: Sensor *laser* Hokuyo UTM-30LX.

3.1.4 Arduino Mega 2560

O Arduino Mega 2560 (Figura 14) é uma plataforma de *hardware* livre de baixo custo e fácil de usar. A escolha desse dispositivo se deu pela facilidade da programação e os recursos

disponíveis. Ele possui 54 pinos de entradas/saídas digitais (15 podem ser usadas como saídas PWM), 16 entradas analógicas, quatro portas seriais, um oscilador de cristal de 16 MHz, uma conexão USB, entre outros recursos. Essa plataforma utiliza uma linguagem de programação de alto nível chama *Wiring* e, devido às suas facilidades, é muito utilizado em projetos amadores.

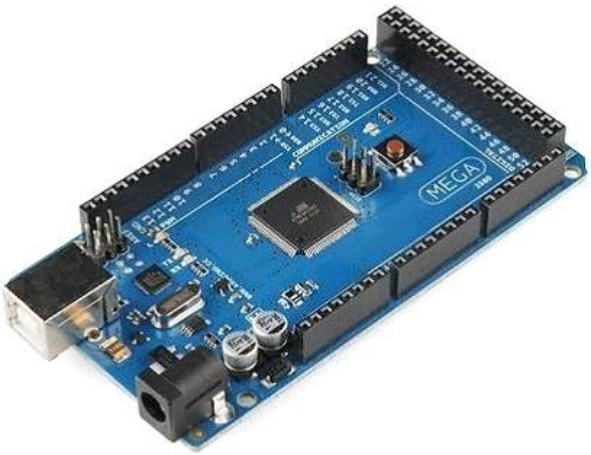

Figura 14: Arduino Mega 2560.

3.2 *Software*

Nesta pesquisa, foram adotadas para programação a plataforma *Robot Operating System* (ROS) rodando no sistema operacional Linux e o ambiente Arduino IDE. Ainda, o MATLAB foi utilizado para tratamento e análise de dados.

3.2.1 ROS

O ROS é uma ferramenta de plataforma aberta muito utilizada para o desenvolvimento de aplicações robóticas. Esse ambiente de programação possui vários recursos inclusos como *drivers*, bibliotecas, visualizadores, gerenciamento de pacotes, entre outros. Ele é capaz de gerenciar dispositivos de forma paralela, sendo que cada dispositivo é denominado por nó, e a comunicação entre nós é feita por mensagens através de tópicos (QUIGLEY et al., 2009).

O ROS utiliza a comunicação *publish-subscribe*, que significa que os nós do tipo *publishers* não direcionam suas mensagens para um nó específico. Ao invés disso, eles publicam suas mensagens dentro de tópicos que ficam acessíveis para qualquer nó do tipo *subscriber*.

3.2.2 Arduino IDE

O Arduino IDE (Figura 15) é um ambiente de programação fácil de usar, projetado para pessoas com pouca ou nenhuma experiência em programação. Ele possui recursos como realce de sintaxe, endentação automática e parênteses correspondentes, que auxiliam bastante na programação. Ainda, com essa aplicação é possível compilar e realizar o *upload* do programa para a placa com apenas um clique.

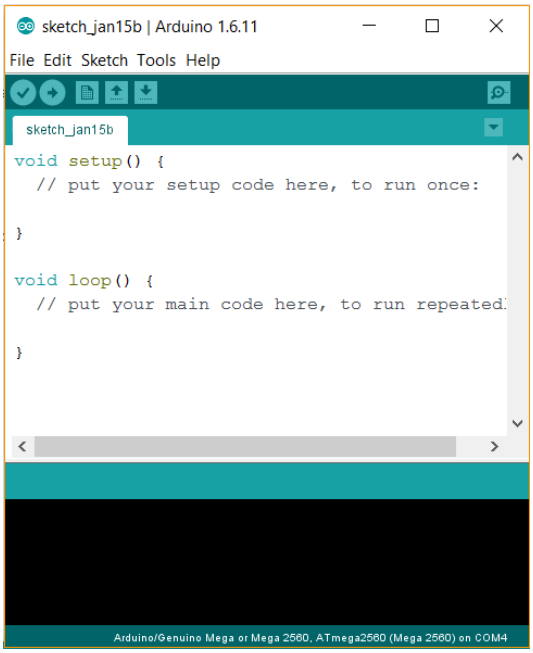

Figura 15: Arduino IDE.

3.2.3 MATLAB

O MATLAB é um ambiente de computação numérica bastante utilizado por engenheiros e cientistas no mundo todo (Figura 16). É uma ferramenta usada para tratamento de dados, processamento de sinais, processamento de imagem, visão computacional, cálculo número, comunicação, computação financeira, controle de sistemas, robótica, entre outras aplicações. Além disso, esse ambiente permite a interação com programas escritos em outras linguagens, incluindo C, C++, C#, Java, Fortran e Python (MATLAB, 2017).

O MATLAB possui um conjunto de aplicações específicas chamadas de *toolboxes*, que são nada mais que uma coleção de funções destinadas a resolverem problemas particulares. Por exemplo, nesse ambiente é possível fazer o uso de recursos como redes neurais, lógica *fuzzy* e muitos outros.

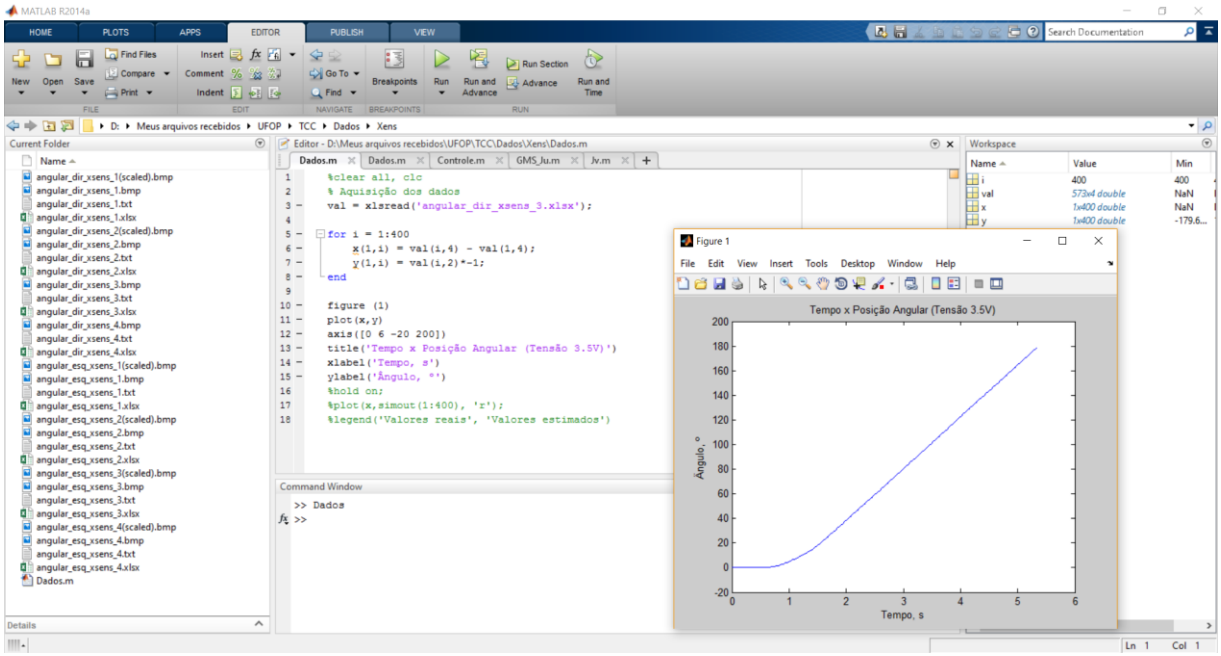

Figura 16: MATLAB: ambiente de computação numérica utilizado em diversas aplicações.

4 CONTROLE DA CADEIRA

Neste capítulo é descrita a abordagem do sistema de controle da cadeira de rodas, assim como a modelagem cinemática e sintonia de controladores.

4.1 Funcionamento da cadeira de rodas Jazzy 600

Originalmente, a cadeira de rodas é comandada em malha aberta por um *joystick* (Figura 17), que transmite sinais analógicos de tensão para os *drivers* de acionamento da cadeira. Esse dispositivo possui 8 conexões e funciona na faixa de tensão [1,1 ~ 3,9]V (Figura 18). As conexões 1 e 3 são utilizadas para alimentação. As conexões 2 e 7 representam as saídas analógicas duplicadas que controlam os movimentos para esquerda e para direita da cadeira. Já as saídas 4 e 5 controlam os movimentos para frente e para trás. A conexão 6, *Centre tap*, é utilizada para verificar a integridade da fonte do *joystick*. Ela deve indicar aproximadamente 50% do valor da alimentação. Caso este valor não esteja entre 49,16% e 50,84% do valor da fonte, um sinal de falha é indicado. Por último, a conexão 8 não é utilizada.

Figura 17: *Joystick* original da cadeira de rodas Jazzy 600.

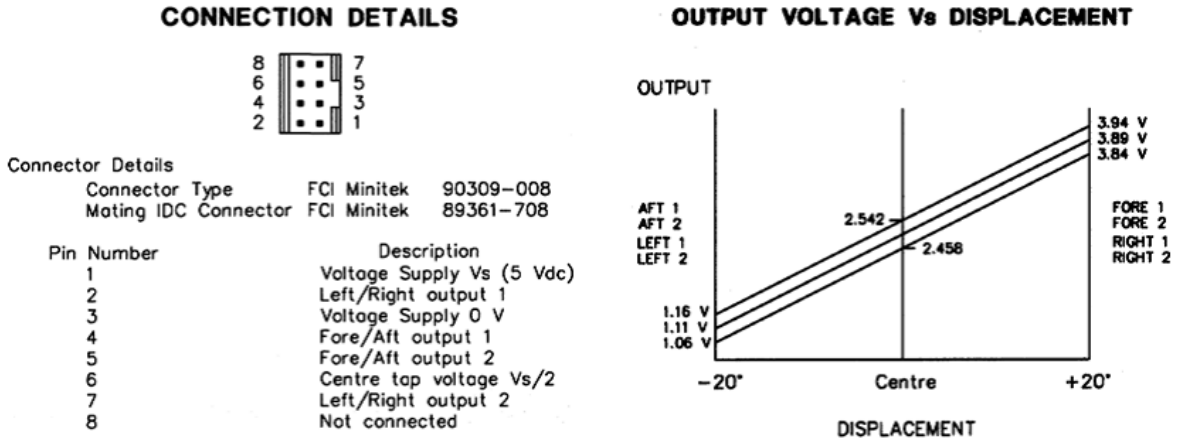

Figura 18: Detalhes da conexão do *joystick* da cadeira de rodas.
Fonte: (CURTISS-WRIGHT, 2015).

As conexões 2, 4, 5, 6 e 7 devem enviar um sinal de 2,5V +/- 1% para que a cadeira seja ligada e permaneça parada. As conexões 2 e 7 controlam a velocidade angular da cadeira. Ao enviarem uma faixa de tensão [2,5 ~ 3,9]V, a cadeira vai para direta e, ao enviarem uma faixa [1,1 ~ 2,5]V, a cadeira move-se para esquerda. Já as conexões 4 e 5 controlam a velocidade linear. Ao transmitirem um sinal de tensão [2,5 ~ 3,9]V, a cadeira move-se para frente. E ao enviarem uma faixa de [1,1 ~ 2,5]V, a cadeira vai para trás.

4.2 Circuito de condicionamento de sinais

Para fazer o controle da cadeira de rodas elétrica, o Arduino é utilizado para emular sinais de tensão analógicos por PWM. No entanto, devido à alta frequência e ruídos, foi desenvolvido um circuito de condicionamento de sinais. Neste circuito, foi implementado um filtro RC para condicionar esses sinais. Ainda, o circuito possui alimentação USB (5V), um divisor de tensão, 2 transistores e 2 relés de 5V. O divisor de tensão é utilizado para gerar uma tensão de 2,5V, necessária para ligar a cadeira. Os transistores chaveiam os relés, responsáveis por selecionar quando a cadeira receberá os sinais PWM condicionados ou 2,5V, significando que cadeira estará em movimento ou estática, respectivamente. É importante ressaltar que as resistências R2 e R3 do divisor de tensão são de precisão (erro máximo de 1%), importante para garantir o funcionamento da cadeira. O circuito elétrico e placa de circuito impresso estão ilustrados nas Figuras 19 e 20.

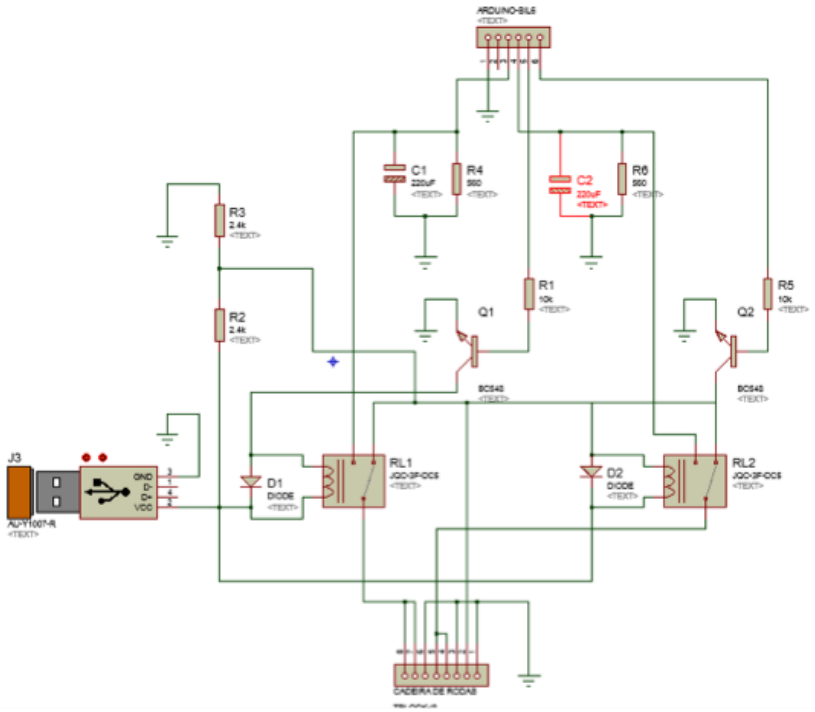

Figura 19: Circuito elétrico projetado para condicionamentos de sinais e controle da cadeira de rodas.

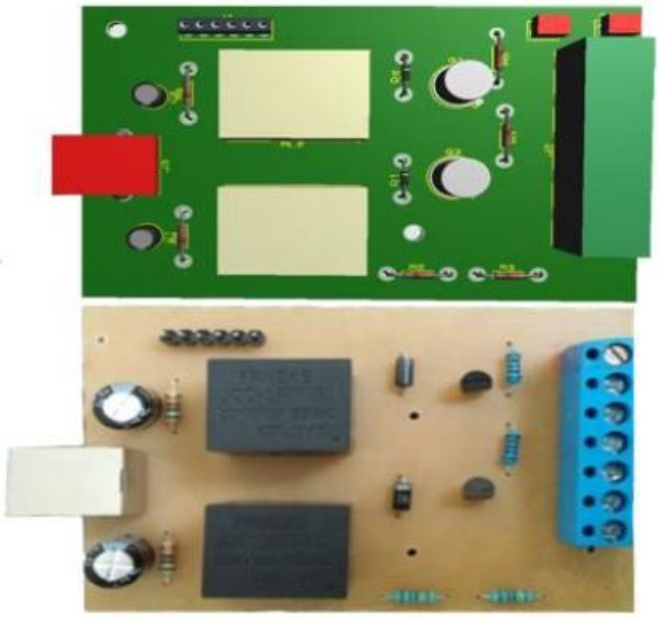

Figura 20: Placa de circuito impresso desenvolvida para condicionamentos de sinais e controle da cadeira de rodas.

4.3 Sistema de controle da cadeira de rodas

Neste trabalho, o objetivo foi desenvolver um sistema de navegação para controlar a cadeira de modo a seguir uma pessoa e evitar colisões. Para tal, foi utilizado o sensor Kinect do Xbox 360 para o rastreamento do indivíduo e o sensor *laser* Hokuyo UTM-30LX para detecção de obstáculos.

Para garantir o conforto do usuário e o bom funcionamento do sistema, foram feitas adaptações na cadeira de rodas para facilitar o acesso do usuário e permitir uma leitura sem oclusão dos sensores. As adaptações podem ser observadas na Figura 21.

Figura 21: Cadeira de rodas elétrica com sistema de instrumentação e controle embarcados para realizar o rastreamento de uma pessoa e evitar obstáculos.

Observe que o Kinect foi posicionado a uma altura que garante que uma pessoa possa sentar na cadeira de rodas sem obstruir o sensor. Pelo mesmo motivo, o *laser* foi posicionado na parte inferior frontal da cadeira de rodas. O sistema embarcado encontra-se logo abaixo do braço esquerdo da cadeira. Por fim, para assegurar o livre movimento da cadeira, os sensores Kinect e *laser* foram adaptados para funcionar com uma bateria de 12V.

Devido às limitações das bibliotecas requeridas pelo Kinect, as versões de *software* escolhidas foram o Ubuntu 14.04 LTS (Trusty Tahr) e o ROS Indigo. O diagrama do sistema está ilustrado na Figura 22.

Figura 22: Diagrama do sistema controle da cadeira de rodas.

O primeiro passo foi realizar a identificação do indivíduo a ser seguido. Para tal função, foi utilizado o pacote (ou nó) *skeleton_tracker* do ROS. Esse pacote é capaz de identificar a posição e orientação de 15 juntas do corpo humano, além de conseguir rastrear até 15 usuários simultaneamente. Outro detalhe importante é que o pacote só começa a rastrear o indivíduo após este fazer uma pose inicial, que é a pose de rendição (mãos para o alto). Caso múltiplos indivíduos façam a pose inicial, o sistema terá como referência o último a ser reconhecido.

Apesar da identificação de várias juntas, apenas a junta do torso foi utilizada nessa aplicação (Figura 23).

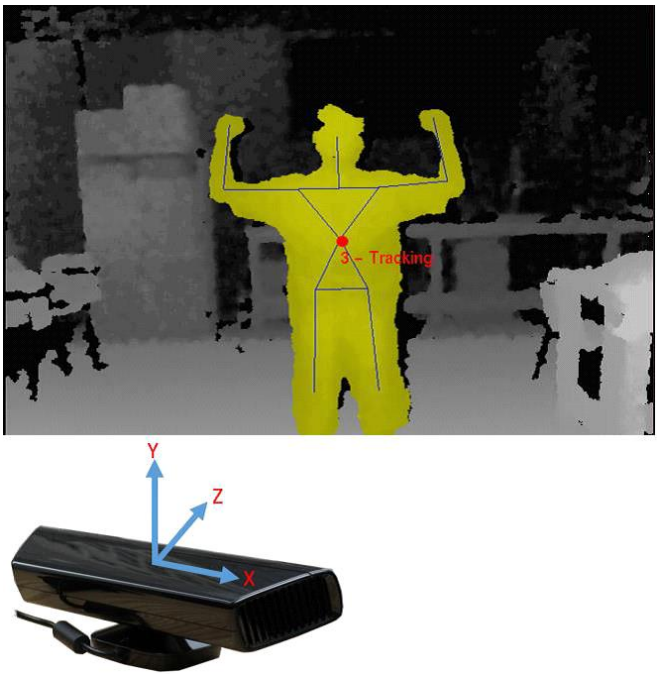

Figura 23: Funcionamento do sensor Kinect: sistema de coordenadas e identificação da pessoa por juntas.

O nó do ROS *skeleton_tracker* foi modificado para publicar (enviar) somente a posição do torso em relação aos eixos X, Y e Z do Kinect através do tópico de nome “posicao”. A mensagem publicada é do tipo *Vector3* da biblioteca *geometry_msgs* do ROS, que contém as coordenadas cartesianas em X, Y, e Z em metros do torso. Essas coordenadas representam os deslocamentos lateral, vertical e longitudinal, conforme ilustrado na Figura 23. No entanto, como o movimento da cadeira ocorre no plano XZ, o controle da cadeira foi feito utilizando coordenadas polares, onde D é a distância e α a orientação entre a pessoa e o Kinect (Figura 24).

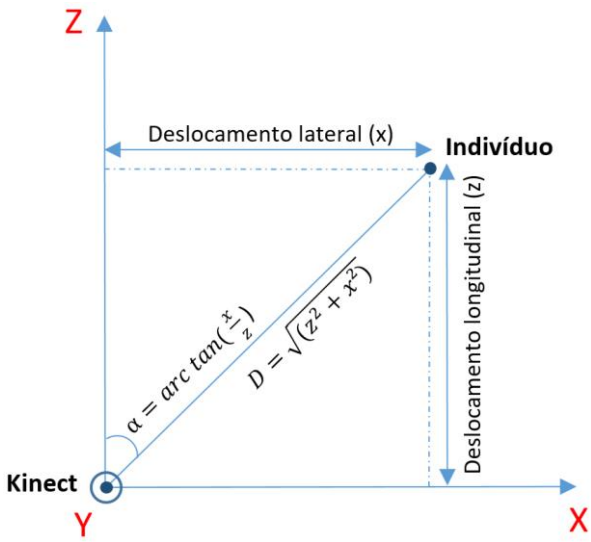

Figura 24: Representação das coordenadas do indivíduo em relação ao Kinect.

Em seguida, um novo nó do ROS chamado “Cadeira” foi criado para controle. Esse nó é responsável por receber as mensagens do tópico “/posicao” e gerar novas mensagens de controle que são publicadas através do tópico “/controle” (ANEXO I). Essas mensagens são recebidas pelo Arduino através da comunicação serial e então, sinais PWM são gerados para controlar as velocidades lineares e angulares da cadeira de rodas (ANEXO II). Ainda, o ROS permite que os usuários possam visualizar a comunicação entre os nós de forma gráfica. O grafo dos nós pode ser observado na Figura 25.

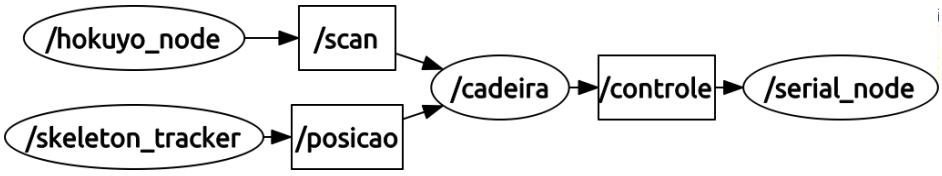

Figura 25: Grafo de comunicação entre os nós do ROS implementados para o controle da cadeira de rodas.

As mensagens de controle contêm uma variável booleana, que sinaliza se a cadeira estará em movimento ou repouso, e os valores de PWM linear e angular. Esses valores foram condicionados de modo a corrigir os erros de distância e orientação da cadeira em relação a

pessoa seguida. Como referência, a distância foi definida com um valor de 1,5m. Para evitar acidentes, foi estabelecido que a cadeira só se move para frente, e que para distâncias inferiores a 1,5m, a cadeira permaneceria em repouso. Devido às limitações do Kinect, também foram definidos uma distância máxima de 4m entre a pessoa o Kinect e um campo de visão de 60º ($|\alpha| \leq 30º$). Acima destes valores, a cadeira também permanece em repouso. Em relação à orientação, o sistema busca manter a cadeira alinhada com o indivíduo. As malhas de controle do sistema estão representadas nas Figuras 26 e 27.

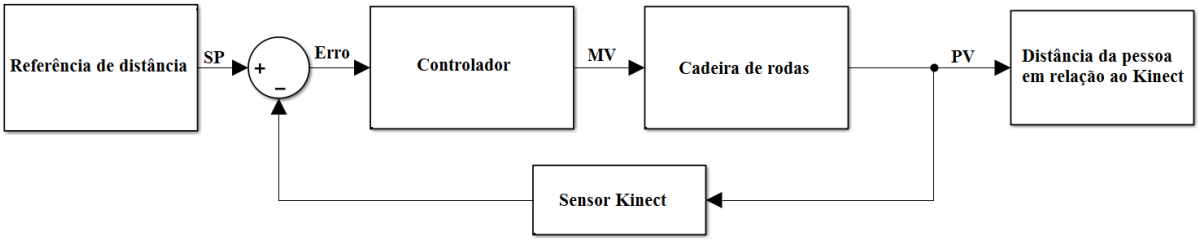

Figura 26: Malha de controle da distância da pessoa em relação ao Kinect.

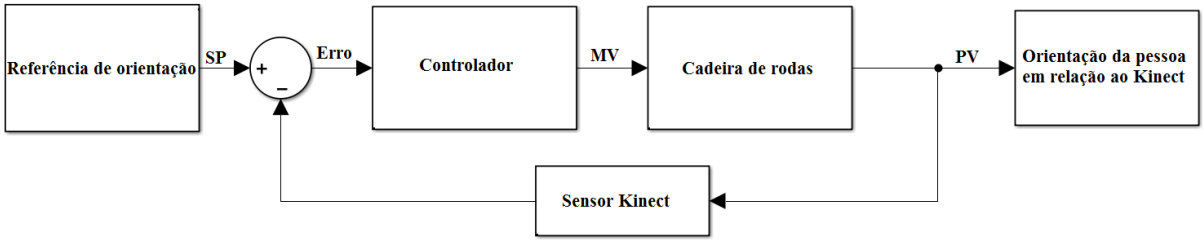

Figura 27: Malha de controle da orientação da pessoa em relação ao Kinect.

Estas malhas possuem alguns termos que devem ser devidamente explicados:

- Variável de Processo (PV): é a variável controlada do sistema. Neste caso, as variáveis são a distância e a orientação da pessoa em relação ao Kinect;
- Set point (SP): é o valor desejado para a variável de processo. Para a distância, o SP foi definido como 1,5m. Para a orientação, o SP é de 0º;
- Variável Manipulada (MV): é a variável sobre a qual o controlador atua para controlar a cadeira de rodas. Neste caso, as variáveis manipuladas são os sinais de tensão simulados por PWM para comandar os movimentos lineares e angulares da cadeira;
- Erro: é a diferença entre SP e PV.

Para condicionar os valores de MV de modo a corrigir os erros de distância e orientação, foi necessário fazer a modelagem cinemática da cadeira e, então, a sintonia de controladores. Esses procedimentos serão descritos nas seções subsequentes.

4.4 Modelagem da cadeira de rodas

Neste trabalho é feita uma modelagem cinemática para a cadeira de rodas elétrica Jazzy 600. Isso porque a cadeira é controlada utilizando comandos de velocidade linear e angular.

Os modelos que descrevem a cinemática da cadeira foram obtidos de forma individual através da análise dos sinais de entrada (PWM linear e angular) e dos sinais de saída (distância e orientação). Sendo assim, o problema de modelagem da cadeira foi divido em dois sistemas, um para velocidade linear e o outro para velocidade angular. Para tal, foram utilizados os sensores Kinect, *laser* e IMU.

4.4.1 Modelagem da cadeira em relação à sua velocidade linear

Para a modelagem da cadeira de rodas em relação à sua velocidade linear, foi feita a análise da variável manipulada e da variável de processo da malha de distância. O seguinte procedimento foi adotado: com o sistema em malha aberta e a cadeira inicialmente em repouso, foi aplicado uma tensão de 3,5V (degrau de 1V) nos terminais que movimentam a cadeira para frente e, através das leituras realizadas pelo Kinect, foi obtido o comportamento da distância percorrida em função do tempo. O comportamento considerado pode ser ilustrado pela Figura 28.

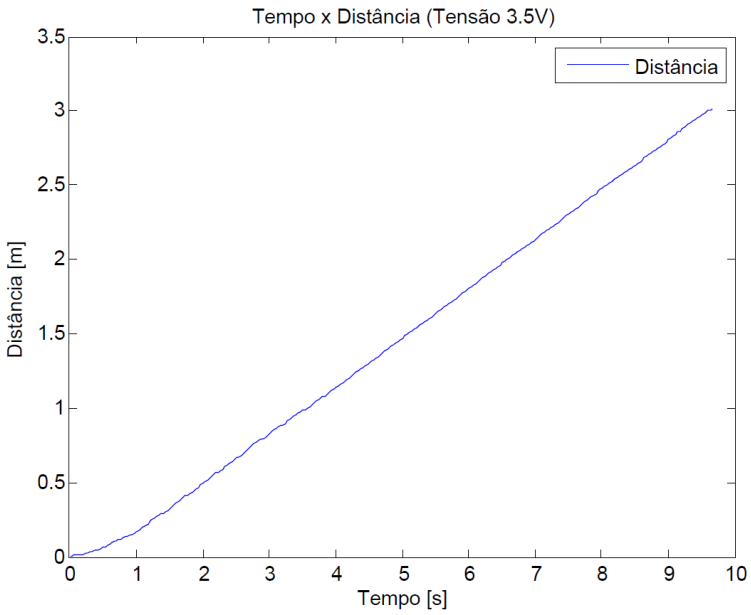

Figura 28: Curva do comportamento da distância percorrida da cadeira em função do tempo utilizando o sensor Kinect, aplicando uma tensão de 3.5V nos terminais que movimentam a cadeira para a direita.

O mesmo procedimento foi realizado utilizando o sensor *laser* para medir as distâncias. Os dados obtidos pelo *laser* são muito semelhantes ao adquirido pelo sensor Kinect. A Figura 29 ilustra tal semelhança.

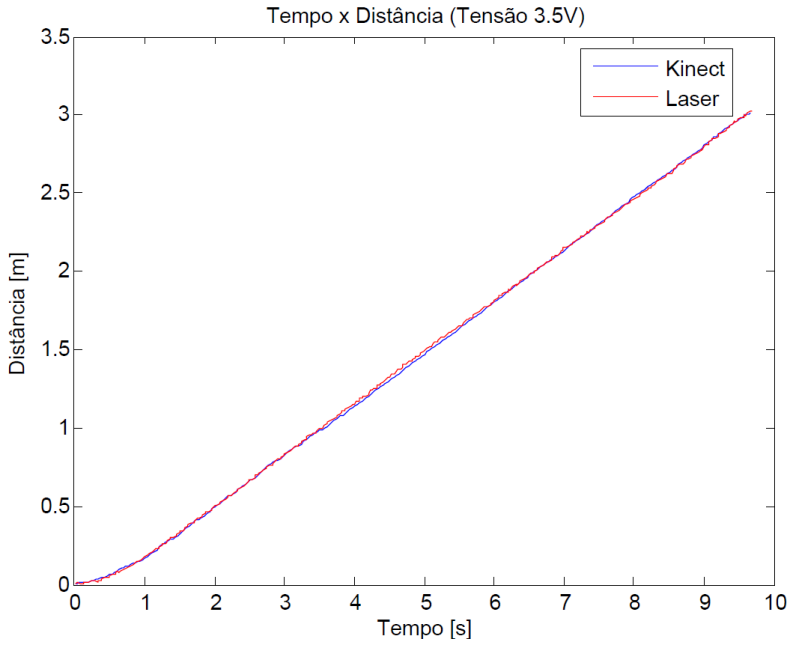

Figura 29: Comparação entre os comportamentos da distância adquiridos pelos sensores Kinect e *laser*.

4.4.2 Modelagem da cadeira em relação à sua velocidade angular

De maneira análoga, a modelagem da cadeira de rodas em relação à sua velocidade angular foi feita através da aplicação de uma tensão de 3,5V (degrau de 1V) nos terminais que movimentam a cadeira pra o lado direito. Sendo assim, através das leituras obtidas pelo Kinect, o comportamento considerado da orientação da cadeira em função do tempo pode ser observado na Figura 30.

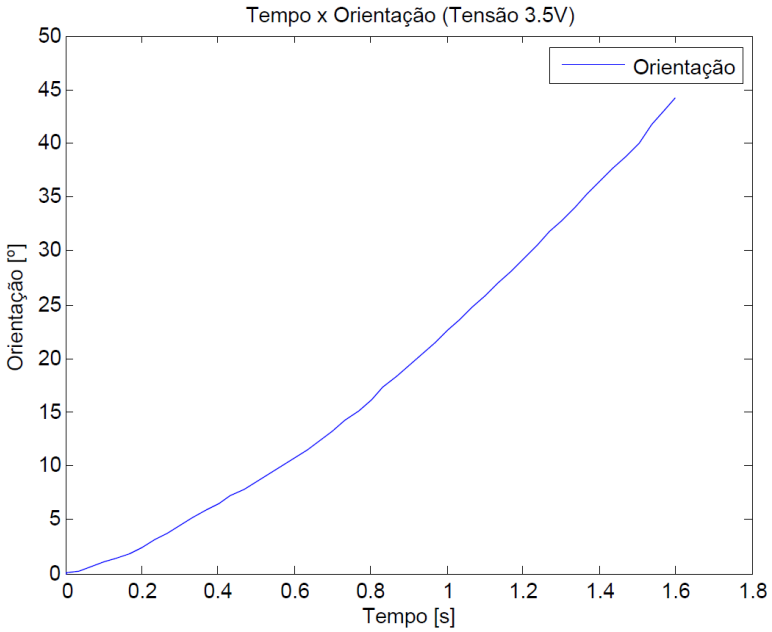

Figura 30: Curva do comportamento angular da cadeira de rodas utilizando o Kinect, aplicando uma tensão de 3.5V nos terminais que movimentam a cadeira para a direita.

O mesmo procedimento foi realizado aplicando uma tensão de 1,5V (degrau de -1V) nos terminais que movimentam a cadeira para a esquerda, e um comportamento semelhante foi obtido (Figura 31). Devido a esta semelhança e por motivos de simplificação, o sistema de navegação proposto neste trabalho considera somente os comportamentos angulares para a direta.

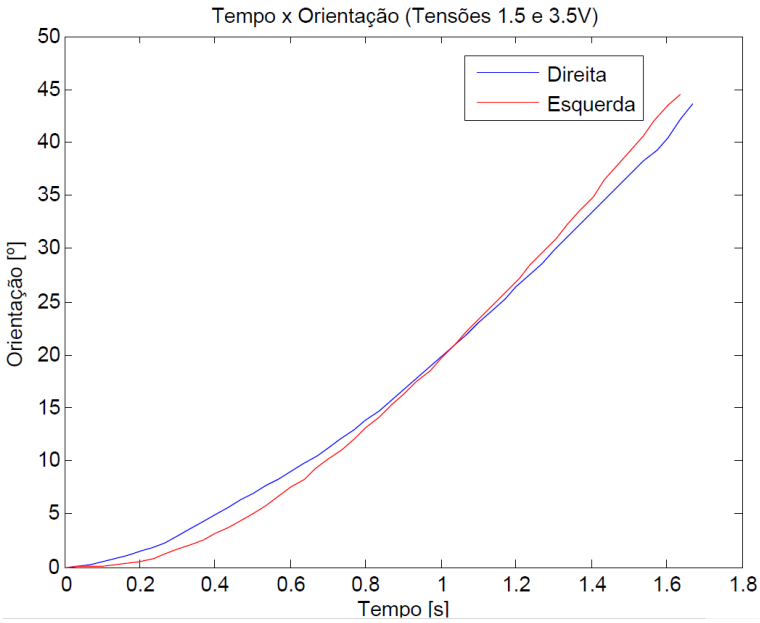

Figura 31: Comparação entre os comportamentos angulares para a direita e para a esquerda da cadeira de rodas utilizando o Kinect.

Ainda, o procedimento foi repetido utilizando a IMU para analisar o comportamento da orientação da cadeira em função do tempo. Os dados obtidos pela IMU resultaram em um comportamento muito semelhante ao adquirido pelo sensor Kinect (Figura 32).

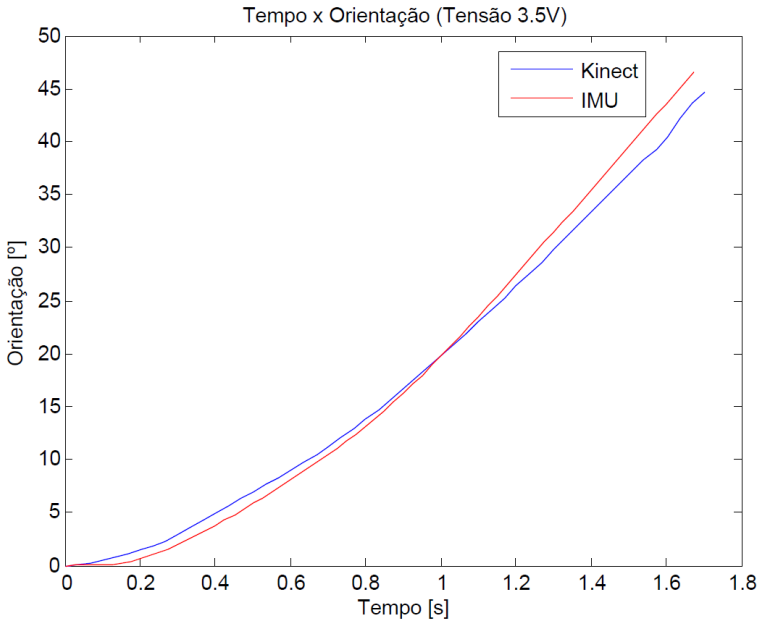

Figura 32: Comparação entre os comportamentos angulares da cadeira de rodas adquiridos pelos sensores Kinect e IMU.

4.4.3 Funções de transferências dos modelos de distância e orientação da cadeira de rodas

Devido à semelhança dos comportamentos obtidos pelos sensores Kinect, *laser* e IMU, os modelos seguintes foram baseados nos dados dos sensores *laser* e IMU devido às suas características de robustez e confiabilidade. Sendo assim, analisando os comportamentos obtidos nas seções anteriores, é possível observar que ambos os modelos apresentaram características de dinâmica integradora com atraso. Consequentemente, a seguinte equação pode ser utilizada para representar os modelos:

$$F(s) = \frac{k'e^{-\theta s}}{s}, \tag{2}$$

onde k'representa o ganho do processo integrador e θ o atraso em segundos (SKOGESTAD; GRIMGOLT, 2012). Esses valores podem ser obtidos a partir da resposta ao degrau do sistema em malha aberta (Figura 33).

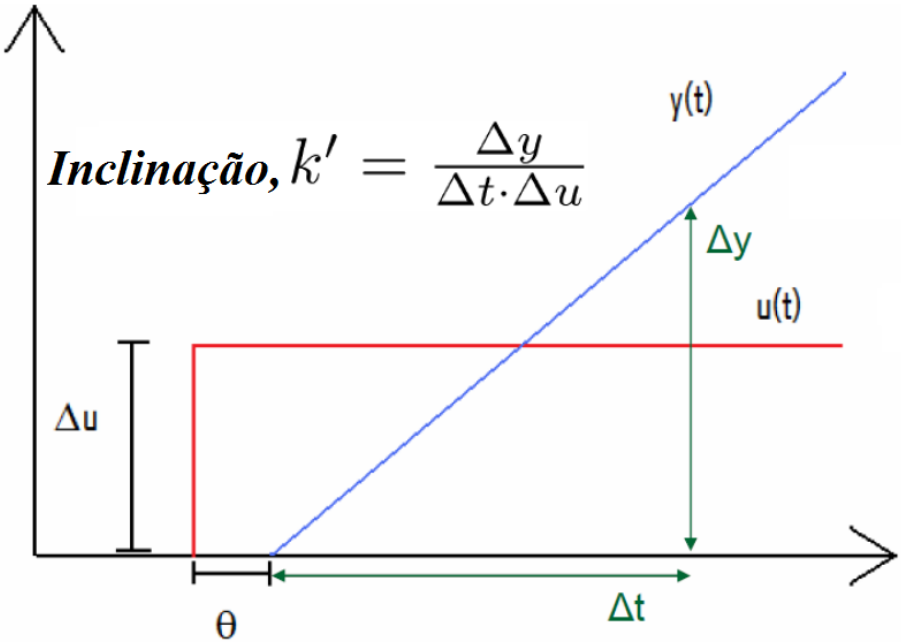

Figura 33: Resposta de um sistema integrador em malha aberta para obtenção dos valores k' e θ.
Fonte: Modificado de Skogestad e Grimholt (2012).

Em relação à distância, o seguinte modelo foi estimado:

$$G(s) = \frac{0.01e^{-0.57s}}{s} \tag{3}$$

A análise gráfica entre os valores reais e estimados pode ser observada na Figura 34.

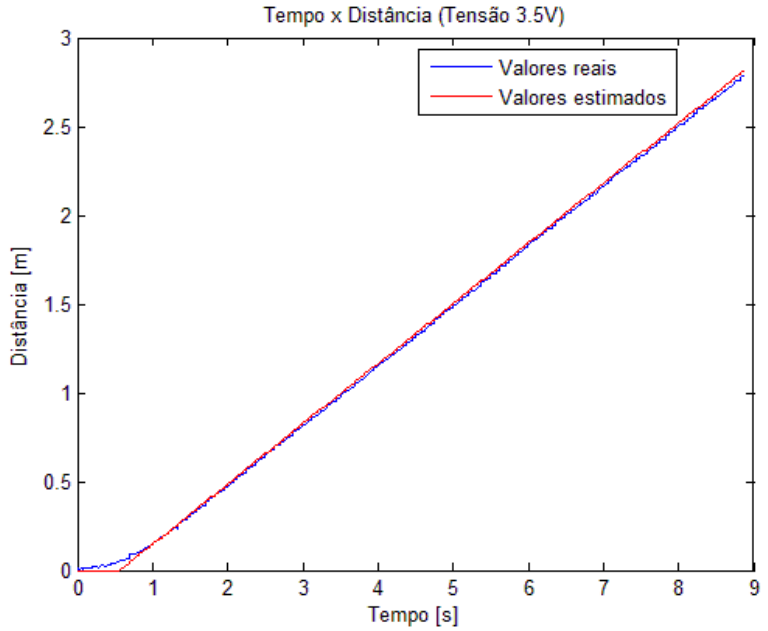

Figura 34: Análise gráfica dos modelos real e estimado de distância.

Em relação à orientação, o seguinte modelo foi estimado:

$$H(s) = \frac{1.55e^{-0.35s}}{s} \tag{4}$$

A análise entre os valores reais e estimados pode ser observada pela Figura 35.

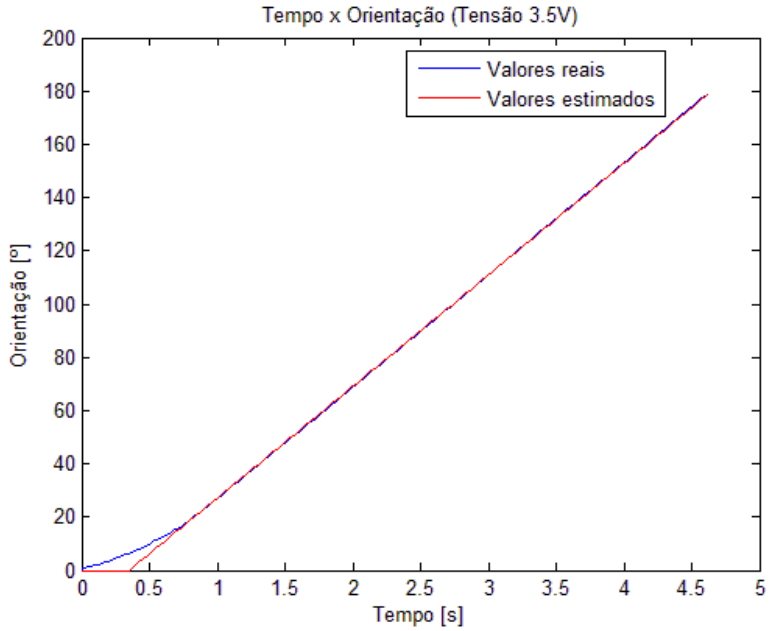

Figura 35: Análise gráfica dos modelos real e estimado de orientação.

Para a avaliar as funções de transferência dos modelos de distância e orientação, o método dos mínimos quadrados (MMQ) ou **R²** foi adotado. Esse método procura encontrar o melhor ajuste para um conjunto de dados minimizando a soma dos quadrados das diferenças entre os valores reais e os valores estimados. Em outras palavras, o **R²** indica de 0 a 1, o quanto

o modelo consegue se aproximar dos valores reais. Quanto maior o valor de **R²**, melhor será a resposta do modelo em relação aos valores estimados.

O modelo descrito pela Equação 3 obteve um coeficiente **R²** de 0,9997 e o modelo da Equação 4 obteve um **R²** de 0,9993.

4.5 Sintonia do controlador

Grande parte das aplicações industriais e em robótica utiliza o controlador do tipo PID. Uma formulação do controlador PID é dada pela seguinte equação:

$$c(s) = K_P + \frac{K_I}{s} + K_D s, \tag{5}$$

onde K_P, K_I e K_D, são as os ganhos proporcional, integral e derivativo, respectivamente.

A ação proporcional do controlador consiste em um amplificador de ganho ajustável que atua de forma proporcional ao erro do sistema. Uma limitação desta ação de controle, quando aplicada isoladamente, é a existência de um erro constante em regimente permanente. Por isso, utiliza-se a componente integral do controlador para eliminar o erro estacionário, pois ela adiciona um polo na origem da função de transferência do controlador. Já a ação derivativa atua na taxa de variação do erro. A adição dessa componente ao controlador não afeta diretamente o erro estacionário e adiciona amortecimento ao sistema.

Apesar do controlador PID possuir apenas três parâmetros, sua sintonia não é trivial. Devido a este fato, existem na literatura diversos métodos de sintonia como Ziegler-Nichols (1942), CHR (CHIEN et al. 1952), Cohen-Coon (1953), IMC (RIVERA et al., 1986), ISE (ZHUANG; ATHERTON, 1993), DS-d (CHEN; SEBORG, 2002), SIMC (SKOGESTAD, 2003), AMIGO (ÅSTRÖM; HÄGGLUND 2004), entre outros.

Neste trabalho são sintonizados controladores P e PI para as malhas de controle de navegação da cadeira de rodas. Uma modificação importante foi implementada no algoritmo do controlador: o acumulador do integrador é zerado assim que o erro se anula, ou seja, ação de controle neste momento também é nula. Isto porque, como mencionando na seção 4.3, foi estabelecido que a cadeira só se move para frente, não permitindo ações reversas de controle. Portanto, o acumulador do controlador nunca seria decrementado, fazendo com que a cadeira ultrapassasse a distância de segurança podendo causar acidentes.

Em relação ao modelo de orientação (Equação 4), um controlador puramente proporcional foi sintonizado a partir da dinâmica da cadeira. No entanto, para o modelo

distância (Equação 3), foi escolhido o método de sintonia ótima para o cálculo dos parâmetros do controlador PI. A vantagem desse método de otimização é que ele pode ser utilizado para qualquer sistema, independentemente da sua ordem.

O método de sintonia ótima é baseado naquele proposto por Kristiansson e Lennartson (2006, apud EUZÉBIO et al., 2012). Neste método, um critério de desempenho e dois de robustez são utilizados no problema de otimização dos parâmetros do controlador PI. Esses critérios são baseados em funções sensibilidade do sistema de controle em malha fechada, onde o objetivo é minimizar a ação de distúrbios (J_v) no processo sob restrições que garantam tanto a estabilidade do sistema quanto limites na atividade de controle.

O critério de desempenho está relacionado com a capacidade do sistema em lidar com distúrbios de carga de baixa frequência, e é uma alternativa no domínio da frequência ao critério baseado em alguma função do erro, por exemplo $IAE = \int_0^\infty |e(t)|dt$ (ÅSTRÖM; HAGGLUND, 1995 apud EUZÉBIO et al., 2012). Em baixas frequências, $J_v \approx \frac{1}{K_I}$. Resultado este que mostra que o critério é praticamente independente da ordem do modelo do processo. Segundo Åström e Hagglund (1995, apud EUZÉBIO et al., 2012), minimizar J_v implica também em uma redução do IAE. A integral do erro absoluto pode ser ilustrada pela Figura 36.

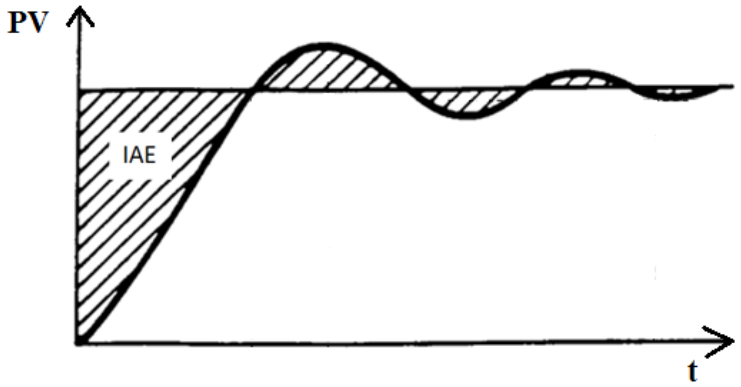

Figura 36: Integral do erro absoluto (IAE).

O critério de robustez tem como objetivo limitar o máximo das funções sensibilidade e sensibilidade complementar. Pata tal, são adotados os parâmetros M_S e M_T, respectivamente. Esses parâmetros correspondem a dois círculos no plano complexo (Figura 37), que impedem que a curva de Nyquist (L) passe por dentro destas regiões, garantindo a estabilidade do sistema quando as restrições de robustez são cumpridas (EUZÉBIO et al., 2012).

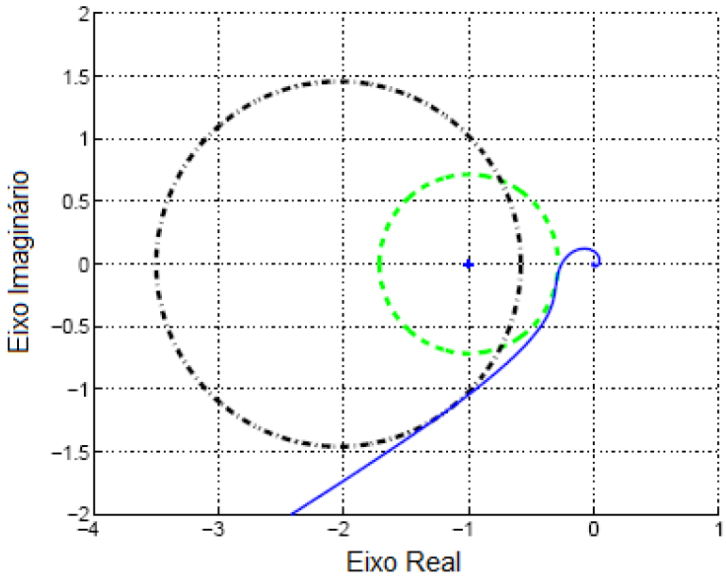

Figura 37: Curva de Nyquist L (sólido) quando as restrições M_S(círculo tracejado) e M_T(círculo traço-ponto) de robustez são cumpridas.

4.6 Detecção de obstáculos

A detecção de obstáculos é feita utilizando as medições do sensor *laser* Hokuyo UTM-30LX, que foi posicionado na parte inferior frontal da cadeira de rodas (Figura 38). Deste modo, o sensor consegue varrer toda área à frente da cadeira.

Figura 38: Posicionamento do sensor *laser* em relação à cadeira de rodas.

O algoritmo de detecção de obstáculos desenvolvido no nó “Cadeira” do ROS, faz uma varredura constante de 120º, verificando se há algum objeto a uma distância de até 70cm do sensor. Caso algum objeto seja detectado, o algoritmo envia um sinal para impedir que a cadeira avance. Ainda, o algoritmo permite que a cadeira gire em torno do seu próprio eixo para

acompanhar a orientação do indivíduo até que a área de varredura esteja completamente livre. Uma vez que nenhum objeto é detectado, o sistema volta para o seu funcionamento original, tentando manter uma distância de 1,5m e sempre alinhada com a pessoa a ser seguida.

5 RESULTADOS EXPERIMENTAIS E AJUSTE DE SINTONIA

Neste capitulo são descritos os testes do sistema de controle de navegação da cadeira de rodas, assim como os resultados e suas respectivas análises.

5.1 Descrição dos testes

Os testes foram realizados em um ambiente de laboratório, onde foi construída uma pista com marcações ao chão (Figura 39). Esta pista serve como referência para a pessoa a ser seguida, e não para a cadeira de rodas. Todos os testes acontecerem da seguinte forma: a cadeira de rodas é sempre posicionada em uma mesma posição inicial, ilustrada na Figura 39, a partir daí, o indivíduo caminha através da trajetória sem obstáculos, tentando manter uma velocidade constante. É importante ressaltar que para evitar a saturação dos controladores, a pessoa seguida manteve uma velocidade inferior à velocidade máxima cadeira de rodas, que é de aproximadamente 6.5 km/h. Os testes duraram cerca de 48 segundos e uma representação 2D da pista com marcações de tempo está ilustrada na Figura 40.

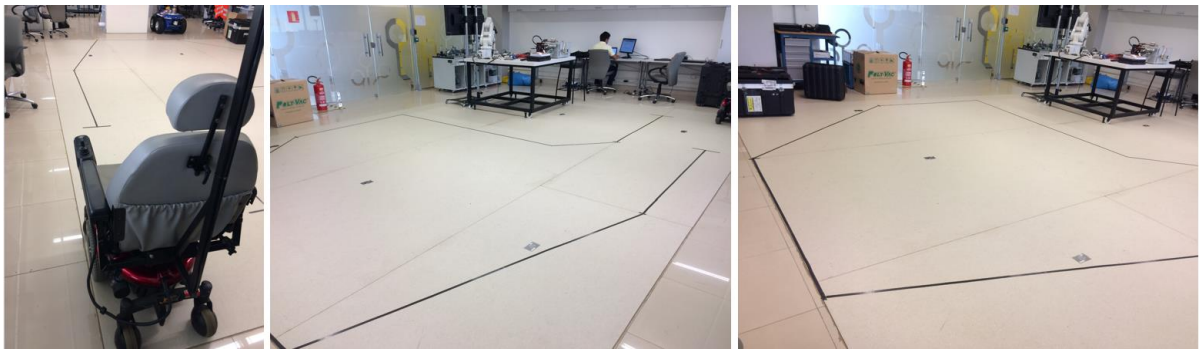

Figura 39: Pista construída como referência para a pessoa a ser seguida pela cadeira de rodas.

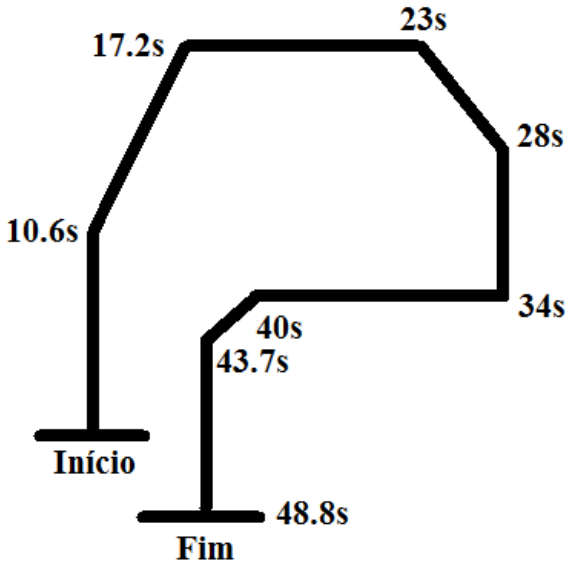

Figura 40: Representação 2D da pista onde foram realizados os testes da cadeira de todas.

Posteriormente, foram adicionados objetos estáticos no ambiente para validação do algoritmo anti-colisão (Figura 41). Um vídeo dos experimentos pode ser encontrado no link: https://www.youtube.com/watch?v=D0DofbOzExk.

Figura 41: Ambiente com obstáculos para validação do algoritmo de anti-colisão.

5.2 Resultados

Inicialmente, foram testados nesta trajetória controladores com apenas ação proporcional para os modelos de distância e orientação da cadeira de rodas. Para tal, foram definidos vários valores para o parâmetro K_P dos controladores. Os comportamentos das variáveis de processo e das ações de controle podem ser ilustrados pela Figura 42.

Posteriormente, o método de sintonia ótima foi utilizado para calcular os parâmetros de um controlador PI para o modelo referente à velocidade linear. As restrições de robustez utilizadas foram MS e MT iguais a 1,15. A curva de Nyquist e as restrições podem ser observadas na Figura 43. Os valores encontrados foram $K_P = 87{,}8$ e $K_I = 16{,}1$. Em relação ao controlador referente à malha de distância, o valor de $K_P = 2$ foi mantido. Os comportamentos das variáveis de processo e das ações de controle podem ser ilustrados pela Figura 44.

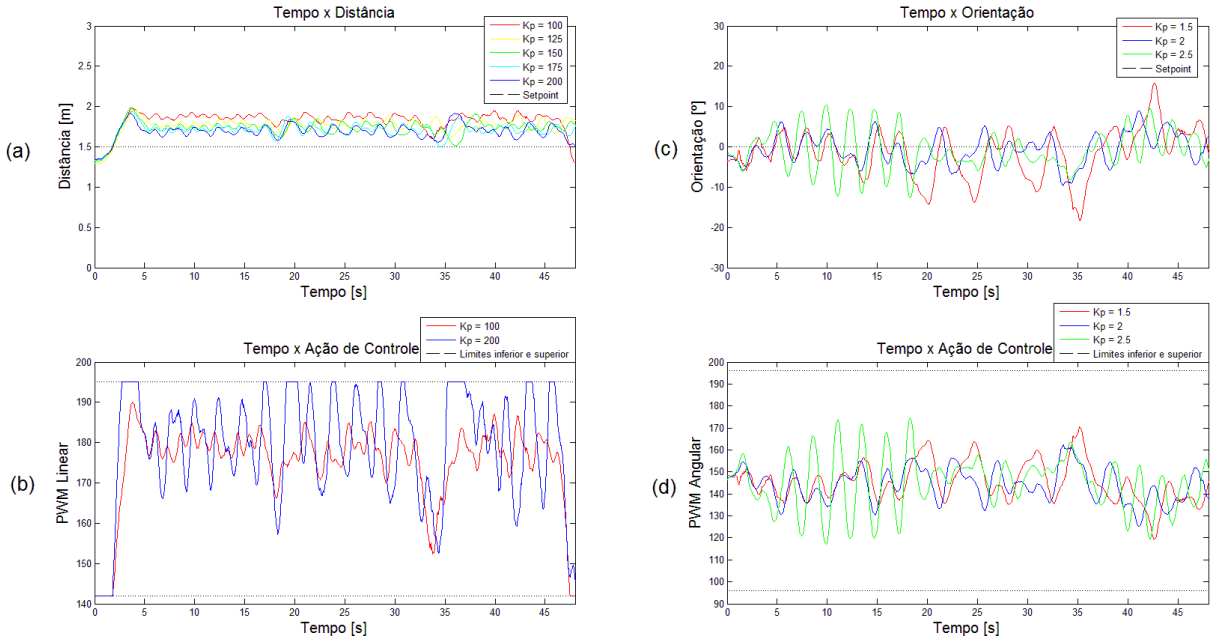

Figura 42: Comportamento das variáveis de processo (a, c) e das ações de controle (b, d) em relação aos valores do parâmetro K_P dos controladores referentes aos modelos de distância e orientação da cadeira de rodas.

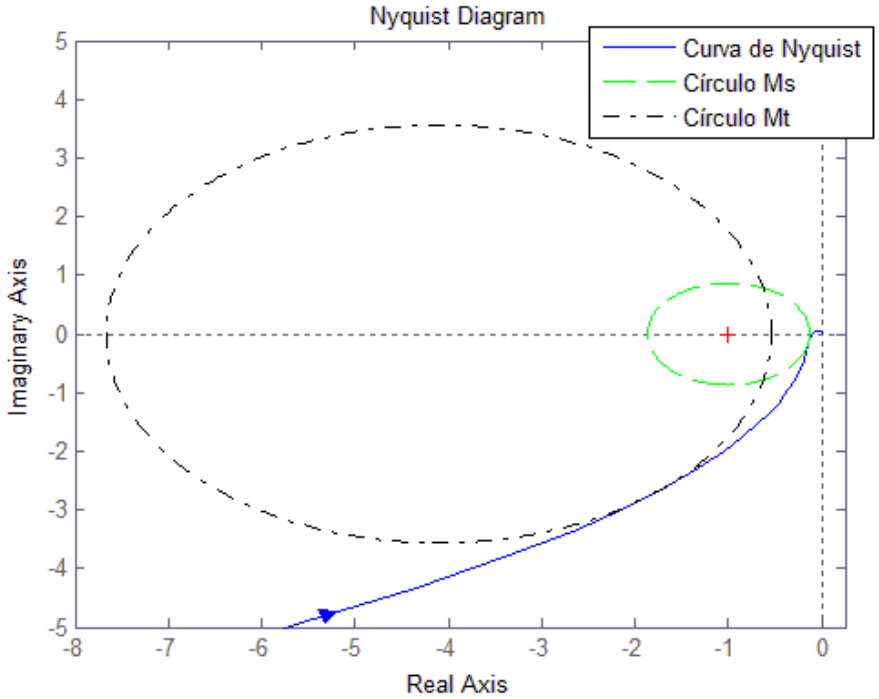

Figura 43: Diagrama de Nyquist da solução do problema de otimização utilizando o método de sintonia ótima.

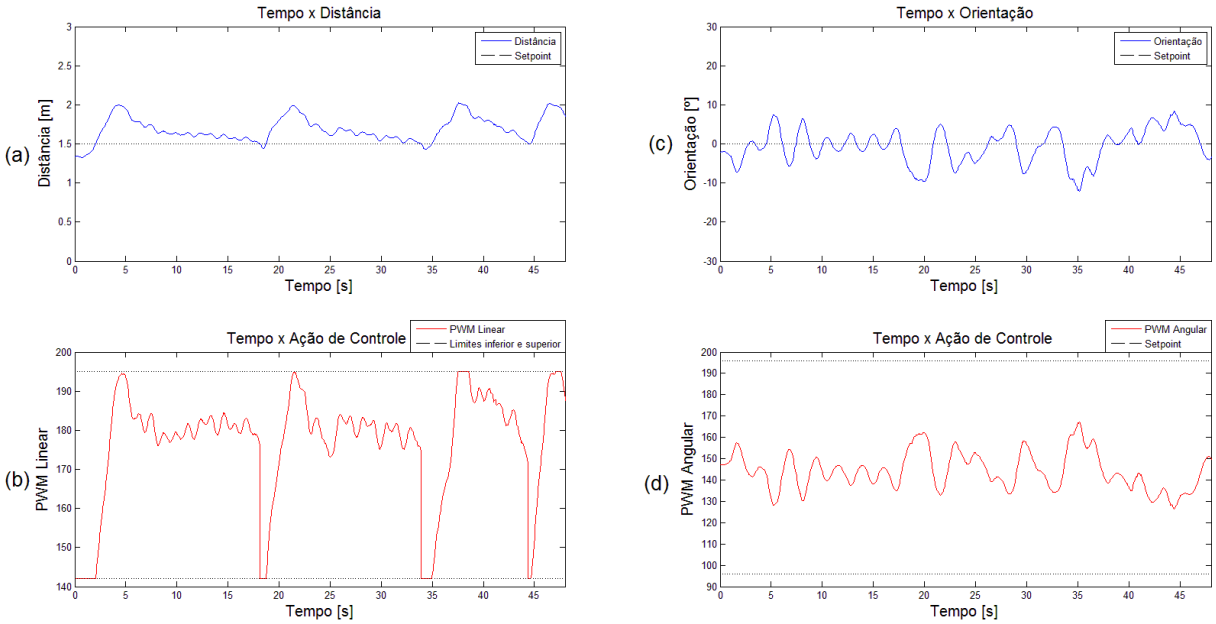

Figura 44: Ação de controle da cadeira de rodas (b, d) em relação às variáveis de processo (a, c) utilizando os parâmetros calculados pelo método de sintonia ótima.

5.3 Análise dos resultados

Analisando os comportamento das variáveis de processo e das ações de controle em relação aos valores do parâmetro K_P dos controladores P para os modelos de distância e orientação da cadeira de rodas (Figura 42- a), é possível observar que existe um erro em regime permanente da distância entre o Kinect e a cadeira de rodas. Como esperado, esse erro diminui à medida que o valor de K_P é aumentado, tornando mais agressiva a ação de controle. Isso porque a ação de controle puramente proporcional não é capaz de eliminar completamente o erro estacionário, sendo necessário a adição de uma ação integradora.

Para eliminar o erro em regimente permanente da distância entre o Kinect e o indivíduo, foi adicionada uma ação integradora ao controlador referente à malha de distância da cadeira de rodas. Sendo assim, é possível observar que nos instantes $T = 17{,}2s$, $T = 34s$ e $T = 43{,}7s$ da pista (Figura 40) o erro foi anulado (Figura 44 - a). Isso porque a parcela de contribuição integradora atua na acumulação dos erros passados, aumentando a ação de controle até que o erro seja zerado. Devido à modificação feita no algoritmo do controlador PI, ação de controle também é anulada nestes momentos. Quando isto ocorre, a velocidade linear da cadeira de rodas é nula, mantendo-a em repouso. E isto explica porque logo após esses instantes o valor da distância cresce até um valor máximo de 2m, pois a cadeira tem que novamente vencer a inércia.

Em relação a orientação (Figura 42- b), o valor de $K_P = 1{,}5$ resultou em ação de controle suave, onde o erro máximo de orientação foi de 18º. O valor de $K_P = 2{,}5$ resultou em uma ação

de controle mais agressiva, onde o erro máximo foi de 13º. Por fim, o valor de $K_P = 2$ resultou em uma ação de controle intermediária, onde o erro máximo foi de 10º.

É importante ressaltar que o sensor Kinect é muito sensível a vibrações e possui um campo de visão estreito. Logo, ações de controle muito suaves eventualmente causam a perda de rastreamento da pessoa seguida, e, ações muito agressivas, geram maiores vibrações para o Kinect, também resultando na perda de rastreamento. Tendo em mente estas restrições, o valor de $K_P = 2$ foi adotado para o controlador referente à malha de orientação da cadeira de rodas.

6 CONCLUSÃO

A aplicação desenvolvida neste trabalho visa promover o acesso de cadeirantes com limitações a cadeira de rodas elétrica, proporcionando maior conforto e melhor qualidade de vida tanto para os usuários, quanto para as pessoas que os assistem.

Com os resultados obtidos, foi concluído que o sistema de controle de navegação desenvolvido obteve um bom desempenho. Os controladores referentes aos modelos de distância e orientação mostraram ser bastante eficazes na tarefa de seguir uma pessoa. O algoritmo de colisão implementado conseguiu detectar os obstáculos, evitando colisões com os mesmos. Ainda, o sistema desenvolvido é flexível e pode ser implementado em qualquer cadeira de rodas elétrica que seja comandada por sinais de tensão.

O ROS e os dispositivos utilizados também mostraram ser bastante eficazes. Porém, algumas limitações foram observadas quando o sistema foi testado em ambientes com exposição à luz solar e solos irregulares, já que o Kinect foi desenvolvido para funcionar de maneira estática em ambientes.

O sistema apresentado neste trabalho pode ser utilizado na automatização de carrinhos de bebê, carrinhos de supermercados, cortadores de grama, guias autônomos em museus, e outros dispositivos robóticos de aplicação doméstica.

Como trabalhos futuros há duas linhas possíveis de pesquisa. A primeira seria o desenvolvimento de um produto comercial de baixo custo. A outra consiste em desenvolver um sistema de navegação totalmente autônomo.

Os trabalhos com a cadeira de rodas tiveram início na disciplina de Robótica do curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto, ministrada pelo professor José Alberto Naves Cocota Junior. Este trabalho foi apresentado na IX Mostra de Profissões na Universidade Federal de Ouro Preto e no Fórum de Ciência e Tecnologia da Escola de Minas (FCTEM). Ainda, este trabalho foi premiado com menção honrosa pela Sociedade Mineira de Engenheiros no 24º Prêmio SME de Ciência, Tecnologia e Inovação. Os integrantes do grupo que receberam a premiação são os acadêmicos Alexandre Souza Santos, Rodrigo Marcos Venâncio, Eduardo Filipe Pereira Maciel, Felipe Seguchi Orlette, Sandro Geraldo Alves Sobreira e Allan Alves Colona.

7 REFERÊNCIA BIBLIOGRÁFICA

ALCANTRA, C. A.; ALVES, F.; TEIXEIRA, P. M. K. N.; VANESSA, B.; PASSOS, W. **Medição de Distância dos obstáculos utilizando sensores de infravermelho e plataforma arduino**, Savlador, Bahia, 2013. Disponível em: <http://www.ebah.com.br/content/ABAAAf8gYAF/medicao-distancia-dos-obstaculos-utilizando-sensores-infravermelho-plataforma-arduino#comments>. Acesso em: 17 jan. 2017.

ÅSTRÖM, K. J.; HÄGGLUND, T. Revisiting the Ziegler–Nichols step response method for PID control. **Journal of process control**, v. 14, n. 6, p. 635-650, 2004.

BARBOSA, A. O. G; FREITAS, D. Z; GUEDES, J. Q. M; MEGGIOLARO, M. A. Implementation of a wheelchair control using a four-command brain computer interface. **22nd International Congress of Mechanical Engineering (COBEM 2013)**, Ribeirão Preto, São Paulo, 2013.

BECKER, M. **Aplicação de tecnologias assistivas e técnicas de controle em cadeiras de rodas inteligentes**. 2000. 136f. Tese (Doutorado em Engenharia Mecânica) – Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas, São Paulo, 2000.

BERSCH, R. Introdução à Tecnologia Assistiva. **Porto Alegre: CEDI**, 2008.

BERSCH, R; RONOLLI, J. **Introdução ao conceito de tecnologia assistiva e modelos de abordagem de deficiência**, 2006. Disponível em: <http://www.bengalalegal.com/tecnologia-assistiva>.

BOHRER, M. **Como funciona o Kinect**, 2016. Disponível em: <https://www.oficinadanet.com.br/post/11069-como-funciona-o-kinect>. Acesso em: 23 jan. 2017.

CENSO DEMOGRÁFICO 2010. Características gerais da população, religião e pessoas com deficiência. Rio de Janeiro: **IBGE**, 2010. Disponível em: <http://biblioteca.ibge.gov.br/visualizacao/periodicos/94/cd_2010_religiao_deficiencia.pdf> . Acesso em: 23 jan. 2017.

CHANG, C.-L.; CHEN, C.-C.; CHEN. C.-Y.; LIN, B.-S. Kinect-based Powered Wheelchair Control System. In: IEEE. **2013 4th International Conference on Intelligent Systems, Modelling and Simulation**. [S.I], 2013. p. 186-189.

CHEN, D.; SEBORG, D. E. PI/PID controller design based on direct synthesis and disturbance rejection. **Industrial & engineering chemistry research**, v. 41, n. 19, p. 4807-4822, 2002.

CHIEN; HRONES; RESWICK. On the automatic tuning of generalized passive systems. **Trans. Asme**, 74, 175–185 (1952)

COHEN, G. H.; COON, G. A. Theoretical consideration of retarded control. **Trans. Asme**, 75, pp. 827-834.

CURTISS-WRIGHT. **JC2000 – MULTI-AXIS CONTACTLESS JOYSTICK CONTROLLER**. JC2000 Product Data Sheet. 2015.

EUZÉBIO, T. A. M.; LIMA, M. F.; BARROS, P. R. Aplicação de Sintonia Ótima de Controlador PI em Malha de Refluxo de Topo de Coluna de Destilação. In: XIX Congresso Brasileiro de Automática (CBA), 2012, Campina Grande. **Anais do XIX Congresso Brasileiro de Automática (CBA), 2012.**

FERREIRA, A.; SILVA, R. L.; FILHO, T. F. B.; FILHO, M. S. Cadeira de rodas robótica com interface de comunicação por PDA comandada por sinais cerebrais. **Anais do VIII Simpósio Brasileiro de Automação Inteligente**, UFSC, Florianópolis, 2007.

G1. **Dona da Vespa cria 'mala ambulante' autônoma para levar compras**. Disponível em: <http://g1.globo.com/carros/motos/noticia/dona-da-vespa-cria-mala-ambulante-autonoma-para-levar-compras.ghtml>. Acesso em: 09 jan. 2017.

HOKUYO. **Scanning Laser Range Finder UTM-30LX/LN: Specification**. 27 nov. 2012. 6 p.

IFMELETRONICLTDA. **Sensores de distância**. Disponível em: <http://www.ifm.com/ifmbr/web/pmain/010_070_030.html>. Acesso em: 17 jan. 2017.

KRISTIANSSON, B.; LENNARTSON, B. Robust tuning of PI and PID controllers: using derivative action despite sensor noise. In: IEEE. **Control Systems**, v. 26, n. 1, p. 55-69, 2006.

LI, R; OSKOEI, M. A.; MC-DONALD-MAIER, K. D.; HU, H. Ros based multi-sensor navigation of intelligent wheelchair. In: IEEE. **Emerging Security Technologies (EST), 2013 Fourth International Conference on**. [S.I], 2013. p. 83-88.

LOURENÇO, J. Sintonia de Controladores P.I.D. **Escola superior de tecnologia**, 1997.

MARCHI, J. **Navegação de robôs móveis autônomos: estudo implementação de abordagens.** 2001. 119f. Dissertação (Mestrado em Engenharia Elétrica) – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Santa Catarina, Florianópolis, Santa Catarina, 2001.

MASUMOTO, Y. **Global positioning system**. U.S. Patent n. 5,210,540, 11 maio. 1993. Disponível em: <https://www.google.com/patents/US5210540>. Acesso em: 18 jan. 2017.

MATLAB. **The Language of Technical Computing**. Disponível em: <https://www.mathworks.com/products/matlab.html>. Acesso em: 25 jan. 2017.

NASRI, Y.; VAUCHEY, V.; KHEMMAR, R.; RAGOT, N.; SIRLANTZIS, K.; ERTAUD, J.-Y. ROS-based Autonomous Navigation Wheelchair using Omnidirectional Sensor. **International Journal of Computer Applications**, v. 133, n. 6, p. 12-17, 2016.

NGELÉTRICA. **A importância da automação na indústria e setores produtivos**. Disponível em: <http://www.ngeletrica.com.br/blog/a-importancia-automacao-industria-setores-produtivos>. Acesso em: 17 jan. 2017.

PEPPERL+FUCHS. **Os sensores ultrassônicos tornaram-se produtos padrão**. Disponível em: <http://www.pepperl-fuchs.com.br/brazil/pt/16122.htm>. Acesso em: 18 jan. 2017.

QUIGLEY, M; CONLEY, K.; GERKEY, B.; FAUST, J.; FOOTE, T.; LEIBS, J.; NG, A. Y. ROS: an open-source Robot Operating System. **In *ICRA workshop on open source software***, v. 3, n. 3.2, p. 5.

RADABAUGH, M. P. NIDRR's Long Range Plan-Technology for access and function research section two: NIDDR Research Agenda Chapter 5: Technology for access and function. **United States**, 1993.

RIBEIRO, M. I. **Localização em Robótica Móvel Odometria**. Lisboa, Portugal, 1999. Disponível em: <http://users.isr.ist.utl.pt/~mir/cadeiras/robmovel/Odometry.pdf>. Acesso em: 30 jan. 2017.

RIVERA, D. E.; MORARI, M.; SKOGESTAD, S. Internal model control: PID controller design. **Industrial & engineering chemistry process design and development**, v. 25, n. 1, p. 252-265, 1986.

ROBOTSTOP. **Hokuyo UTM-30LX Scanning Laser Rangefinder**. Disponível em: <http://www.robotshop.com/en/hokuyo-utm-03lx-laser-scanning-rangefinder.html>. Acesso em: 23 jan. 2017.

SCHNEIDER, D. G.; SILVA, L. L.; DIEHL, P.; LEITE, A. H. R.; BASTOS, G. S. Robot Navigation by Gesture Recognition with ROS and Kinect. In: IEEE. **Robotics Symposium (LARS) and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR), 2015 12th Latin American**. [S.I], 2015. p. 145-150.

SICILIANO, B.; KHATIB, O. Sonar Sensing. **Springer handbook of robotics**. Springer Science & Business Media, 2008. p.491-519.

SKOGESTAD, S. Simple analytic rules for model reduction and PID controller tuning. **Journal of process control**, v. 13, n. 4, p. 291-309, 2003.

SKOGESTAD, S.; GRIMHOLT, C. The SIMC method for smooth PID controller tuning. In: **PID Control in the Third Millennium**. Springer London, 2012. p. 147-175.

SMITH, Carlos A.; CORRIPIO, Armando B. **Principles and practice of automatic process control**. New York: Wiley, 1985.

TRIGUEIROS, P.; RIBEIRO, A. F. Vision-based hand Wheel-chair control. **Revista Robótica**, v. 97, p. 8-11, 2015.

XSENS. **MTi User Manual: MTi 10-series and MTi 100-series**. Document MT0605P, Revision I, 20 dec. 2016. 81p.

ZANETELLO, F. **People tracking and following with smart wheelchair using omnidirectional camera and a RGB-D camera**. Tese (Mestrado em Engenharia da Computação), Shanghai Jiao Tong University, Xangai, China, 2013/2014.

ZHUANG, M.; ATHERTON, D. P. Automatic tuning of optimum PID controllers. In IEEE. **Proceedings D-Control Theory and Applications**. IET, 1993. p. 216-224.

ZIEGLER, J. G.; NICHOLS, N. B. Optimum settings for automatic controllers. **Trans. Asme**, v. 64, n. 11, 1942.

8 ANEXOS

ANEXO I – Programação do ROS para o controle da cadeira de rodas

```
/* Declaração das bibliotecas */
#include "ros/ros.h"
#include "std_msgs/String.h"
#include "std_msgs/Bool.h"
#include <geometry_msgs/Vector3.h>
#include <cadeira/vec_ultrassom.h>
#include <cadeira/controle.h>
#include <math.h>
#include <custom_msgs/velocityEstimate.h>
#include <fstream>
#include <iostream>
#include <custom_msgs/orientationEstimate.h>
#include <sensor_msgs/LaserScan.h>
#include "pid.h"
#include <boost/thread/mutex.hpp>

/* Declaração da classe e das variáveis */
using namespace std;
class Subscriber_Publisher{
private:
	float deslocamento, angular, linear, val_x, val_z,  aux_linear, aux_angular, pwm_linear,
	pwm_angular;
	bool falha, obstaculo;
	int cont, aux_time, i, cont_obstaculo;
	ros::NodeHandle n;
	ros::Publisher pub_pwm;
	ros::Subscriber sub;
	ros::Subscriber vel;
	ros::Subscriber pos;
	ros::Subscriber laser;
	cadeira::controle con1;
	cadeira::controle con2;
	boost::mutex mtx_spin;
public:
	double begin;
	ros::Time time;
	ofstream myfile;
	double secs;
	double now;
	double inc_angular,inc_linear;
	double SP_angular, SP_linear;
	PID pid_angular = PID(0.025, 53, -47, 2.0, 0.0, 0.0);
	PID pid_linear = PID(0.025, 0, -53, 87.8, 0.0, 16.1);

	Subscriber_Publisher()
	{
```

```
  sub = n.subscribe("posicao", 1000, &Subscriber_Publisher::messageCb, this);
  pub_pwm = n.advertise<cadeira::controle>("/controle", 1000);
  laser = n.subscribe("/scan",1000, &Subscriber_Publisher::messageCb_laser, this);
  falha = true;
  obstaculo = true;
  cont = 0;
  aux_linear = 0;
  aux_angular = 0;
  aux_time = 0;
  SP_angular = 0;
  SP_linear = 1.5;
  pwm_angular = 143;
  pwm_linear = 142;
}

/* Função que verifica se a pessoa está sendo rastreada pelo Kinect */
void Falha()
{
  if ((linear >= 0) && (linear <= 4) && (angular >= -30) && (angular <= 30))
  {
    if ((aux_linear == linear)&&(aux_angular = angular))
     {
       if (cont < 12)
           cont++;
       if (cont >= 10)
           falha = true;
      }
     else
      {
       cont = 0;
       falha = false;
      }
    }
    else
       falha = true;

  aux_linear = linear;
  aux_angular = angular;
}
/* Função de controle */
void Controle()
{
  inc_angular = pid_angular.calculate(SP_angular, angular);
  pwm_angular = 143 + inc_angular;
  mtx_spin.lock();
  if (obstaculo == false)
   {
    inc_linear = pid_linear.calculate(SP_linear, linear);
    pwm_linear = 142 - inc_linear;
    }
```

```
  else
      pwm_linear = 142;
      mtx_spin.unlock();
 }
/* Função de detecção de obstáculos */
void messageCb_laser(const sensor_msgs::LaserScan& laser)
{
  for (i = 120; i<600; i++)
  {
     if (laser.ranges[i] < 0.7)
     {
        cont_obstaculo++;
       break;
     }
  }
  mtx_spin.lock();
  if (cont_obstaculo == 0)
      obstaculo = false;
  else
  {
     obstaculo = true;
     cont_obstaculo = 0;
   }
   mtx_spin.unlock();
 }

 /* Função que recebe as coordenadas do torso da pessoa*/
 void messageCb(const geometry_msgs::Vector3& deslocamento)
 {
   val_x = deslocamento.x; // -1.2 a 1.2
   val_z = deslocamento.z; // 0.7 a 3
   linear = sqrt(val_x*val_x + val_z*val_z);
   angular = atan(val_x/val_z)*180/M_PI; //
  Falha();
  if (falha == false)
      Controle();
  con1.falha = falha;
  con1.pwm_angular = pwm_angular;
  con1.pwm_linear = pwm_linear;
  pub_pwm.publish(con1);
 };

 /* Função que inicia o nó cadeira*/
int main(int argc, char **argv)
{
  ros::init(argc, argv, "cadeira");
  Subscriber_Publisher object;
  ros::spin();
  return 0;
 }
```

ANEXO II – Programação do Arduino para controle cadeira de rodas

```
#if (ARDUINO >= 100)
#include <Arduino.h>
#else
#include <WProgram.h>
#endif
#include <ros.h>
#include <geometry_msgs/Vector3.h>
#include "/home/alexandre/sketchbook/libraries/ros_lib/cadeira/controle.h"
#include "/home/alexandre/sketchbook/libraries/ros_lib/cadeira/vec_ultrassom.h"

ros::NodeHandle nh;
cadeira::controle controle;
bool falha = true;
float  pwm_angular, pwm_linear, aux_angular = 0, aux_linear = 0;
int cont = 0;
void pwmCb( const cadeira::controle& controle)
{
  falha = controle.falha;
  pwm_linear = controle.pwm_linear;
  pwm_angular = controle.pwm_angular;
}
ros::Subscriber<cadeira::controle> sub("controle", pwmCb);
ros::Publisher pub("/ultra", &vet);

#define rele_ed  9     // Relé esquerda/direita
#define rele_ft  10    // Relé frente
#define pwm_ed 11  // PWM esquerda/direta
#define pwm_ft 12   // PWM frente

void setup()
{
  Serial.begin(57600);
  pinMode(rele_ft, OUTPUT);
  pinMode(rele_ed, OUTPUT);
  pinMode(pwm_ft, OUTPUT);
  pinMode(pwm_ed, OUTPUT);
  pinMode(pwPinRX, OUTPUT);
  pinMode(pwPinA, INPUT);
  pinMode(pwPinB, INPUT);
  pinMode(pwPinC, INPUT);
  pinMode(pwPinD, INPUT);
  pinMode(pwPin4, INPUT);
  delay(1000);
  nh.initNode();     // Inicia o nó do Arduino
  delay(1000);
  nh.subscribe(sub); // Recebe o tópico posicao do ROS
delay(1000);
```

```
/* Configura a frequência dos pinos 11 e 12 do Arduino para 1KHz */
int myEraser = 7;
TCCR1B &= ~myEraser;
int myPrescaler = 2;
TCCR1B |= myPrescaler;

/* Desativa os relés que controlam os movimentos da cadeira */
digitalWrite(rele_ft, LOW);
digitalWrite(rele_ed, LOW);
}

/* Algoritmo que verifica se a pessoa está sendo rastreada pelo Kinect e aciona os relés que
controlam os movimentos da cadeira */
void loop()
{
  if ((aux_angular == pwm_angular)&&(aux_linear == pwm_linear))
  {
    if (cont < 12)
      cont++;
    if (cont >= 10)
      falha = true;
  }
  else
  {
    cont = 0;
    falha = false;
  }
  aux_angular = pwm_angular;
  aux_linear = pwm_linear;

  if (falha == false)
  {
    analogWrite(pwm_ft, pwm_linear);
    analogWrite(pwm_ed, pwm_angular);
    digitalWrite(rele_ft, HIGH);
    digitalWrite(rele_ed, HIGH);
  }
  else
  {
    digitalWrite(rele_ft, LOW);
    digitalWrite(rele_ed, LOW);
  }
  delay(20);
  nh.spinOnce();
}
```