UNIVERSIDADE DE CAXIAS DO SUL - UCS
CAMPUS UNIVERSITÁRIO DA REGIÃO DOS VINHEDOS - CARVI
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIAS
CURSO DE ENGENHARIA ELÉTRICA

VINICIUS KLERING BOSCHETTI

NAVEGAÇÃO ASSISTIVA DE CADEIRA DE RODAS EM PASSAGEM POR PORTAS COM O USO DO SENSOR KINECT

BENTO GONÇALVES
2019

VINICIUS KLERING BOSCHETTI

NAVEGAÇÃO ASSISTIVA DE CADEIRA DE RODAS EM PASSAGEM POR PORTAS COM O USO DO SENSOR KINECT

Trabalho de conclusão II apresentado na Universidade de Caxias do Sul como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica.

Orientador: Prof. Me. Patric Janner Marques

BENTO GONÇALVES
2019

VINICIUS KLERING BOSCHETTI

NAVEGAÇÃO ASSISTIVA DE CADEIRA DE RODAS EM PASSAGEM POR PORTAS COM O USO DO SENSOR KINECT

Trabalho de conclusão II apresentado na Universidade de Caxias do Sul como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica.

Orientador: Prof. Me. Patric Janner Marques

Aprovado em ____/_____/_____

Banca Examinadora:

Examinadores:

_____
Prof. Me. Patric Janner Marques
Universidade de Caxias do Sul - UCS

_____
Prof. Me. Felipe Augusto Tondo
Universidade de Caxias do Sul - UCS

_____
Prof. Dra. Marilda Machado Spindola
Universidade de Caxias do Sul - UCS

AGRADECIMENTOS

Agradeço aos meus pais, irmã e demais familiares pelo incentivo e palavras de apoio durante todo processo de graduação.

Agradeço aos colegas pelo apoio e auxílio em momentos de dúvida.

Ao Prof. Me. Patric Janner Marques, pela confiança em mim depositada e paciência, orientando este trabalho com sabedoria.

“Só se vê bem com o coração,

o essencial é invisível aos olhos”

Antoine de Saint Exupéry

RESUMO

Há um elevado número de pessoas que enfrentam dificuldades em seu dia a dia devido às limitações impostas por sua condição física, essa estimativa motiva pesquisas voltadas a tecnologias assistivas, em prol da autonomia e independência funcional nas interações sociais e familiares dessas pessoas. Diante deste cenário o trabalho apresenta um sistema de navegação assistiva para uma cadeira de rodas motorizada, na tarefa de realizar o cruzamento de portas ou passagens estreitas de maneira segura. O sistema detecta e localiza portas através da aquisição de imagens de profundidade obtidas por uma câmera RGB-D Kinect v2, e com estas informações realiza o controle da trajetória até a conclusão da travessia. As técnicas presentes neste trabalho fundamentam-se em pesquisas de trabalhos já implementados, a partir de conceitos sobre navegação de robôs e sensores de profundidade. A técnica implementada neste trabalho consiste em detectar uma porta através da diferença de profundidade presente entre a parede e seu vão, identificando seus extremos esquerdo e direito. Desta forma determina-se a localização da cadeira de rodas em coordenadas x, y e um ângulo de orientação em relação a um ponto de passagem no centro da porta e a partir daí realiza o controle da trajetória até este ponto. O processo de controle da trajetória utiliza um algoritmo baseado em lógica Fuzzy atuando na velocidade angular da cadeira de rodas, controlando sua posição ao longo da trajetória, e assim realizando o movimento de forma suave até a transpassar a porta detectada. Foram realizados testes funcionais do algoritmo de detecção da portas, a fim de validar o funcionamento e estabelecer suas limitações. Posteriormente foram realizadas simulações a partir de diferentes posições iniciais da cadeira de rodas, verificando se o sistema é capaz de identificar a porta e realizar o movimento livre de colisões. Por fim foram realizados teste do sistema implementado em uma ambiente controlado, com a cadeira disposta em diversas posições em frente a uma porta, de modo a avaliar o desempenho do sistema de detecção e localização da porta assim como o trajeto realizado pelo controlador Fuzzy. O sistema se mostrou capaz de detectar portas e realizar o controle do movimento da cadeira de rodas até o ponto de passagem, mesmo que a trajetória realizada pela cadeira tenha um desvio em relação a trajetória simulada, o controlador foi capaz de corrigir o movimento não impactando de forma significativa no resultado final.

Palavras-chave: Cadeira de rodas. Câmera RGB-D. Localização. Controle de trajetória. Lógica Fuzzy. Acessibilidade.

ABSTRACT

There is a large number of people who face difficulties in their daily lives due to the limitations imposed by their physical condition. This estimate motivates research focused on assistive technologies, all for autonomy and functional independence in their social and family interactions. Bearing this scenario in mind the work presents an assistive navigation system for a motorized wheelchair, in the task of safely crossing either doors or narrow passages. The system detects and locates doors through the acquisition of depth images obtained by a RGB-D Kinect v2 camera, and with this information it controls the trajectory until the crossing is completed. The techniques present in this work are based on research of works already implemented, drawing on concepts about robot navigation and depth sensors The technique implemented in this work consists of detecting a door through the difference in depth between the wall and its span, identifying its left and right ends. In this way, the location of the wheelchair is determined in x, y coordinates and an angle of orientation in relation to a point of passage in the center of the door and from there the trajectory is controlled to this point. The trajectory control process uses a fuzzy-logic based algorithm acting on the angular speed of the wheelchair, controlling its position along the trajectory, and thus performing the movement in a smooth manner until it passes through the detected door. Functional tests of the door detection algorithm were performed in order to validate the operation and establish its limitations. Subsequently, simulations were performed from different initial positions of the wheelchair, checking whether the system is able to identify the door and perform the collision-free movement. Finally, testing of the system implemented in a controlled environment was performed, with the chair arranged in various positions in front of a door, in order to evaluate the performance of the detection system and location of the door as well as the path taken by the fuzzy controller. The system was able to detect doors and control the movement of the wheelchair to the point of passage, even if the trajectory performed by the chair has a deviation from the simulated trajectory, the controller was able to correct the movement without significantly impacting the final result.

Keywords: Wheelchair. RGB-D camera. Location. Trajectory control. Fuzzy Logic. Accessibility

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO E JUSTIFICATIVA

Segundo o Relatório Mundial sobre a Deficiência realizado pela *World Health Organization* (WHO, 2011), estima-se que existam 1 bilhão de pessoas em todo o mundo com alguma deficiência. Deste número, entre 110 e 190 milhões vivem com dificuldades funcionais consideráveis.

Com base nessas informações, percebe-se o elevado número de pessoas que enfrentam dificuldades em seu dia a dia devido às limitações impostas por sua condição física. Em contrapartida, as pesquisas em Tecnologias Assistivas (TA) vem crescendo em prol da busca por formas de auxílio às pessoas portadoras de deficiência (BRAGA, 2010).

A Tecnologia Assistiva inclui qualquer equipamento que contém dispositivos assistivos, adaptativos e de reabilitação para pessoas com alguma deficiência (DESAI et al. 2017). Segundo Bittencourt et al. (2016), a Tecnologia Assistiva proporciona às pessoas a autonomia e a independência funcional nas interações sociais e familiares, da mesma maneira que facilita a inclusão à educação e ao mercado de trabalho. Como exemplos de equipamentos para atender os deficientes motores, cita-se: muletas, próteses, órteses, cadeiras de rodas e triciclos (WHO, 2011).

As cadeiras de rodas são os dispositivos mais utilizados para melhorar a mobilidade de pessoas com dificuldade de locomoção. Mesmo que as necessidades de grande parte dos indivíduos com deficiência possam ser satisfeitas com o uso das cadeiras de rodas tradicionais ou motorizadas, em alguns casos de deficiência severa, ou falta de habilidade motora, os usuários encontram dificuldades ou impossibilidades em conduzir estes tipos de cadeiras, principalmente em situações específicas, como subir uma plataforma elevada ou mesmo na passagem por portas. Nestes casos, os portadores de deficiência constantemente necessitam do auxílio de outros indivíduos para conduzir a cadeira de forma correta (BRAGA, 2010).

Segundo Braga (2010), o uso de tecnologias desenvolvidas inicialmente para robôs móveis são utilizadas para originar as cadeiras de rodas inteligentes. As cadeiras de rodas inteligentes, por vez, são projetadas para proporcionar auxílio de

navegação ao usuário de várias formas, como no caso de evitar colisões, auxiliar na condução da cadeira por passagens estreitas (por exemplo, por portas e elevadores) e, ainda, ser capaz de realizar o transporte autônomo e adequado do usuário entre diferentes locais.

A discussão de Zhou et. al. (2015) aponta que a detecção e localização de portas em uma navegação autônoma interna é de fundamental importância na identificação do caminho a ser percorrido, o que se aplica às cadeiras inteligentes. Entre os métodos de detecção e localização de portas destacam-se os que utilizam informações ultrassônicas, sonar e vídeo. No entanto, as ondas sonoras podem ser facilmente afetadas pelo coeficiente de reflexão da porta, ocasionando um efeito negativo no resultado. Na mesma discussão, a detecção da distância entre as bordas de uma porta também pode ser realizada com um medidor a laser, contudo a utilização de um motor de alta precisão necessário para esta aplicação apresenta-se como algo de custo elevado (YUAN et. al., 2015).

Ainda sobre a detecção de portas e corredores, as técnicas de processamento de imagem vem recebendo recentemente destaque na literatura, como em Kakillioglu, Ozcan, e Velipasalar (2017) e Quintana et. al. (2018). Uma porta pode ser modelada com base nas características de suas bordas e cantos, ou mesmo pelas informações de cor. No entanto, estes métodos podem ser limitados devido às diferentes formas e tipos de portas, ou devido a semelhança de cores dos objetos em sua volta. Sendo assim, o sensor Kinect pode ser uma alternativa para realizar a detecção e localização da porta. O sensor Kinect é um dispositivo, desenvolvido para jogos eletrônicos, que é capaz de fornecer imagens de profundidade e cor, devido a combinação entre câmeras digitais e sensores infravermelho, denominado câmera *Red, Green, Blue and Depth* (RGB-D) (ZHOU et. al. 2015).

Diante do contexto apresentado, este trabalho propõe implementar a prova de conceito de um sistema de navegação assistiva de uma cadeira de rodas motorizada, que quando ativado realiza a passagem por portas de maneira autônoma. Para esse fim, serão utilizadas as informações de profundidade, obtidas pelo sensor Kinect, para detectar e localizar uma porta. Por meio das informações de

localização da porta será desenvolvido um algoritmo baseado em lógica *fuzzy* para planejar e controlar a trajetória da cadeira e realizar a passagem atrás de portas.

1.2 OBJETIVO GERAL

Este trabalho tem como objetivo principal o desenvolvimento de um sistema de navegação assistiva, capaz de realizar a passagem, de uma cadeira de rodas motorizada, através de portas. O sistema deve ser capaz de detectar e localizar uma porta através de imagens de profundidade obtidas por uma câmera RGB-D e então realizar o planejamento e controle da trajetória até a conclusão da travessia.

1.3 OBJETIVOS ESPECÍFICOS

De modo a alcançar o objetivo geral, destacam-se os objetivos específicos citados a seguir:

a. Definir o algoritmo e técnica para detecção da porta ou abertura identificada;
b. Estimar a distância da porta até a cadeira por meio do uso do Kinect;
c. Definir a técnica para realizar a trajetória que a cadeira deve seguir com base no modelo cinemático da própria cadeira;
d. Desenvolver o sistema de controle da cadeira para o momento da passagem pelas portas;
e. Testar o sistema desenvolvido e avaliar o resultado.

1.4 ESCOPO E RESTRIÇÕES

Neste tópico são listadas algumas restrições deste trabalho, tratando dos limites do mesmo diante dos objetivos discutidos.

a. O sistema não se encarregará de realizar a navegação autônoma da cadeira pelo ambiente, apenas conduzirá a cadeira através de portas quando solicitado pelo usuário. Sendo necessária que a cadeira inicialmente esteja parada quando o sistema for ativado.

b. O sistema efetuará a identificação de portas totalmente abertas e livre de obstáculos, onde pelo menos um ponto de descontinuidade possa ser observado pelo ponto de vista do Kinect.
c. A identificação das portas apenas se encarrega de identificar portas com uma largura de abertura de 0,8 a 1,5m, com suas bordas estando de 1 a 3,5m de distância do ponto focal do Kinect e dentro do seu campo de visão de 70 graus.
d. O desenvolvimento do sistema de controle e potência dos motores da cadeira de rodas não faz parte do escopo deste trabalho. Sendo assim, o sistema apenas enviará os comandos de velocidade para o controle da própria cadeira.

1.5 APRESENTAÇÃO DO TRABALHO

Com intuito de organizar os conteúdos abordados no trabalho, optou-se pela divisão em 6 capítulos, sendo que: no primeiro capítulo é realizada uma contextualização do assunto a fim de justificar a escolha do tema, bem como a apresentação do objetivo geral e dos específicos; o segundo capítulo apresenta uma revisão bibliográfica com conteúdos relacionados ao tema do trabalho; no terceiro capítulo, são apresentados trabalhos relacionados a identificação de portas e navegação de robôs; o quarto capítulo aborda os assuntos em torno da metodologia utilizada no trabalho, demonstrando o processo e as etapas de execução do trabalho; no quinto capítulo, são demonstrados os resultados atingidos; e por fim, o sexto capítulo trata das contribuições deixadas por este documento e os respectivos trabalhos futuros, dos quais poderão surgir novos desafios acadêmicos.

2 REVISÃO BIBLIOGRÁFICA

Com a finalidade de fundamentar os métodos a serem utilizados no decorrer do desenvolvimento deste trabalho, neste capítulo serão apresentados os conceitos básicos sobre navegação de cadeiras de rodas robótica, abordando os princípios de localização e sensores de profundidade. Em seguida serão apresentados métodos de planejamento e controle da trajetória.

2.1 NAVEGAÇÃO DE CADEIRA DE RODAS ROBÓTICA

Segundo Perdigão (2014), a robótica móvel é um amplo campo da engenharia e um importante ramo dentro das tecnologias assistivas, como no caso das técnicas aplicadas ao controle de cadeiras de rodas automatizadas. Na sua forma básica, um robô móvel é capaz de prover a sua própria locomoção, o que significa que ele pode se mover livremente em seu ambiente, seja por meio do uso de rodas ou pernas, de forma subaquática ou mesmo aerotransportada.

Como uma forma particular de robô móvel, a cadeira de rodas robótica é considerada um avanço tecnológico da tradicional cadeira de rodas motorizada, onde esta versão moderna tem como objetivo simplificar e auxiliar a navegação por ambientes com muitos obstáculos ou de difícil acesso (PERDIGÃO, 2014).

Nos robôs móveis, a autonomia é um aspecto de fundamental importância para executar tarefas de navegação. Um robô autônomo, ou inteligente, pode ser definido como um dispositivo eletromecânico capaz de funcionar sem a necessidade de auxílio externo, integrando capacidades como percepção, raciocínio e controle, o que lhe permite se adaptar às mudanças que ocorrem no ambiente. Essas capacidades tornam o robô capaz de observar o ambiente, tomar decisões e ações de maneira autônoma. Logo, os robôs autônomos são dispositivos que contém instrumentos para mensurar e atuar conforme premissas e controle de um sistema computacional, a fim de executar determinadas tarefas, como no caso da navegação por uma trajetória predefinida (BELO, 2006), (MARCHI, 2001).

De acordo com Marchi (2001), navegar baseia-se em planejar e executar uma trajetória de um ponto inicial até um ponto final enquanto desvia de obstáculos que

podem estar em seu caminho. Para assegurar que o processo de navegação seja executado corretamente é necessário conhecer o corpo que se busca movimentar e o ambiente no qual o movimento deve ocorrer. Quando há conhecimento do ambiente, a navegação se resume basicamente ao planejamento da trajetória. Do contrário, a navegação ocorre conforme as informações do ambiente são adquiridas. Em suma, as principais questões que englobam o problema da navegação, segundo Braga (2010), são: “Onde estou? Onde estou indo? e Como chegar lá?”

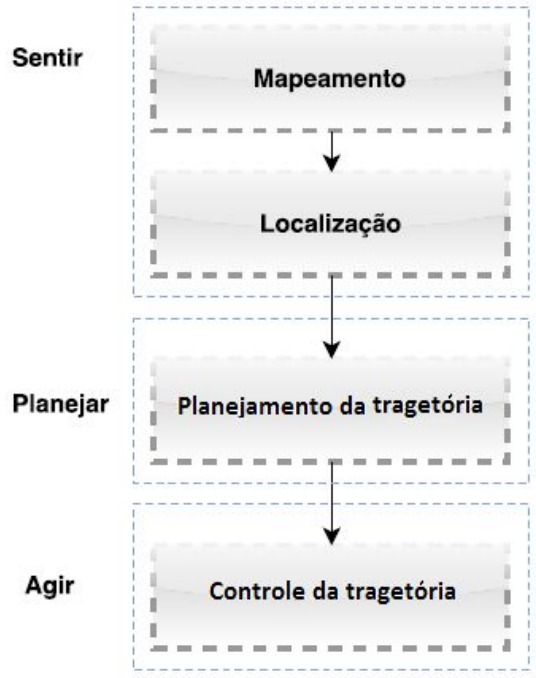
Segundo Mechi (2001) e Alves (2018), a navegação dos robôs pode ser classificada em três tipos:

a. Manual - o operador é responsável por realizar todo o controle da navegação;
b. Semi-Autônoma - o operador indica o controle a ser executado e o sistema realiza a navegação; consiste em princípios como evitar obstáculos, passar através de portas, corredores estreitos e na realização de manobras difíceis;
c. Autônoma - a navegação ocorre sem o auxílio do operador, com o sistema tomando as próprias decisões; o usuário apenas necessita indicar um objetivo ou destino e o sistema de controle assume a realização do planejamento e execução da trajetória.

Em relação à navegação autônoma e semi-autônoma de robôs móveis, é necessário que o desenvolvimento do robô seja baseado em uma arquitetura organizada em uma sequência de primitivas como: sentir, planejar e agir. Dentro destas primitivas, o processo de navegação é subdividido em quatro tarefas básicas, sendo o Mapeamento, a Localização, o Planejamento da trajetória e o Controle da trajetória, tarefas que seguem o fluxo conforme pode ser visto na Figura 1 (BRAGA, 2010), (CASTRO, 2017).

Basicamente, a tarefa do mapeamento é a tarefa de construção de mapas, sendo geralmente realizada por meio do uso de sistemas sensoriais capazes de reconhecer e modelar as principais características do ambiente. Já a localização consiste no robô encontrar sua posição em relação ao ambiente, geralmente baseando-se em informações de sensores e da representação do ambiente em forma de mapa (BRAGA, 2010), (CASTRO, 2017).

Figura 1 - Arquitetura para a Navegação.



Autor: Adaptado de Castro (2017).

Seguindo as tarefas básicas de navegação, a tarefa de planejamento consiste em determinar um caminho adequado a percorrer no ambiente, a partir de uma posição inicial e um objetivo final, evitando a colisão com os obstáculos. Por fim, a tarefa de controle é responsável pela condução do robô através de um caminho planejado, sendo necessário um conhecimento prévio do modelo cinemático e dinâmico do robô (BRAGA, 2010).

2.2 LOCALIZAÇÃO DE ROBÔS MÓVEIS

A localização é crucial para a maioria das tarefas envolvidas com a robótica móvel, sendo ela definida como um problema em torno da estimação da posição relativa do robô no ambiente em que se encontra (PERDIGÃO, 2014). O rastreamento da posição é o problema de percepção mais básico na robótica. Se um robô não sabe onde se encontra no ambiente, a tomada de decisão se torna problemática. Localizar um robô baseia-se em determinar sua posição e orientação

em um instante de tempo, obtendo as suas coordenadas *x* e *y* em um plano cartesiano, e seu ângulo *θ* de orientação em relação a um referencial fixo (CASTRO, 2017).

Com a finalidade de localizar e obter informações do ambiente ao seu redor, diferentes tipos de sensores podem ser utilizados, como por exemplo, os *encoders*, câmeras digitais, *lasers*, bússolas, gps, sonares, dentre outros. Em alguns casos é habitual a combinação de dois ou mais tipos de sensores trabalhando ao mesmo tempo, reduzindo assim erros associados a cada um deles. (BEZERRA, 2004).

Segundo Bezerra (2004) e Castro (2017), existem diversos métodos de localização de robôs móveis, cada um com suas vantagens e desvantagens. Estes métodos podem ser classificados em duas principais categorias: Localização Relativa e Localização Absoluta.

2.2.1 Localização relativa - odometria

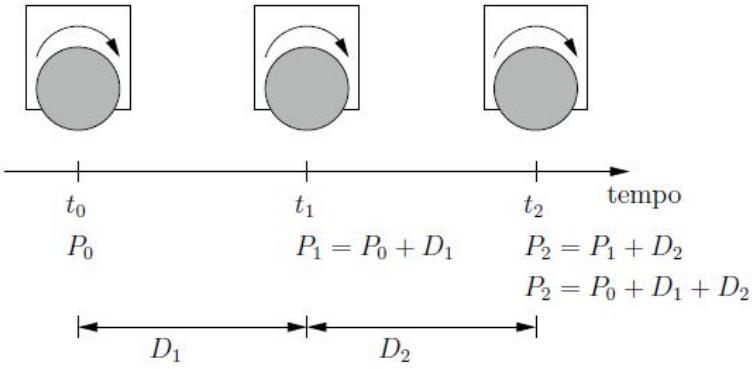
O método de Localização relativa ou *Dead Reckoning* baseia-se na estimativa da posição atual através de posições anteriores. Conhecendo a localização inicial do robô e sua velocidade é possível estimar a sua localização atual. Este método é um dos mais aplicados para estimar a posição momentânea de um robô devido a sua simplicidade. Contudo, imprecisões na localização podem ocorrer ao longo do caminho devido ao seu erro acumulativo. Na robótica móvel as localizações relativas são adquiridas por técnicas como a Odometria ou Navegação Inercial (CASTRO, 2017), (TORRES, 2018).

Segundo Bezerra (2004) e Castro (2017), a Odometria consiste em determinar a posição e orientação do robô, que são medidas a partir de um referencial fixo, através de informações dos deslocamentos incrementais das rodas do robô ao longo do tempo.

A fim de exemplificar este método, considere um robô se deslocando em uma trajetória, conforme mostra a Figura 2. No instante t_0 o robô encontra-se na posição P_0 e então inicia o deslocamento. No seguinte instante t_1 o robô encontra-se na posição P_1, que pode ser calculada com a soma da posição anterior e o deslocamento D_1, realizado entre os instantes t_0 e t_1. Em um último instante t_2 o robô

se encontra na posição P_2, que pode ser calculada com a soma da posição anterior e deslocamento D_2, realizado desde o instante t_1. Deste modo, a posição atual em que um robô se encontra pode ser calculada, com base no acúmulo dos deslocamentos em relação à sua posição inicial (BEZERRA, 2004).

Figura 2 - Deslocamento do robô utilizando odometria.



Autor: Bezerra (2004).

Para calcular o deslocamento do robô, se faz necessário o uso de sensores capazes de medir a rotação das rodas. Um tipo de sensor bastante utilizado na odometria é o *encoder* óptico. O *encoder* funciona basicamente na transmissão e recepção de luz por entre um disco perfurado, geralmente acoplado ao eixo da roda do robô (BEZERRA, 2004).

Sistemas baseados em odometria para estimar a posição são muito utilizados, por permitirem altas taxas de amostragem, serem de fácil implementação e ter boa precisão em trajetórias de curtas distâncias. Entretanto, apresenta a grande desvantagem do acúmulo de erros que aumenta de forma significativa à medida que a distância é percorrida (BRAGA, 2010) , (CASTRO, 2017).

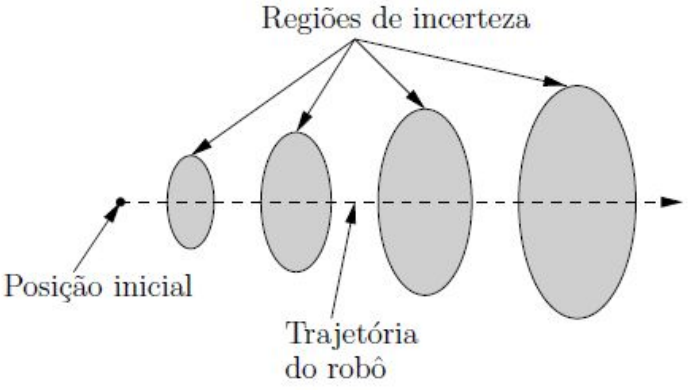
Os erros associados a odometria podem ser classificados em erros sistemáticos e erros não-sistemáticos. Erros sistemáticos são aqueles provocados por incertezas nos parâmetros que formam o modelo cinemático do robô. Este tipo de erro causa uma grande distorção na determinação da localização atual, por ocorrer durante toda a trajetória, acumulando continuamente. Já os erros não-sistemáticos são provocados por situações que surgem de maneiras

inesperadas. Diferente do erro sistemático, este tipo de erro não ocorre durante toda a trajetória, sendo mais difíceis de corrigir. As principais causas que envolvem estes tipos de erros são (BEZERRA, 2004), (BRAGA, 2010):

- Erros sistemáticos
  - Diferença de diâmetro das rodas;
  - Medida incorreta da distância entre as rodas;
  - Desalinhamento das rodas;
  - Resolução baixa do *encoder*;
- Erros não-sistemáticos
  - Terrenos irregulares;
  - Objetos inesperados no chão;
  - Escorregamento das rodas;

O erro causado pela odometria pode ser representado por uma região elíptica, que se dispõe em torno da trajetória que o robô deve seguir, como pode ser observado na Figura 3. Essa região apresenta o grau de incerteza, aumentando proporcionalmente com a distância percorrida pelo robô (BEZERRA, 2004).

Figura 3 - Região de incerteza durante o movimento.



Autor: Bezerra (2004).

Uma forma de reduzir o erro sistemático é aplicar técnicas de calibração do sistema. Outra forma de correção do erro associado a odometria é a utilização do método de localização absoluta, através de outros tipos de sensores que seriam

usados em conjunto com o odômetro, de maneira a realizar o zeramento periódico do erro acumulado em determinados pontos do ambiente. (BEZERRA, 2004).

2.2.2 Localização absoluta

Os métodos de localização absoluta proporcionam as informações de localização sem depender de medidas realizadas anteriormente, ou seja, a localização atual não é uma resultante de uma sequência de medidas e sim de uma única medida. A maior vantagem deste método em relação ao método de localização relativa é que não há o acúmulo do erro ao longo do tempo. Os métodos de localização absoluta podem ser baseados na utilização de Marcos ou pontos de referência e na utilização de mapas (CASTRO, 2017).

Segundo Castro (2017), a utilização de Marcos para a localização absoluta de robôs se baseia em detectar características no ambiente e combiná-las com informações pré estabelecidas para determinar sua posição. Os marcos se dividem em Ativos e Passivos. Nos Ativos, os marcos enviam as informações ativamente para o robô capturar, processar e se localizar. Entre os Marcos Ativos estão os que utilizam GPS (Global System Position), ondas de rádio ou ultrassom.

Já os Marcos Passivos são aqueles que o robô precisa enxergá-los no ambiente para receber informações de posição. Os marcos passivos podem ser divididos nos tipos Artificiais ou Naturais. O tipo Artificial é especificamente desenvolvido para uma fácil identificação pelo robô. São marcos previamente posicionados em locais bem visíveis aos sensores. Exemplos destes marcos são o código de barras ou figuras geométricas contendo algum padrão. Já os marcos naturais fazem parte do ambiente de operação do robô, como portas, janelas e lâmpadas. Tanto marcos Naturais quanto Artificiais, normalmente utilizam um sistema de visão, o que pode ter um custo elevado de processamento e dependem de uma leitura clara do Marco a ser detectado (CASTRO, 2017).

Outro tipo de localização absoluta é a baseada em Mapas. Essa técnica utiliza as características geométricas do ambiente, que são descritas pelas linhas das paredes portas e salas. O robô é localizado combinando as características do ambiente, que compõem o mapa, com as informações de saída de sensores, como

câmaras RGB-D, *Laser Scanners* ou sonares (CASTRO, 2017). De acordo com Braga (2010), os mapas, não somente proporcionam uma visão global do ambiente, mas também são importantes para a auto-localização do robô. Quando o mapa não está disponível, um robô pode construir o mapa enquanto se localiza no ambiente. Esta técnica é chamada de Localização e Mapeamento Simultâneos (SLAM).

2.2.3 Imagens de profundidade

De acordo com Silva (2000), as imagens de profundidade fornecem características tridimensionais (3D) da cena observada, o que é uma vantagem em relação às imagens de intensidade luminosa. Permitindo, assim, extrair diversas informações importantes do ambiente, como a classificação de borda e a orientação das superfícies. Cada *pixel* da imagem extraída pode ser representado, por um vetor com três valores, que significam as coordenadas tridimensionais *(x, y, z).*

Estas imagens podem ser obtidas por métodos como triangulação, baseada em Luz Estruturada (SL), ou métodos baseados em tempo de reflexão, chamado de *Time-of-flight* (ToF). Ambos os métodos permitem obter a distância de cada *pixel* da superfície observada até a origem de um sistema de coordenadas (SILVA, 2000).

Segundo Alenya, Foix e Torras (2014), o avanço na tecnologia das câmeras 3D possibilitou a produção de dispositivos de prateleira com potencial para aplicações de realidade virtual, vigilância e robótica, oferecendo uma alternativa de menor custo que a visão estéreo ou *laser scanning*. As câmeras baseadas em ToF surgiram em 2004, fornecidas pela Mesa Imaging e PMD Technologies, tendo como principal uso a navegação de robôs e outras tarefas de longo alcance.

Mais recentemente, as câmeras RGB-D, receberam uma atenção ainda maior pelo seu baixo custo e simplicidade. Essas câmeras são compostas por uma câmera RGB clássica e um sensor de profundidade, baseado em Luz Estruturada, inicialmente fornecidas pela empresa PrimeSense. As câmaras RGB-D oferecem dados de qualidade suficiente para a maioria das aplicações, sendo amplamente utilizadas em aplicações de robótica móvel, reconhecimento de objetos e atividades humanas (ALENYA; FOIX; TORRAS, 2014).

A Tabela 1 apresenta as características das câmeras RGB-D, voltadas ao público consumidor, com as tecnologias baseadas em ToF e das baseadas em luz estruturada.

Tabela 1 - Comparação entre câmeras RGB-D.

Especificações	Microsoft Kinect v1[1]	Asus Xtion PRO LIVE[2]	Microsoft Kinect v2[1]	Intel RealSense D435i[3]
Tecnologia sensor de profundidade	Luz Estruturada	Luz Estruturada	Time-of-flight	Active IR Stereo
Câmera RGB [pixel x pixel]	1280 x 960	1280 x 1024	1920 x 1080	1920 x 1080
Taxa de quadros RGB [fps]	30	30	30	30
Câmera profundidade [pixel x pixel]	640 x 480	640 x 480	512 x 424	1280 x 720
Taxa de quadros Profundidade [fps]	30	30	30	90
Alcance [mm]	800 - 4000	800 - 3500	500 - 4500	105 - 10000
Campo de visão H x V [grau x grau]	57 x 43	58 x 45	70 x 60	87 x 58
Alimentação	Externo	USB	Externo	USB
Interface	USB 2.0	USB 2.0/3.0	USB 3.0	USB 3.1 Gen 1
Preço	Descontinuado R$ 129,00 (usado)	Descontinuado -	Descontinuado R$ 399,00 (usado)	Importado $ 199,00

Autor: Adaptado de Pizzetta (2016).

Alenya, Foix e Torras (2014), apontam algumas diferenças entre as tecnologias de luz estruturada e ToF. Enquanto a luz estruturada não consegue realizar medições a distância menores que 0,4m, a tecnologia ToF consegue adquirir imagens a uma distância mínima de até 0,2m com uma abertura maior no campo de visão.

A tecnologia baseada em luz estruturada não é projetada para operar sob condições de iluminação forte, como ao ar livre, não conseguindo proporcionar

[1] Pagliari e Pinto (2015) e Pizzetta (2016).
[2] AsusTek Computer Inc (2019).
[3] Intel Corporation (2019).

informações de profundidade corretamente. Já as câmeras ToF fornecem informações de profundidade, porém apresentam ruídos nas partes expostas diretamente a luz solar (ALENYA; FOIX; TORRAS, 2014). Segundo Pagliari e Pinto (2015), a estabilidade e precisão nas medições utilizando a tecnologia ToF aplicadas ao Kinect v2, são superiores a luz estrutura utilizada no Kinect v1.

2.2.3.1 Luz estruturada

De acordo com Khoshelham e Elberink (2012), um sensor de luz estruturada consiste em um emissor laser infravermelho e uma câmera infravermelha. A medida de profundidade pode ser descrita com um processo de triangulação, onde o emissor laser emite um único feixe que então é dividido em múltiplos feixes, e, por meio de uma grade de difração, cria-se um padrão de manchas projetadas na cena. Então, esse padrão é capturado pela câmera infravermelha e correlacionado com um padrão de referência. O padrão de referência é estabelecido com a captura de um plano a uma distância já conhecida, este armazenado previamente na memória do sensor.

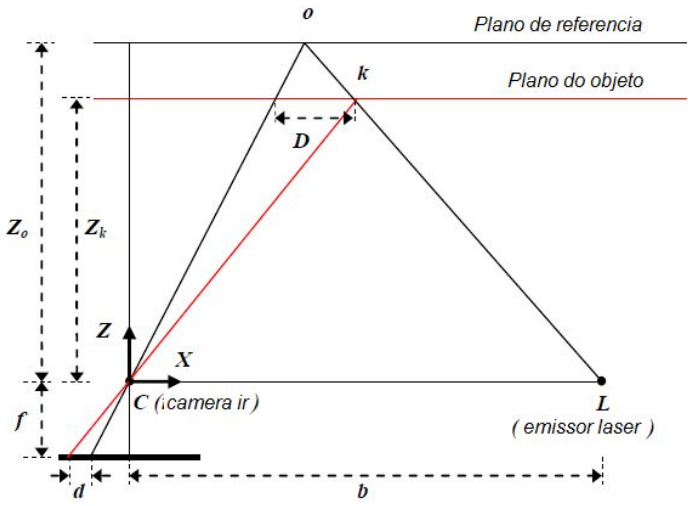
Para cada pixel, a distância até o sensor pode ser obtida da disparidade correspondente, através da correlação entre o plano de referência e o objeto. Quando um feixe é projetado em um objeto que está a uma distância menor ou maior que o plano de referência, há um deslocamento da captura deste feixe entre projetor e o centro da perspectiva da câmera infravermelha. Este deslocamento produz a imagem de disparidade (KHOSHELHAM; ELBERINK, 2012).

A Figura 4 ilustra um objeto *k* a uma distância do plano de referência, o objeto está mais próximo do sensor, logo a localização do feixe no plano da imagem será deslocada na direção X. Isso é medido no espaço da imagem como uma disparidade *d*.

Através de uma semelhança de triângulos, pode-se obter as Equações 1 e 2 e, por fim, a Equação 3 define a distância em relação a disparidade observada. Nestas equações, Z_k é distância entre o sensor e objeto, Z_0 a distância entre o sensor e o plano de referência, *f* é a distância focal da câmera, *b* é o comprimento da base, *D* o deslocamento do ponto *k* no espaço do objeto e *d* é disparidade

observada no espaço da imagem (KHOSHELHAM; ELBERINK, 2012).

Figura 4 - Relação entre distância e disparidade.



Autor: Adaptado de Khoshelham e Elberink (2012).

$$\frac{D}{b} = \frac{Z_0 - Z_k}{Z_0} \quad (1)$$

$$\frac{d}{f} = \frac{D}{Z_k} \quad (2)$$

$$Z_k = \frac{Z_0}{1 + \frac{Z_0}{f_b} d} \quad (3)$$

Vale observar que os parâmetros Z_0, *f* e *b* são constantes determinadas na calibração do sensor (KHOSHELHAM; ELBERINK, 2012).

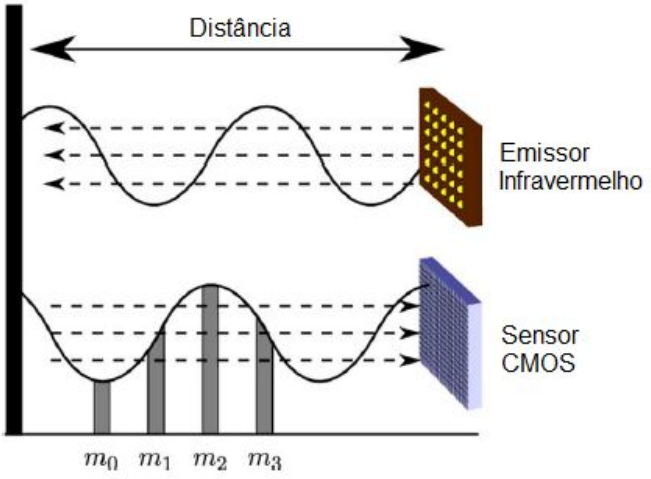
2.2.3.2 Time-of-flight

As medições de profundidade na tecnologia ToF baseiam-se na utilização da modulação por pulso ou onda contínua. Embora existam câmeras baseadas em ambas as tecnologias, as baseadas em pulso são menos utilizadas que as com onda contínua, já disponíveis no mercado a mais de uma década. Entre as de onda contínua, se destacam as que utilizam *pixel* de bloqueio de demodulação, não

importando se a demodulação é digital ou analógica (FOIX; ALENYA; TORRAS, 2011).

De acordo com Foix, Alenya e Torras (2011), a estimativa de profundidade é baseada na diferença de fase entre os sinais emitidos e recebidos. O feixe de luz infravermelho é emitido pelo sistema e ao atingir um objeto é refletido de volta para o sensor. Cada pixel do sensor indica, quatro vezes a quantidade de luz refletida pelo ambiente, com intervalos iguais para cada período (m_0, m_1, m_2, e m_3), o que possibilita a medição paralela de sua fase. A Figura 5 apresenta o princípio de funcionamento de tecnologia ToF.

Figura 5 - Medição de distância por deslocamento de fase.



Autor: Adaptado de Foix, Alenya e Torras, (2011).

O deslocamento pode ser calculado através da Equação 4. Essa técnica de demodulação de fase é corriqueiramente conhecida como amostragem de “quatro baldes” e permite calcular a profundidade alvo *D*, através da Equação 5.

$$\varphi = tan^{-1}\frac{m_3 - m_1}{m_0 - m_2} \tag{4}$$

$$D = \frac{c\varphi}{4f_m\pi} \tag{5}$$

Onde φ é o deslocamento de fase, f_m é a frequência de modulação e c é a velocidade da luz no vácuo.

2.3 PLANEJAMENTO DA TRAJETÓRIA

O planejamento da trajetória é considerado um dos principais problemas da robótica móvel. Planejar uma trajetória consiste em, a partir de uma posição inicial e um ponto de destino no ambiente, identificar o melhor caminho a ser percorrido pelo robô entre estes dois pontos. Na navegação autônoma, o robô, durante a execução de uma trajetória, deve reagir a obstáculos, redefinindo a sua trajetória a fim de evitar colisões (RODRIGUES, 2017).

Segundo Perdigão (2014), o planejamento da trajetória pode ser dividido em uma tarefa de planejamento de caminho global e uma de planejamento de caminho local. A tarefa de planejamento global basicamente estabelece o caminho da posição inicial até o ponto de destino sem se importar com os possíveis obstáculos presentes na trajetória. Já o planejamento local é um sistema reativo, baseado em sensores, com objetivo principal de evitar possíveis obstáculos e manter o robô na trajetória global.

Planejamento de caminho global são geralmente baseados em algoritmos de busca de grafos e árvores, como o A-Star (A*), Dijkstra, entre outros. Esses algoritmos são computacionalmente eficientes e proporcionam um solução teórica ideal para um determinado mapa. Em relação ao planejamento de caminho local, algumas das soluções são a abordagem de janelas dinâmicas, Trajetória Rollout e o método de campos potenciais (PERDIGAO, 2014).

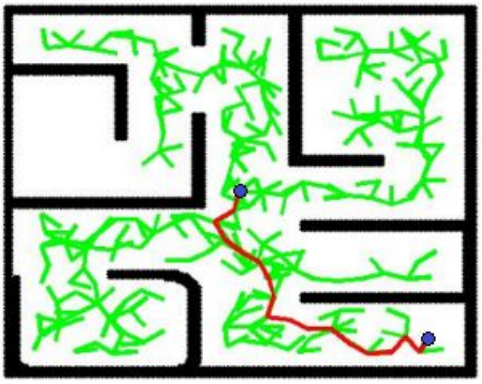
O método de campo potencial é criado considerando um ponto de destino e os obstáculos presentes no ambiente. O ponto de destino influência no campo potencial criando uma força atrativa e os obstáculos criando uma força repulsiva afastando o robô. Um campo potencial artificial é criado correspondendo ao somatório de todos os campos formados até o destino, através da influência desses campos o robô é então guiado até o objetivo (SOUSA, 2017), (RODRIGUES, 2017).

Para lidar com problemas mais complexos e de alta carga computacional, o método *Rapid-exploring Random Tree* (RRT) pode ser utilizado. Através da

amostragem aleatória, o algoritmo RRT gera uma árvore, no espaço livre, expandindo na direção do destino contornando os obstáculos.

Basicamente, o algoritmo busca encontrar um caminho ideal mais curto entre o ponto de início e o ponto de destino. Para isso, são realizadas uma quantidade de tentativas predefinidas, começando na posição inicial do robô, e a partir daí estendendo os ramos, gerados aleatoriamente, até o ponto objetivo. Quanto maior o número de tentativas mais o espaço será preenchido e maiores as chances do melhor caminho ser encontrado, porém maior será o tempo de execução e complexibilidade computacional. A Figura 6 apresenta um caminho encontrado entre dois pontos e a árvore gerada pelo RRT, considerando 200 tentativas. (SOUSA, 2017), (PRADO. 2013).

Figura 6 - Trajetória baseada em RRT.



Autor: Sousa (2017).

2.4 CONTROLE DA TRAJETÓRIA

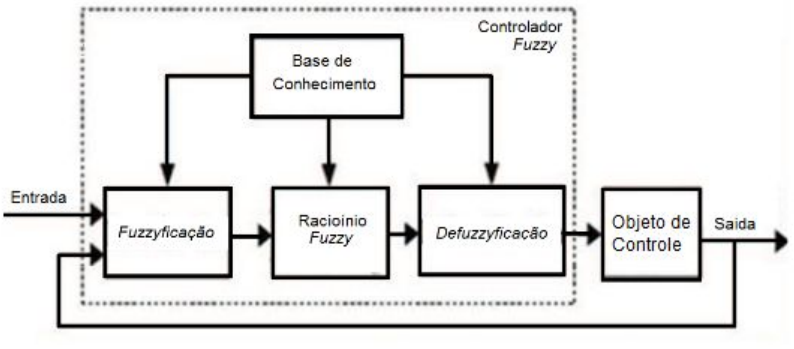
O controle de trajetória consiste em determinar as velocidades do robô a fim de seguir uma trajetória, garantindo que este cumpra, com erro mínimo, a trajetória de referência com a velocidade pretendida. Esta trajetória geralmente é constituída por um conjunto de pontos em coordenadas do ambiente. No caso de robôs diferenciais, o controlador determina as velocidades de cada uma das rodas, tendo uma velocidade linear e angular, para atingir ponto da trajetória (*x, y*) com orientação θ desejada (MAGALHÃES, 2017), (RODRIGUES, 2017).

2.4.1 Controlador *fuzzy*

De acordo com Sandeep e Supritya (2016), a lógica *fuzzy* é uma ferramenta aplicada em sistema de tomada de decisão automática, adequado para lidar com dados imprecisos e incertos. O controle de lógica fuzzy é um método de controle conveniente para muitos sistemas complexos não-lineares e aplicável na navegação de robôs móveis. A lógica fuzzy é mais intuitiva e representa a forma como os humanos pensam.

A lógica *fuzzy* é um método que admite vários valores e gera um grau de pertinência dentro de um conjunto de associações. Inicialmente as variáveis de entradas passam por um escalonamento, a fim de normalizar os valores transformando os números em variáveis linguísticas, esta etapa é chamada fuzzyficação. Após os dados serem fuzzyficados, estes são processados junto com a base de regras *fuzzy,* de maneira a inferir as ações de controle *fuzzy* a serem tomadas. Por fim, é realizado o estágio da defuzzyficação, onde as ações de controle em variáveis linguísticas são transformadas em ações de controle numéricas. A Figura 7 mostra o diagrama de blocos para um sistema baseado em *fuzzy* aplicável a robôs móveis (BECKER, 2000).

Figura 7 - Sistema *fuzzy*



Autor: Adaptado de Sandeep e Supritya (2017).

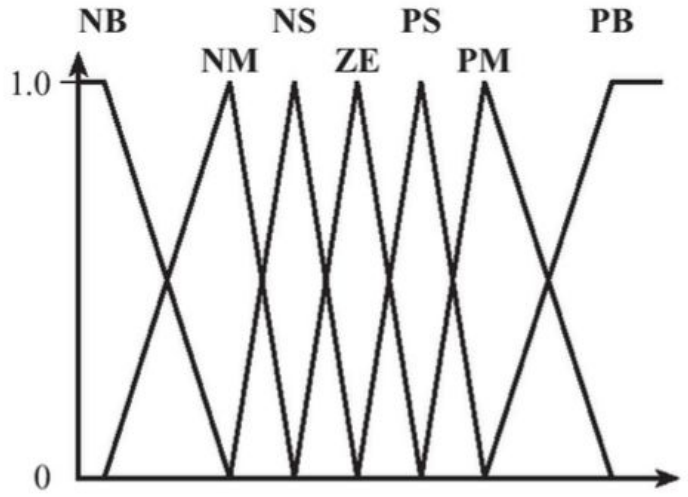
A interface de fuzzificação utiliza as funções de pertinência contidas na base de conhecimento para converter os valores numéricos em valores linguísticos. As funções de pertinência representam um aspecto fundamental para das ações do

sistema *fuzzy*. Uma função de pertinência é uma função numérica que atribui valores de pertinência *fuzzy* para valores discretos de uma variável de discurso (SIMÕES; SHAW, 2014).

As funções de pertinência geralmente são triangulares ou trapezoidais, e a quantidade de funções e o seu formato são determinadas com base na experiência e referente a natureza do processo a ser controlado, podendo ser uma tarefa trabalhosa e não trivial. Porém, há técnicas que utilizam redes neurais e algoritmos genéricos que podem gerar as funções de modo automático.

A Figura 8 apresenta as funções de pertinência de uma lógica *fuzzy*. No qual os conjuntos de pertinência *fuzzy* são nomeados de: grande negativo (NB), médio negativo (NM), pequeno negativo (NZ), zero (ZE), pequeno positivo (PS), médio positivo (PM) e grande positivo (PB) (SIMÕES; SHAW, 2014).

Figura 8 - Funções de pertinência *fuzzy*



Autor: Simões e Shaw (2014).

Segundo Simões e Shaw (2014), os sistemas *fuzzy* tem como seus dois principais componentes a estrutura e as funções de pertinência. Geralmente as estruturas do controlador *fuzzy* podem ser:

a. Controlador *fuzzy* baseado em regras;
b. Controlador *fuzzy* paramétricos;
c. Controlador *fuzzy* baseado em equações relacionais.

O controlador baseado em regras se baseia na utilização de linguagem natural, construído por modelos linguísticos, para caracterizar o comportamento do sistema. A base de regras caracteriza os objetivos e estratégias de controle, por meio de um conjunto de regras. Para cada regra se faz necessário aplicar uma técnica de agregação dos conjuntos antecedentes, para que então seja gerado um conjunto de consequentes. A formulação das regras f*uzzy* podem ser realizadas por um especialista humano articulando associações entrada/saída linguísticas, desta forma conseguir estimar um sistema complexo não-linear, sem recorrer a modelos matemáticos (SIMÕES; SHAW, 2014), (BARROS FILHO, 2011).

De acordo com Becker (2010), um controlador f*uzzy* pode navegar um veículo autônomo de uma posição inicial até uma posição de destino, expressa em termos (x, y, θ), chegando na posição desejada com o ângulo de orientação desejado.

2.5 CONTROLE SEMI-AUTÔNOMO

De acordo com Perdigão (2014), os controles autônomos são capazes de autogovernança no desempenho de suas funções, existindo diferentes graus de autonomia em função da situação em que o robô será aplicado. Já um sistema dito semi-autônomo é um sistema que requer alguma assistência externa, seja de um ser humano ou de outro sistema.

As premissas do controle semi-autônomo são oriundas dos conceitos de teleoperações e de robótica autônoma. No quesito da teleoperação, o homem e a máquina interagem nas mesmas funções do sistema. Já no conceito de robô autônomo, a inteligência é incorporada no nível da máquina, permitindo que durante a interação com o homem, o robô adapte suas ações às condições da tarefa a ser executada de forma automática.

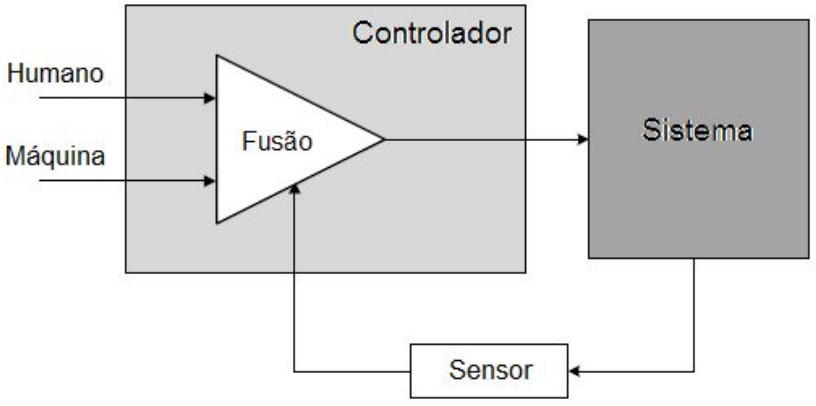
Em um sistema semi-autônomo, o robô pode operar com um certo nível de capacidade autônoma conhecida, operando na ausência de supervisão ou manejo humano por um período de tempo definido. Para isso, é necessário prover a habilidade básica e o grau de autonomia do robô que atenda a aplicação desejada, reduzindo assim o grau de supervisão humana (ONG et. al., 2005).

No que diz respeito ao controle semi-autônomo, Perdigao (2014) e Ong et. al. (2005) os classificam em três tipos, Paralelo ou Partilhado, Serial ou Negociado e uma combinação dos tipos Paralelo e Serial, chamado de Colaborativo.

2.5.1 Controle partilhado

Segundo Ong et. al. (2005), o controle do tipo Partilhado ou Paralelo é uma abordagem que incorpora as decisões do homem e do robô, permitindo que eles realizem o controle conjunto de diferentes aspectos do sistema ao mesmo tempo, conforme pode ser observado na Figura 9. Em outras palavras, parte de uma determinada tarefa é desenvolvida por uma máquina, e outra parte é desenvolvida por um agente humano.

Figura 9 - Malha de Controle Partilhado.



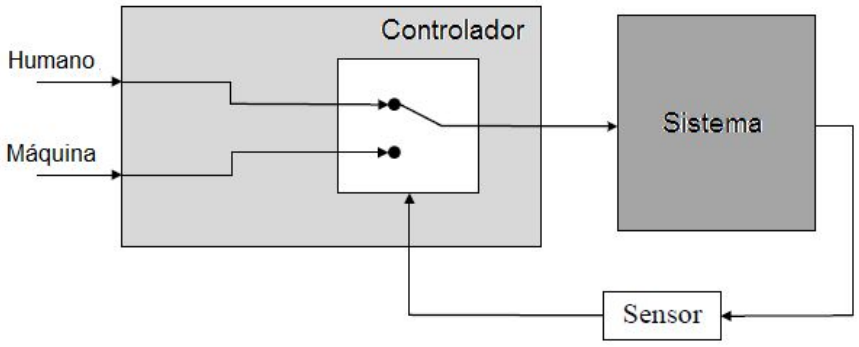
Autor: Adaptado de Perdigão (2014).

O controle Partilhado objetiva a união da competência de um sistema computacional em realizar tarefas repetitivas de maneira rápida precisa e incansável, com a capacidade criativa e adaptativa do ser humano (NASCIMENTO JUNIOR, 2016).

2.5.2 Controle negociado

O tipo de controle Negociado ou Série é uma abordagem em que o controle manual ou autônomo pode ser selecionado a qualquer momento para assumir a operação de uma máquina. O tipo Negociado consiste em um mecanismo que seleciona o controle entre homem e máquina, havendo uma troca entre os dois. Em outras palavras, o controlador decide se o controle se encontra completamente do lado humano ou do lado da máquina, conforme pode ser observado na Figura 10 (ONG et. al., 2005), (PERDIGÃO, 2014).

Figura 10 - Malha de Controle Negociado.



Autor: Adaptado de Perdigão (2014).

De acordo com Perdigao (2014), tanto o homem quanto o robô podem trocar o controle baseando-se na demanda da tarefa e nas restrições do ambiente, havendo duas principais condições em que o controle é negociado, ou seja, na execução de uma tarefa de navegação, onde o robô toma uma direção errada e o ser humano pode intervir e assumir o controle do robô para a correção da direção; e no caso do robô sobrepor os comandos do humano, quando estes comandos possam vir a causar algum tipo de dano à máquina.

2.5.3 Controle colaborativo

Segundo Perdigão (2014), o Controle Colaborativo ou Combinado é uma junção entre o controle Partilhado e o Negociado. Esta concepção busca abordar o robô como um parceiro, e não como uma ferramenta, aplicando um canal de comunicação entre homem e máquina.

Na configuração combinada, tanto o controle compartilhado quanto o negociado interagem entre si. O controle combinado pode trocar seu método de operação, considerando as necessidades da situação atual, buscando permitir um compartilhamento preciso na negociação do controle. Em situações onde um robô não é capaz de executar uma determinada tarefa com precisão, ele tem o possibilidade de passar o controle para o operador humano, permitindo que a tarefa seja dinamicamente alocada entre humano e robô durante as funções. (PERDIGÃO, 2014).

Segundo Perdigão (2014), a abordagem do controle Colaborativo permite que o ser humano e o robô trabalhem melhor em conjunto, tornando esta característica importante, onde o sucesso do desempenho da tarefa necessita de ambos. Como exemplo, o robô pode utilizar da capacidade do ser humano para a percepção do ambiente e manter as tarefas de comando e controle do lado da máquina.

3 TRABALHOS RELACIONADOS

A navegação autônoma e semi-autônoma de cadeira de rodas motorizada baseadas em robôs móveis tem sido tema de diversas pesquisas atuais. A exigência de detecção de obstáculos e passagens por aberturas estreitas, junto com diferentes formas de controle de trajetória de robôs, tem sido o foco em vários trabalhos encontrados na literatura, onde busca-se explorar diferentes abordagens e ferramentas para resolver os problemas da navegação.

Baseado neste contexto, são abordados neste capítulo os métodos utilizados na detecção e localização de portas, passagens estreitas e o controle da trajetória, buscando atingir o objetivo final com desempenho satisfatório.

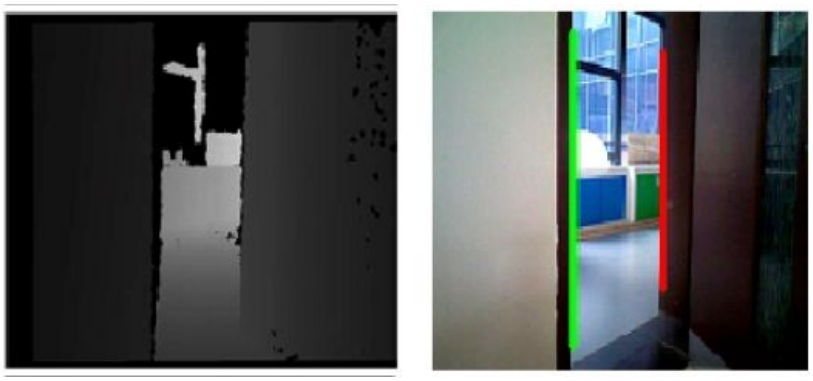
Em seu trabalho, Osman et. al. (2017) desenvolvem um algoritmo em Matlab para a detecção de entradas e obstáculos, utilizando imagens de profundidade obtidas por um sensor Kinect. O algoritmo calcula o deslocamento de profundidade de cada pixel, representando como pixel 0 os pontos com a maior profundidade e pixel 1 os pontos com menor profundidade. Uma entrada é detectada quando a área de pixels 0s calculados pelo algoritmo é alta e apresenta um comprimento suficiente para ser considerada uma porta ou corredor. Os autores estabelecem a posição do sensor fixa e perpendicular a entrada, o que torna o sistema não adequado para aplicações com robôs móveis.

Com o objetivo de detectar e localizar portas, Zhou et. al (2014) apresentam em seu trabalho o uso de uma técnica baseada em imagens de profundidade realizadas por um sensor Kinect V1. O sensor é mantido na horizontal e desta forma, as bordas das portas nas imagens de profundidades são vistas como linhas retas verticais, com geralmente grande diferença de profundidade entre um lado e outro. Por meio da análise dessa diferença de profundidade, uma borda vertical pode ser detectada.

Em seguida, Zhou et. al (2014) distinguem se a borda identificada pertence ao lado esquerdo ou direito da porta. Os autores também estabelecem premissas para verificar se a borda vertical detectada realmente pertence a uma porta ou se trata de ruído ou algum obstáculo.

A Figura 11 apresenta a imagem de profundidade em preto e branco obtida por Zhou et. al (2014), onde pode ser observado a identificação das bordas, sendo que a linha vertical verde representa a borda da esquerda e a linha em vermelho a borda da direita.

Figura 11 - Resultados de detecção de profundidade.



Autor: Adaptado de Zhou et. al (2014).

De posse destas características as bordas verticais são detectadas e localizadas com a origem das coordenadas no plano do sensor. Em seguida, o ângulo formado entre a parede onde as bordas foram identificadas e o plano do sensor é estimado por meio de funções trigonométricas.

Finalmente, a informação da porta é extraída com uma correlação entre a borda da esquerda e da direita da porta identificada. Cada borda da porta pode ser representada pela equação 7.

$$D = (x, X, Z, ty, \theta) \quad (7)$$

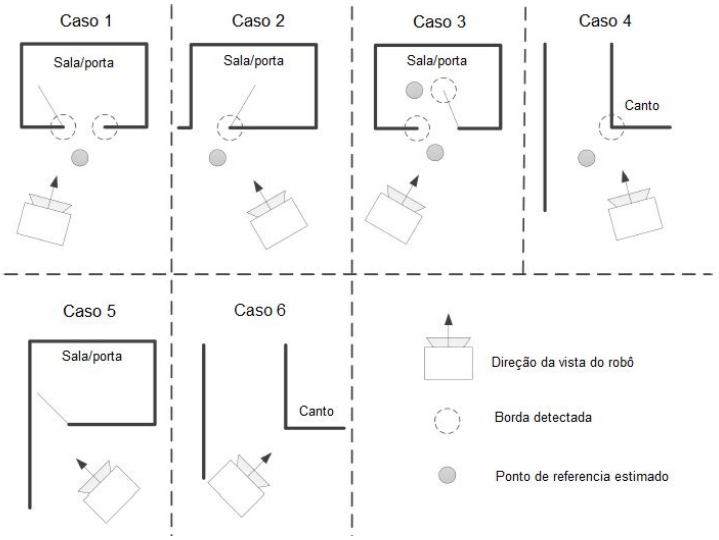
Onde, x corresponde às coordenadas da imagem no eixo x; X e Z são as coordenadas da borda da porta no espaço real; ty representa o tipo da borda, que podem ser esquerda ou direita; θ é o ângulo entre a parede e o plano do sensor.

Por fim, Zhou et. al (2014) determinam pontos de referência para a navegação. Neste trabalho, os autores não realizam o movimento do robô, apenas estipulam os pontos ideias que robô deve passar para cruzar a porta. A posição dos pontos dependem das dimensões do robô e da largura da porta. Os resultados

indicam que o algoritmo proposto pode detectar e gerar os pontos de referência em 95% dos casos, para as condições estabelecidas.

Os autores apontam que em determinadas circunstâncias as duas bordas da porta não podem ser detectadas simultaneamente devido a direção em que o robô se encontra. A Figura 12 apresenta os diferentes casos em que bordas podem ser detectadas, junto com os pontos de referências indicados para a navegação. Os casos 1 e 3 é possível detectar as duas bordas pertencentes a porta. Os casos 2 e 4 apenas uma borda pode ser detectada. Já nos casos 5 e 6 não é possível identificar nenhuma borda, devido a posição do robô.

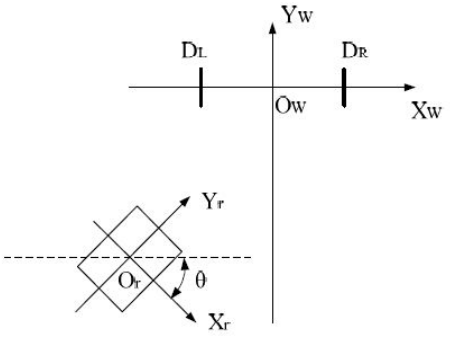
Figura 12 - Casos de detecção da borda.



Autor: Adaptado de Zhou et. al (2014).

Dai et. al. (2013), em seu trabalho, objetivam detectar, localizar e realizar o cruzamento de portas. Para detectar e localizar as bordas das portas os autores utilizam as mesmas técnicas apresentadas por Zhou et. al (2014). As portas são detectadas e localizadas com o sistema de coordenadas na origem do Kinect, que os autores chamam de *Robot Coordinate System* (RCS). Em seguida, através da transformação de coordenadas, o robô é localizado no *World Coordinate System* (WCS), com o sistema de coordenadas tendo o centro da porta na origem, conforme pode ser visto na Figura 13.

Figura 13 - Robô no WCS



Autor: Dai et. al. (2013).

Para o robô alcançar uma posição apropriada e cruzar a porta, Dai et. al. (2013) desenvolvem um controlador adaptativo não-linear, que se baseia nos modelos cinemáticos do robô e nas informações de posição extraídas pelo Kinect.

O algoritmo classifica a posição do robô em “boa posição” ou “má posição”, onde uma má posição pode ser estabelecida quanto o algoritmo não consegue extrair corretamente as duas bordas da porta, ou pela posição da porta em relação ao robô não ser favorável ou pelas limitações de visão do sensor Kinect. Quando a posição se enquadra em “má”, através de um controlador PD, o robô gira até ficar praticamente perpendicular à porta, para assim conseguir estabelecer a posição como “boa”.

Em seguida, aplica-se o controlador não-linear desenvolvido para cruzar a porta. Quando o robô chega perto da porta, há uma perda de visão do sensor e o sistema faz o uso de um algoritmo para desviar obstáculos e assim concluir a passagem da porta. Dai et. al. (2013) apontam que o algoritmo realizou com sucesso a passagem por portas em 85% das vezes.

Maciel (2018), em parte do seu trabalho, propõe um sistema autônomo de controle de movimento para uma cadeira de rodas motorizada, sendo que ao reconhecer alguma passagem estreita, o algoritmo realiza o controle necessário para atravessar esta passagem sem empregar técnicas de localização global e de mapeamento. Para tal tarefa, uma metodologia baseada em varredura bidimensional definida no plano cartesiano do robô, utilizando uma câmera RGB-D Asus XtionPRO, resultando assim um conjunto de pontos com as distâncias e ângulos em relação ao

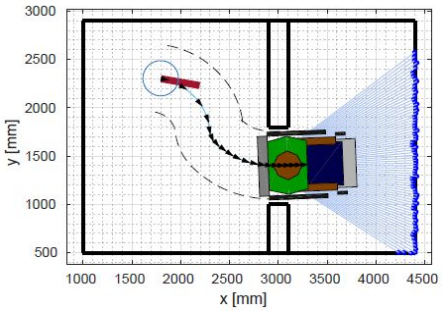
sensor. Vale observar que o algoritmo foi desenvolvido utilizando a linguagem de programação *python* e, posteriormente, embarcado em uma plataforma de desenvolvimento Raspberry Pi 3.

Uma passagem é estimada por dois pontos, localizados nas bordas, esquerda e direita de uma porta ou passagem. As distâncias e angulação das bordas são extraídas em relação ao referencial do robô e transformadas para o sistema de coordenadas com o referencial na passagem, cuja a origem é o ponto médio das duas extremidades.

Em seguida, Maciel (2018) aplica o Filtro de Kalman Estendido (EKF) para obter uma localização mais estável, atenuando os ruídos e incertezas da leitura do sensor, e com isso, é possível estimar localmente a posição do robô, com auxílio dos sensores de odometria.

Por fim, o autor apresenta um controlador de navegação não-linear para robôs diferenciais que age diretamente nas velocidade linear e angular do robô, ambas controladas de forma independente. Para isso, é estabelecida uma reta central a passagem que o robô deve percorrer para atravessá-la. O controlador é dividido em duas etapas, sendo que a primeira etapa é responsável pelo controle que faz com que o robô se aproxime de um ponto próximo a passagem, de tal forma que sua direção fique coincidente a reta da passagem; já a segunda etapa realiza o controle de um seguidor de reta, mantendo o robô próximo a reta central até cruzar a passagem. A figura 14 apresenta o resultado da simulação da trajetória percorrida pelo robô.

Figura 14 - Trajetória cruzamento da passagem



Autor: Adaptado Maciel (2018).

Com o objetivo de realizar a navegação autônoma de uma cadeira de rodas elétrica comercial em ambientes internos de maneira econômica e robusta, Burhanpurkar et. al. (2017) utilizam um sensor Kinect V2 para fornecer dados espaciais em 3D, *encoder* acoplados às rodas para fornecer dados para a odometria e um computador de bordo. O sistema utiliza um *software* de localização e mapeamento simultâneo (SLAM) que construiu o mapa do ambiente, fazendo uso significativo do sistema operacional para robôs (ROS), para fornecer suporte para tarefas de navegação.

Os autores tratam a passagem de portas de forma distinta do restante da navegação, podendo operar independente de um mapa completo do ambiente. As portas ou passagens estreitas são identificadas por uma nuvem de pontos 3D, que se baseia em sua natureza plana e espaço vazio. Quando uma porta a ser atravessada é identificada, o algoritmo entra em uma sub-rotina específica para o cruzamento da porta, de maneira a seguir um caminho suave e seguro até o ponto de destino. Uma taxa de sucesso de 100% foi obtida no cruzamento das portas de um total de 200 testes realizados.

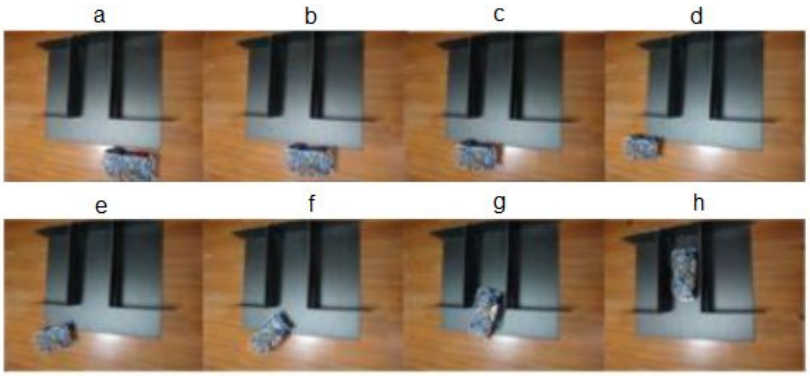
No trabalho apresentado por Wang et. al. (2016), os autores objetivam desenvolver um sistema de estacionamento automático para veículos, baseado em um controlador *fuzzy*. Através do modelo cinemático é possível determinar a próxima posição e direção do veículo, através da sua velocidade e angulação da instante anterior.

O controlador *fuzzy* proposto possui três entradas e duas saídas, sendo as entradas formadas pelas coordenadas (*x*, *y*) e o ângulo do corpo do veículo em relação a vaga, e as saídas sendo o ângulo de direção das rodas e a velocidade do veículo. O projeto das regras de restrição são estabelecidas com base na experiência de manobras do veículo. Por fim, o controlador é validado através de simulações em Matlab e, também, em um robô representando o veículo em uma escala 1:10.

A Figura 15 indica o movimento do robô para realizar o estacionamento. A primeira linha da imagem, da esquerda para a direita, mostra o processo em que o robô encontra a vaga de estacionamento. No início, o robô anda mantendo uma distância segura da barreira (a) e (b). Quando descobre a vaga (c), continua

andando até chegar ao local que o carro deve parar e se preparar para efetuar o estacionamento (d). Em seguida, a segunda linha da imagem, da esquerda para a direita, mostra todo o processo de estacionamento realizado pelo controlador *fuzzy*. O robô entra no espaço de estacionamento mantendo uma distância segura de ambos os lados, (e),(f) e (g), até finalizar o movimento (h).

Figura 15 - Teste de validação em escala



Autor: Adaptado Wang (2016).

Já em Omrane, S. Masmoudi e Masmoudi (2016) é desenvolvido um controlador *fuzzy* para a navegação de um robô diferencial. Inicialmente no trabalho é obtido o modelo cinemático do robô, e a partir daí pode ser determinada qual será a próxima posição do robô no espaço, com base nas velocidades das rodas e o ângulo de orientação atual. O controlador *fuzzy* desenvolvido utiliza duas entradas: a distância do robô até o ponto de navegação e o ângulo de orientação do corpo. As saídas do controlador são as velocidades das rodas esquerda e direita do robô.

Magalhães (2018), em parte do seu trabalho de controle de trajetórias de robôs, propõe o uso de um controlador *go to position*. Considerando a trajetória como um conjunto de pontos, o controlador consiste em uma abordagem de deslocar o robô de um ponto a outro sem se preocupar com o que acontece nos pontos intermediários, garantido apenas que o robô chegue ao ponto desejado *x, y* com ângulo desejado. Para calcular o sinal de controle, o autor optou por utilizar um controlador do tipo PD, modelando o sistema para diferentes velocidades, extraindo os componentes de velocidade linear e angular para atingir o ponto desejado.

4 DESENVOLVIMENTO DO TRABALHO

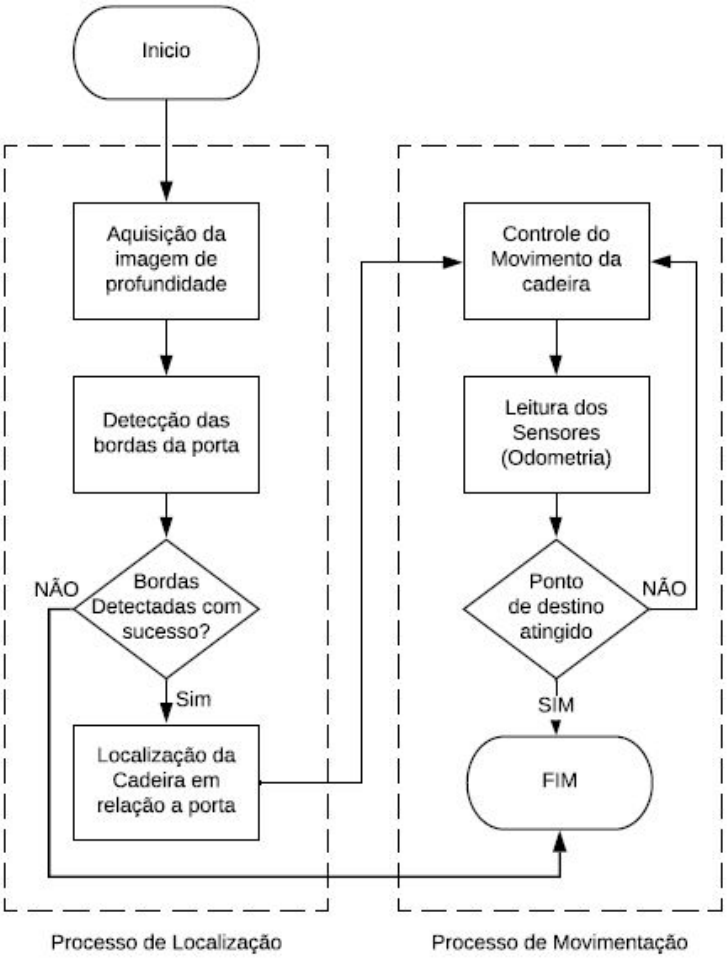
Este trabalho busca uma solução quanto à mobilidade de usuários de cadeiras de rodas, e para tal, objetiva realizar a prova de conceito de um sistema capaz de realizar a passagem segura de uma cadeira de rodas elétrica através de portas em um ambiente interno. A metodologia adotada neste trabalho consiste na aquisição de imagens de profundidade, a fim de localizar alguma porta nas proximidades, e, na sequência, a definição da trajetória e do controle autônomo do movimento. A realização destes passos permite que a cadeira possa transpassar a porta detectada, evitando colisões no deslocamento do usuário.

O sistema de aquisição de imagens desenvolvido extrai apenas as informações de profundidade de uma câmera RGB-D, a fim de obter as coordenadas das bordas verticais de uma porta. No sistema idealizado a navegação da cadeira no ambiente se faz de forma manual pelo usuário, utilizando um *joystick* da própria cadeira, e quando o usuário desejar passar por uma porta, ele deverá trocar o controle da cadeira para automático através de uma chave e indicando ao sistema para executar as funções da navegação.

A imagem de profundidade é processada apenas para extrair a posição inicial da cadeira em relação a porta, sendo que sua localização durante o movimento se dá somente por meio da odometria, que é obtida por meio dos *encoders* instalados junto às rodas da cadeira.

O desenvolvimento deste trabalho é dividido em dois processos distintos, sendo o primeiro destinado a obter a localização da cadeira de rodas em relação a um ponto de passagem e o outro para realizar o movimento da cadeira de rodas até este ponto. As principais etapas de cada um dos processos podem ser observadas no fluxograma apresentado na Figura 16.

Figura 16 - Fluxograma do sistema desenvolvido



Autor: O autor (2019).

Basicamente, o processo de localização é subdividido em etapas. Inicialmente, há a obtenção da imagem de profundidade, após executa-se um algoritmo de varredura que identifica as bordas verticais da porta, e, por fim, estima-se a localização e direção da cadeira em relação a uma origem no centro da porta.

Dada a conclusão do processo de localização inicial da cadeira em relação ao ponto de passagem, inicia-se o processo de movimentação, por meio de um controlador *fuzzy*, que se baseia no modelo cinemático da cadeira e na aquisição da posição atual da cadeira pela odometria.

Nas seções seguintes, é apresentado o hardware utilizado e as metodologias para a localização inicial da cadeira e o processo de movimentação da mesma.

4.1 HARDWARE

Para realizar a identificação das bordas da porta foi utilizado uma câmera RGB-D Microsoft Kinect v2, instalada na horizontal e sobre a cabeceira da cadeira de rodas, a aproximadamente 1,5 metros de altura do chão, a fim de ter uma visão clara do ambiente.

O Kinect v2 é capaz de fornecer tanto imagens de cor quanto imagens de profundidade, por meio do uso da tecnologia ToF, com uma resolução de 512x424 *pixels*, e, além disso, utiliza uma interface de comunicação USB 3.0. Um ponto a ser observado neste sensor é necessidade de uma alimentação de 12V, atípico em sistemas embarcados.

Inicialmente, pretendeu-se embarcar o sistema em uma Raspberry Pi 3 model B. Entretanto, devido ao Kinect v2 e suas bibliotecas necessitarem de uma interface USB 3.0 e a Raspberry Pi 3 contar apenas com USBs 2.0, foi utilizado um Notebook Avell A62 MUV com um sistema operacional Linux Ubuntu 19.04 64-bit. O notebook dispõe de um processador Intel Core i7 com frequência de 2.6GHz, 16GB de memória RAM, além de uma unidade de processamento gráfico (GPU) NVIDIA GeForce GTX 6GB e duas portas USB 3.0 (AVELL, 2019).

Com isso, para realizar os comandos de velocidade para o controle da cadeira e também para efetuar a leitura dos pulsos dos *encoders* foi utilizado uma placa de desenvolvimento ESP-32 DevKit v1. A placa possui dois Microprocessadores Xtensa 32-bit com velocidade de 600 DMIPS, uma frequência de clock de 160Mhz, 34 portas programáveis GPIOs e dois conversores digitais analogico (DAC) de 8-bits. A escolha por esta plataforma se deu principalmente por conter os conversores digitais analógicos, que podem fornecer uma tensão de saída de 0 a 3,3V, necessários para o controle do sistema de potência e acionamento dos motores da cadeira de rodas motorizada. Outro motivo foi por esta plataforma já ter sido utilizada em outros trabalhos similares, sendo constatada sua funcionalidade na aplicação, como em Ransan (2019).

Os *encoders* foram instalados diretamente no eixo dos motores de cada roda, onde cada *encoder* tem uma resolução de 360 pulsos por volta, e conta com dois canais de saída A e B defasados em 90 graus, podendo ser alimentado de 5V a 24V.

Todo o sistema foi instalado em uma cadeira de rodas motorizada fabricada pela empresa Freedom, do modelo Compact 13, que opera com uma bateria de 12V com capacidade de 26Ah. O sistema de controle da cadeira recebe as informações para o movimento por meio de um *joystick* acoplado no apoio de braço da cadeira (FREEDOM, 2019).

A Figura 17 ilustra como os componentes são integrados uns com os outros, sendo que as flechas azuis indicam o caminho da comunicação de informação entre os componentes.

Figura 17 - Integração dos componentes

Autor: O autor (2019).

Vale frisar que o Kinect obtém as imagens de profundidade e comunica com o notebook, que fica responsável por executar o algoritmo de varredura e localizar a cadeira em relação a porta, passando as informações para o ESP-32 executar o

controle do movimento e efetuar a leitura da posição atual da cadeira durante o movimento.

4.1 LOCALIZAÇÃO INICIAL DA CADEIRA

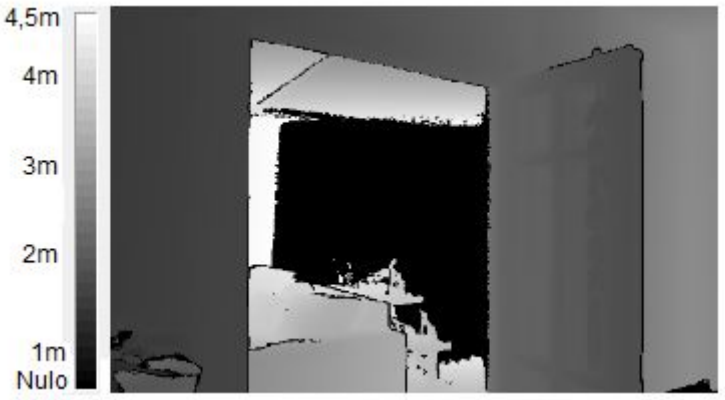
O processo de localização inicial da cadeira busca responder as perguntas “Onde estou?” e para “Onde irei?”, conceitos fundamentais para a navegação de robôs móveis, conforme abordado no item 2.1. Neste processo inicialmente o algoritmo realiza a aquisição da imagem de profundidade e em seguida executa uma varredura na imagem a fim de detectar e localizar portas, definindo um ponto de passagem para a navegação.

4.1.1 Aquisição da imagem de profundidade

A aquisição da imagem de profundidade foi realizada utilizando a biblioteca de código aberto *libfreenect2* da OpenKinect, compatível com o Kinect v2. A biblioteca foi instalada no sistema operacional Linux, seguindo os passos especificados no GitHub na página da OpenKinect (OPENKINECT, 2019). O algoritmo foi desenvolvido a partir do código exemplo, denominado na biblioteca como *Protonect*, escrito em C++, que tem como propósito descrever as funções de inicialização, configurações iniciais do Kinect e também as funções para adquirir a imagem de profundidade de forma não distorcida.

O código exemplo gera uma imagem em escala de cinza, em que cada pixel tem uma tonalidade proporcional à distância do ponto focal do sensor, como pode ser observado na Figura 18. Os objetos mais próximos apresentam um tom mais escuro e objetos mais distantes em tons mais claros. Sendo assim, quando o *range* do sensor é ultrapassado a imagem fica preta (valor de medida nulo), como pode ser observado no centro da Figura 18.

Figura 18 - Imagem de Profundidade



Autor: O autor (2019).

Para extrair as informações de profundidade da cada pixel da imagem, a biblioteca dispõe de uma função com a seguinte sintaxe *getPoitXYZ*(&undistored, pixel linha, pixel coluna, x, y, z). Esta função retorna os valores da distância em metros nos eixos cartesianos X, Y e Z do ponto da imagem desejado, com uma resolução de 0,001m (OPENKINECT, 2019).

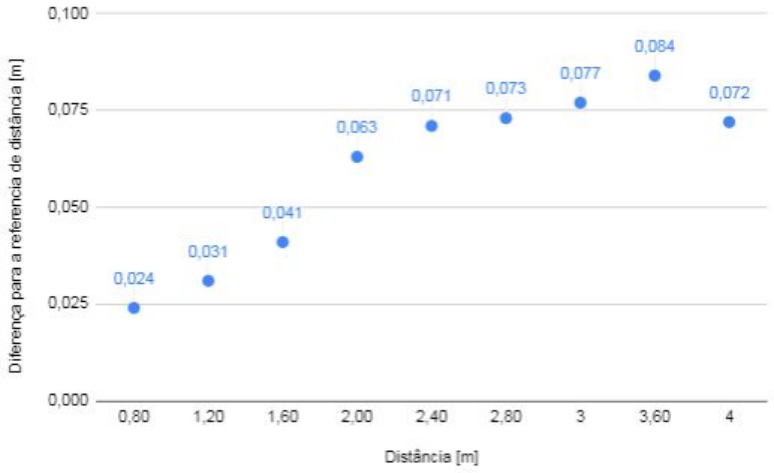
Um procedimento para determinar o erro associado às medidas foi realizado seguindo as especificações apresentadas no trabalho de Pagliari e Pinto (2015). Logo, o sensor foi posicionado a distâncias conhecidas de um objeto para comparar com a distância medida pelo Kinect. O objeto foi afastado do sensor de 0,8 a 4m, com passos regulares de 0,4m, e para cada posição foram adquiridas 100 imagens de profundidade e utilizado sua média da distância no centro da imagem para determinar o erro.

Como forma de referência, as distâncias foram medidas utilizando uma trena laser, medindo nas duas extremidades do sensor para evitar algum efeito de rotação. Foi utilizada a trena laser marca BOSCH, modelo GLM 40 Professional, que apresenta uma resolução de 1mm e precisão de medida de ±1,5mm (BOSCH, 2015).

A Figura 19 mostra a diferença da distância entre a obtida pela trena laser e pelo sensor Kinect. Essa diferença apresenta um aumento em função do acréscimo da distância em relação ao objeto alvo, variando de 0,02m a 0,08m. O erro obtido

atinge no máximo 2,9%, não tendo uma influência significativa na estimativa da localização da porta.

Figura 19 - Erro medidas Kinect v2



Autor: O autor (2019).

4.1.2 Detecção e localização da porta

A metodologia para a detecção e localização das portas segue os conceitos abordados por Dai et. al. (2013) e Maciel (2018). Neste trabalho, uma porta é estimada por dois pontos, sendo a borda direita e esquerda da mesma. Com o Kinect alinhado com o plano do chão, as bordas pertencentes a uma porta podem ser vistas como linhas retas verticais, com uma diferença de profundidade entre a parede e o vão da porta. Baseado nesse conceito, o algoritmo busca identificar quais pontos da imagem em que essa diferença ocorre são pertencentes a uma porta.

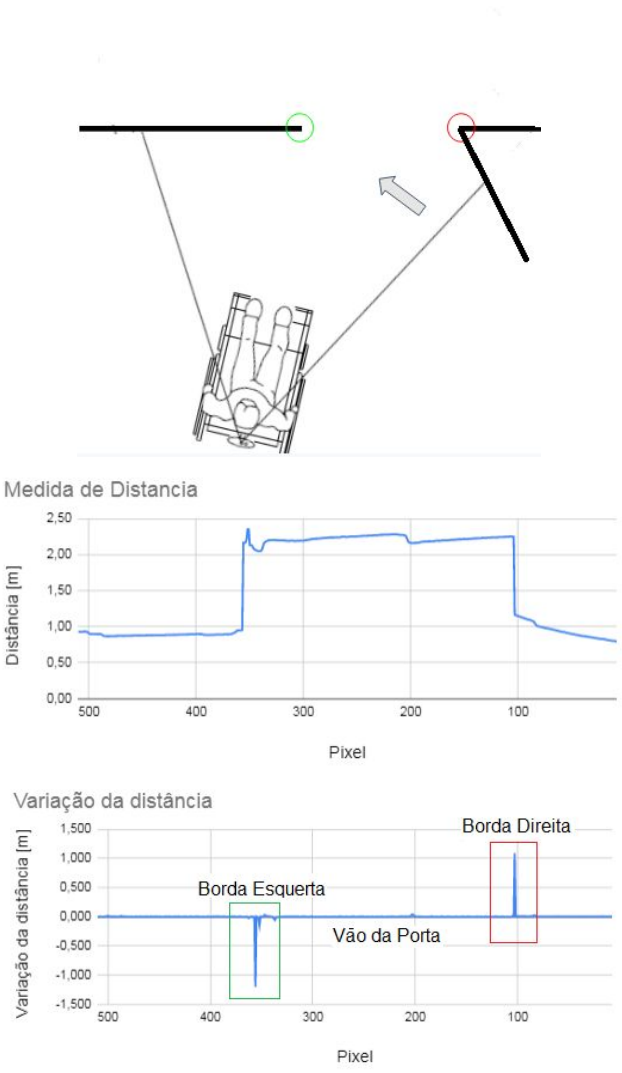
O método para extrair possíveis bordas da porta consiste em realizar uma varredura unidirecional, da direita para a esquerda, aplicada ao centro da imagem, extraindo a profundidade de cada pixel e comparando com o seu subsequente.

Dai et. al. (2013) estabelece o limiar de diferença de profundidade entre a parede e o vão da porta em 500mm, logo quando algoritmo encontra esta diferença determina-se um candidato a borda de porta. Visto que não foi aplicada uma etapa de pré-processamento na imagem de profundidade, para evitar que essa diferença

seja proveniente de algum ruído, o algoritmo também compara as profundidades com um range 10 pixels, com intuito de verificar se essa diferença ainda existe nestas condições.

Quando a taxa de variação da distância entre os pixels é positiva, a borda é uma candidata ao lado direito, e quando a variação é negativa, a borda é uma candidata ao lado esquerdo da porta. A Figura 20 contém um exemplo de uma região com uma passagem, onde podem ser identificadas os dois lados da porta.

Figura 20 - Medidas de uma varredura na imagem



Autor: Adaptado Maciel (2018).

Cada possível borda *B* detectada é representada por:

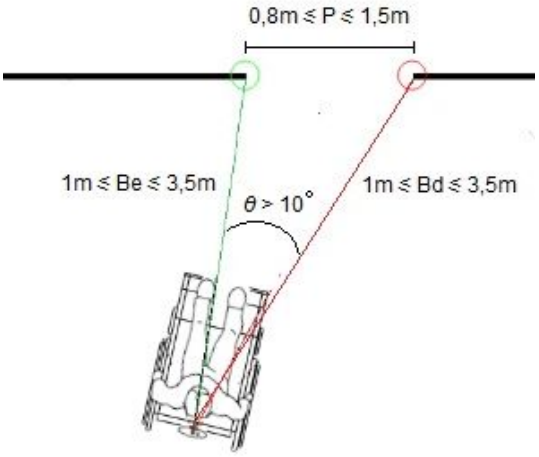
$$B = (Z, X, \theta, Lp, D) \quad (8)$$

Onde, *X* e *Z* são as coordenadas em metros da borda a partir do plano do sensor, θ é o ângulo de orientação em que a borda se encontra, *Lp* representa qual lado pertence a borda, podendo ser 1 quando é ao lado direito e 2 quando é ao lado esquerdo e *D* é a medida da borda até o ponto focal do sensor. Todos os pontos de descontinuidade, que são observados pela varredura na imagem, são armazenados como possíveis bordas de uma porta, cabe ao algoritmo selecionar quais realmente são pertencentes a uma abertura.

A próxima etapa é avaliar os pontos de descontinuidade que foram extraídos. O objetivo desta etapa é selecionar os pares de bordas que satisfazem as condições para serem considerados uma porta, e rejeitar aqueles que foram estimados de forma incorreta pela etapa anterior, devido algum objeto no ambiente ou ruído na imagem. Sendo assim, são avaliados e selecionados os pares que atendem às seguintes condições:

a. Largura da passagem: selecionam-se pares de bordas que apresentam largura mínima de 0,8m, estabelecida pela NBR 9050 (2015), e uma largura máxima de 1,5m. Rejeitam-se portas com pouca largura e portas tão largas que uma navegação autônoma seja desnecessária.
b. Distância da cadeira: para ser identificada como uma porta suas bordas devem estar a distâncias entre 1m e 3,5m do ponto focal do sensor, estas medidas são definidas pelas limitações do Kinect, que apresenta um *range* que varia de 0,5 a 4,5m. passagens mais distantes são mais susceptíveis a uma estimação equivocada.
c. Angulação da passagem: rejeitam-se passagens que não apresentam uma diferença angular superior a 10 graus entre as bordas, quando essa diferença é pequena isso significa que a cadeira está muito próxima a parede em que está a porta, podendo causar uma estimativa equivocada. A Figura 21 apresenta um exemplo de situação onde uma porta pode ser reconhecida.

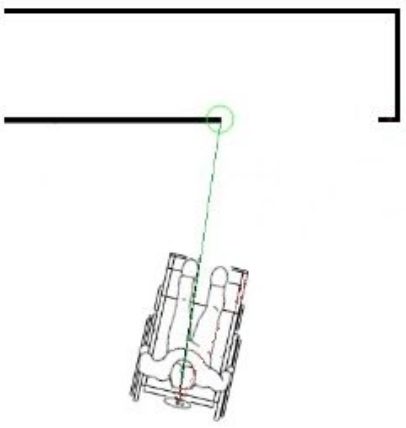
Figura 21 - Exemplo de reconhecimento da porta



Autor: O autor (2019).

Inicialmente a proposta era desenvolver um algoritmo capaz de identificar apenas portas onde dois pontos de descontinuidade podem ser observados do ponto de vista do sensor Kinect. Entretanto também foi desenvolvida uma metodologia, baseada em Maciel (2018), que objetiva identificar portas onde apenas um ponto de descontinuidade pode ser observado pelo sensor, como no caso ilustrado na Figura 22.

Figura 22 - Porta com um ponto de descontinuidade



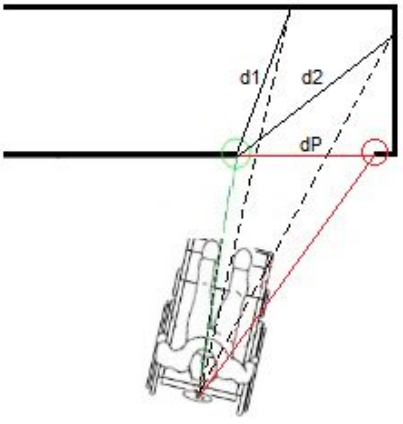
Autor: O autor (2019).

Quando o algoritmo não encontra dois pontos de descontinuidade que satisfazem as condições estabelecidas, uma sub rotina busca encontrar o outro extremo da porta. Uma porta é constituída de pelo menos um ponto de descontinuidade.

Ao identificar um ponto de descontinuidade em $pixel_n$ e se esta descontinuidade for pertencente a borda direita, o algoritmo calcula a distância deste ponto a todos os seguintes, o ponto que apresentar a menor distância é considerado o extremo esquerdo da porta. Da mesma forma se o ponto de descontinuidade pertencer a borda esquerda da porta o algoritmo calcula a distância deste ponto a todos os seus antecessores, encontrando o extremo direito da porta.

A Figura 23 ilustra o processo quando um ponto de descontinuidade observado pertence a borda da esquerda.

Figura 23 - Reconhecimento da porta com um ponto de descontinuidade



Autor: Adaptado Maciel (2018).

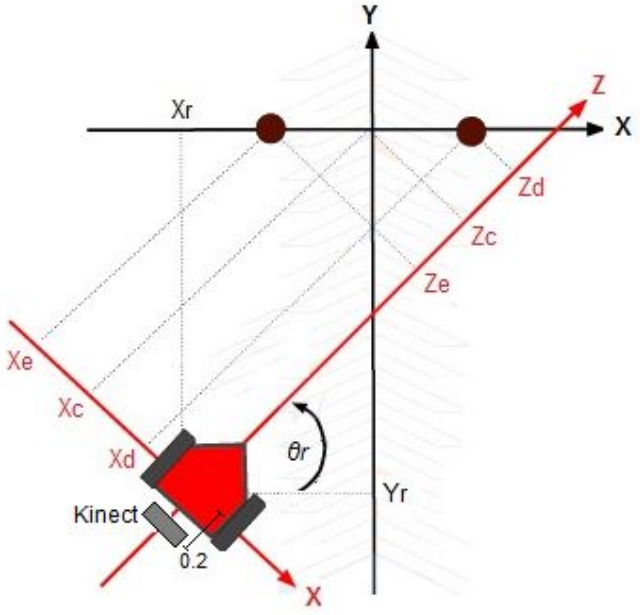
Da mesma forma que o método por dois pontos de descontinuidade, as condições de largura, distância e angulação devem ser atendidos para a porta ser reconhecida.

Após o reconhecimento da porta, a passagem é estimada pelos seus dois pontos extremos esquerdo e direito. Para cada um dos pontos temos as distâncias e ângulos em relação ao referencial da cadeira X_e, Z_e, θ_e, para o lado esquerdo e X_d, Z_d, θ_d, para o lado direito. Em seguida é determinado o ponto médio entre os

extremos, que será o ponto de passagem para a navegação, logo, este ponto também é representado por sua distância e ângulo em relação ao referencial da cadeira.

Por fim, o sistema de coordenadas passa a ter o seu referencial no ponto de passagem e a cadeira passa a ser localizada através das coordenadas X_r , Y_r e orientação θ_r em relação a este ponto. A Figura 24 mostra os sistemas de coordenadas e as variáveis envolvidas na localização.

Figura 24 - Sistemas de coordenadas da porta



Autor: Adaptado de Maciel (2018).

De acordo com Maciel (2018) as cadeira de rodas pode ser considerada um robô diferencial, que tem a posição referenciada no centro das rodas tracionadas. O ponto focal do sensor Kinect ficou deslocado das rodas da cadeira há uma distância de 0,2m, como pode ser observado na Figura 22, sendo necessário uma correção. As coordenadas do ponto médio da porta em relação a cadeira são determinadas através das Equações 9 e 10.

$$X_c = \frac{X_d + X_e}{2} \tag{9}$$

$$Z_c = \frac{Z_d + Z_e}{2} - 0,2 \tag{10}$$

Para localizar a cadeira com o referencial ao ponto de passagem são utilizadas as Equações 11, 12 e 13.

$$\theta_r = atan2(Z_e - Z_d,\ X_e - X_d) + /2 \tag{11}$$

$$X_r = X_c sin(\theta_r) - Z_c cos(\theta_r) \tag{12}$$

$$Y_r = -X_c cos(\theta_r) + Z_c sin(\theta_r) \tag{13}$$

4.2 MOVIMENTAÇÃO DA CADEIRA

O processo de movimentação da cadeira busca responder a pergunta “Como chegar lá?”. Sabendo a posição inicial da cadeira e o ponto de destino, foi projetado um controlador Fuzzy atuando na velocidade angular da cadeira para realizar o seu movimento até a mesma transpassar a porta.

Inicialmente, a proposta foi implementar um controle em dois estágios, sendo o primeiro para executar o movimento até um ponto de segurança e outro para andar em linha reta até passar a porta. Entretanto, o sistema de controle desenvolvido executa o movimento em apenas uma etapa, partido da posição inicial até concluir o movimento, ou seja, quando sua posição atingir o valor 0 no eixo Y.

O algoritmo de controle é baseado em uma aplicação de estacionamento autônomo de veículos apresentado por Lopes, Oliveira e Pinheiro (2014), em um código exemplo. O algoritmo foi modificado para ser aplicado em robôs diferenciais, onde foram definidas as funções de pertinência e regras para executar o movimento necessário para a aplicação.

Para o projeto do controlador é necessário conhecer o modelo cinemático da cadeira de rodas. A cadeira possui um sistema de acionamento diferencial de duas rodas movidas por dois motores de corrente contínua. Através das velocidades da roda esquerda (V_e) e da roda direita (V_d), pode-se determinar sua velocidade linear e

angular, e com isso estimar sua posição e orientação no ambiente (OMRANE et. al, 2016).

Como apresentado no item 4.1.2, a cadeira é representada com coordenadas X_r, Y_r (mm) e orientação θ_r (rad) em relação ao ponto de passagem no centro da porta. Com base nestas informações é possível obter a posição na cadeira ao longo do caminho a partir do conjunto de equações abaixo, onde *L* (mm) representa a distância entre eixos de cada roda, *v* (mm/s) a velocidade linear e *w* (rad/s) a velocidade angular.

$$v = \frac{V_e + V_d}{2} \tag{14}$$

$$\frac{dx}{dt} = \frac{V_e + V_d}{2} cos(\theta_r) \tag{15}$$

$$\frac{dy}{dt} = \frac{V_e + V_d}{2} sen(\theta_r) \tag{16}$$

$$\frac{d\theta}{dt} = \frac{V_e - V_d}{L} = w \tag{17}$$

4.2.1 Projeto do controlador

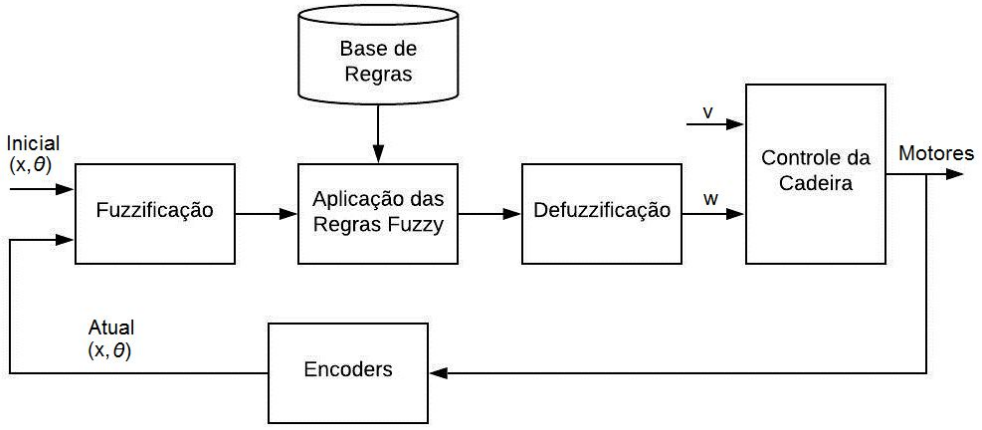
O controlador Fuzzy desenvolvido gerência a tarefa de navegação, controlando a posição e angulação da cadeira de rodas ao longo da trajetória, a fim de atingir o ponto de passagem na posição (0,0) com a orientação de 90 graus ao eixo Y. A estrutura básica do controlador Fuzzy é composto de três blocos: a fuzzificação, regras de inferência, e defuzzyficação.

No desenvolvimento deste trabalho, foi utilizado o controlador Fuzzy de Mamdani, que é construído com base em um conjunto de regras ou implicações *if-then* do tipo {**se** premissa$_1$ **e** premissa$_2$… **e** premissa$_n$ **então** consequência$_i$}. As premissas representam as condições do conjunto Fuzzy e a consequência representa a ação de controle declarada no conjunto Fuzzy (CAVALCANTI et al., 2012).

A interface de fuzzificação utiliza as funções de pertinência para converter os valores numéricos em valores linguísticos. Estes são processados junto com a base de regras Fuzzy, de maneira a inferir as ações de controle Fuzzy a serem tomadas. Por fim, é realizado o estágio da defuzzyficação, que transforma as ações de controle linguísticas em ações de controle numéricas (BECKER, 2000).

No projeto, o controlador deve atuar modificando a velocidade angular da cadeira de rodas, com uma proporção baseada na sua posição e orientação atual. A Figura 25 ilustra o diagrama de blocos do controlador Fuzzy desenvolvido.

Figura 25 - Diagrama de blocos Fuzzy



Autor: O autor (2019).

O primeiro passo no projeto do controlador foi a definição das entradas e saídas linguísticas do sistema. As entradas definidas se referem a posição da cadeira no eixo x e o ângulo de orientação da mesma. O controlador foi projetado para realizar o movimento com uma distância mínima no eixo y de 1,5m, que a é espaço recomendado pela NBR 9050 (2015), permitindo assim a realização de manobras de 90 graus com deslocamento.

Já a saída do controlador Fuzzy é a velocidade angular em rad/s para realizar o giro a cadeira. O controlador atua apenas na velocidade angular, mantendo a velocidade linear fixa. Devido ao fato da cadeira se encontrar a curtas distâncias da porta, foi estabelecido uma velocidade linear de 300mm/s, o que proporciona um

movimento confortável ao usuário. Na Tabela 2 podem ser observadas as variáveis de entrada e saída definidas para controle, bem como a sua faixa de atuação.

Tabela 2 - Variáveis linguísticas

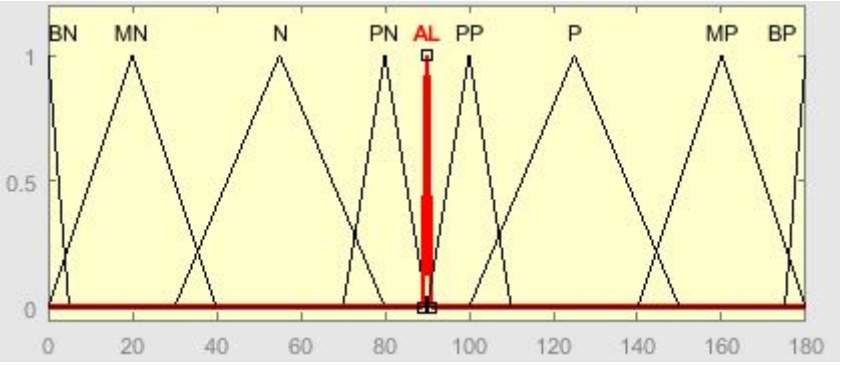
	Variáveis	Intervalo
Entradas	Ângulo de Orientação	[0 a 180] graus
	Posição Eixo X	[-1,5 a 1,5] m
Saída	Velocidade Angular	[-0,8 a 0,8] rad/s

Autor: O autor (2019).

As funções de pertinência e as regras de inferência foram criadas com o auxílio da ferramenta *Fuzzy Logic Designer* do *software* MatLab.

A variável linguística do ângulo de orientação é definida de 0 a 180, tendo como 90 graus o alinhamento com a porta, e o domínio de discurso é repartido em nove funções de pertinência triangulares. Foram definidos os grupos de valores das funções de pertinência, com base no conhecimento heurístico do comportamento do sistema. A variável de entrada pode pertencer aos grupos: Bastante Negativo (BN), Muito Negativo (ML), Negativo (N), Pouco Negativo (PN), Alinhado (AL), Pouco Positivo (PP), Positivo (P), Muito Positivo (MP) e Bastante Positivo (BP). A Figura 26 mostra as funções de pertinência para o ângulo de orientação.

Figura 26 - Funções de pertinência para ângulo de orientação

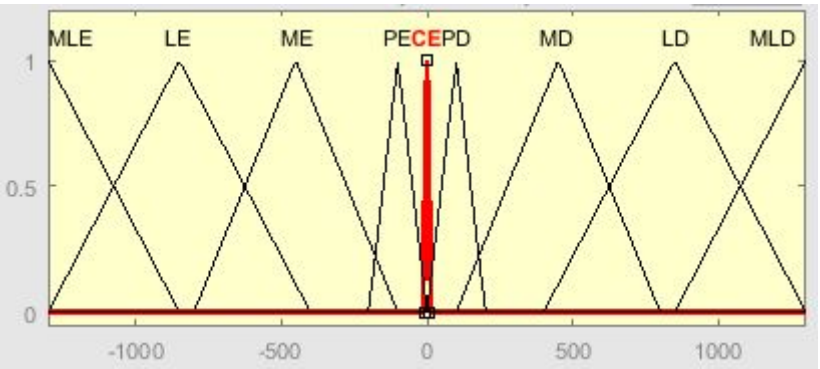


Autor: O autor (2019).

Com base no conhecimento heurístico foram definidos os grupos para a variável de posição, o seu domínio de discurso, de -1,5 a 1,5m, é repartido em nove funções de pertinências triangulares. A variável de entrada pode pertencer aos

grupos: Muito Longe a Esquerda (MLE), Longe a Esquerda (LE), Médio a Esquerda (ME), Perto a Esquerda (PE), Centralizada (CE), Perto a Direita (PD), Médio a Direita (MD), Longe a Direita (LD) e Muito Longe a Direita (MLD). A Figura 27 mostra as funções de pertinência para a posição da cadeira.

Figura 27 - Funções de pertinência posição x



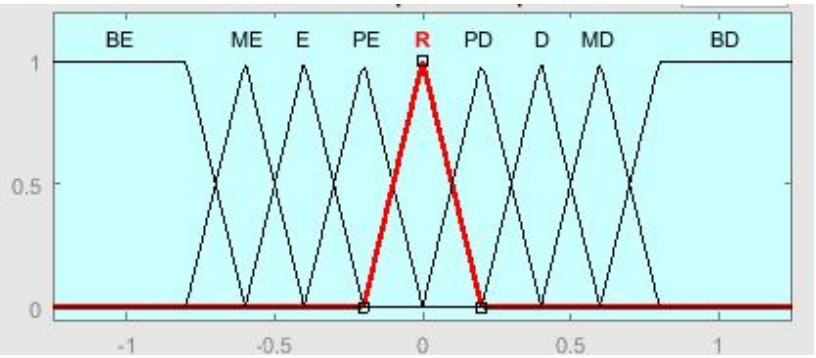
Autor: O autor (2019).

A variável de saída da velocidade angular tem seus valores gerados pelo controlador que depende das condições das premissas de entrada, logo, este valor é enviado para o controle da cadeira para a execução da rotação desejada.

As funções de pertinência da saída foram divididas em nove grupos, com a finalidade manter uma transição suave entre os movimentos. A variável de saída pode pertencer aos grupos: Bastante para a Esquerda, (BE), Muito para a Esquerda (MB), Esquerda (E), Pouco para a Esquerda (PE), Reto (R), Pouco para a Direita (PD), Muito para a Direita (MD) e Bastante para a Direita (BD).

A Figura 28 mostra as funções de pertinência para a saída de velocidade angular.

Figura 28 - Funções de pertinência saída velocidade



Autor: O autor (2019).

Após estabelecer as funções de pertinência, foi definido as bases para as regras de inferência. As regras foram estabelecidas com base no conhecimento heurístico do comportamento do sistema, com um total de 81 regras do tipo *if-then*, de acordo com o modelo: Se Ângulo Pouco Negativo e Posição Longe a Esquerda então Velocidade Muito para a Direita. As regras de inferência para a velocidade angular da cadeira de rodas podem ser observadas na Tabela 3.

Tabela 3 - Regras de Inferência

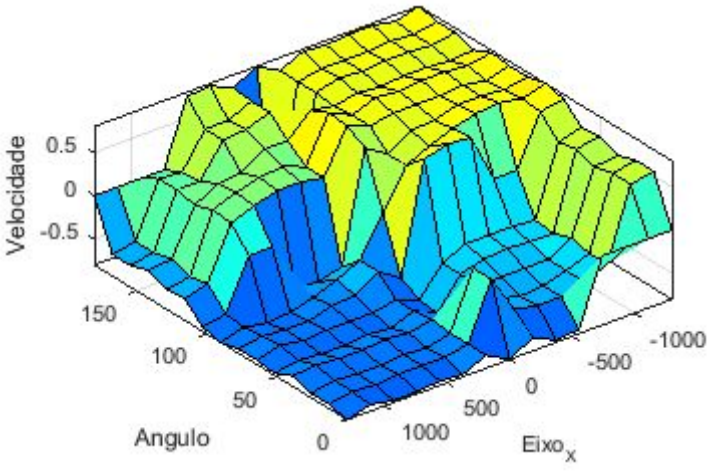
		Eixo X								
		MLE	**LE**	**ME**	**PE**	**CE**	**PD**	**MD**	**LD**	**MLD**
Ângulo	**BN**	R	BD	BD	BD	BD	BD	BD	BD	BD
	MN	R	PD	D	MD	BD	BD	BD	BD	BD
	N	BE	ME	E	BD	MD	MD	BD	BD	BD
	PN	BE	BE	BE	R	D	D	MD	MD	MD
	AL	BE	BE	BE	ME	R	MD	BD	BD	BD
	PP	ME	ME	ME	E	E	R	BD	BD	BD
	P	BE	BE	BE	ME	ME	BE	D	MD	BD
	MP	BE	BE	BE	BE	BE	ME	E	PE	R
	BP	BE	BE	BE	BE	BE	BE	BE	BE	R

Bastante Negativo (BN), Muito Negativo (ML), Negativo (N), Pouco Negativo (PN), Alinhado (AL), Pouco Positivo (PP), Positivo (P), Muito Positivo (MP), Bastante Positivo (BP), Muito Longe a Esquerda (MLE), Longe a Esquerda (LE), Médio a Esquerda (ME), Perto a Esquerda (PE), Centralizada (CE), Perto a Direita (PD), Médio a Direita (MD), Longe a Direita (LD), Muito Longe a Direita (MLD) Bastante para a Esquerda, (BE), Muito para a Esquerda (MB), Esquerda (E), Pouco para a Esquerda (PE), Reto (R), Pouco para a Direita (PD), Muito para a Direita (MD) e Bastante para a Direita (BD).

Autor: O autor (2019).

A resposta do controlador projetado pode ser observado de forma gráfica na Figura 29. O gráfico apresenta a resposta do sistema de controle, com base nas condições de entrada de posição e orientação aplicadas as regras de inferência, resultando na saída de velocidade angular para as rodas da cadeira. Este passo foi realizado utilizando a ferramenta *Suface Viewer* do *MatLab*. No gráfico, os eixos horizontais representam as entradas e o eixo vertical a saída de velocidade angular em rad/s.

Figura 29 - Resposta do controle



Autor: O autor (2019).

4.2.1.1 Simulações do comportamento do controle

Antes de implementar o controle na cadeira de rodas foram realizadas simulações do comportamento da trajetória da cadeira a partir de sua posição inicial, com a finalidade de verificar a funcionalidade do controle projetado. Para isso, a cadeira de rodas foi descrita com base no seu modelo cinemático discretizado, caracterizado pelas Equações 18, 19 e 20.

$$x_{k+1} = x_k + T\, v\cos(\theta_k) \tag{18}$$

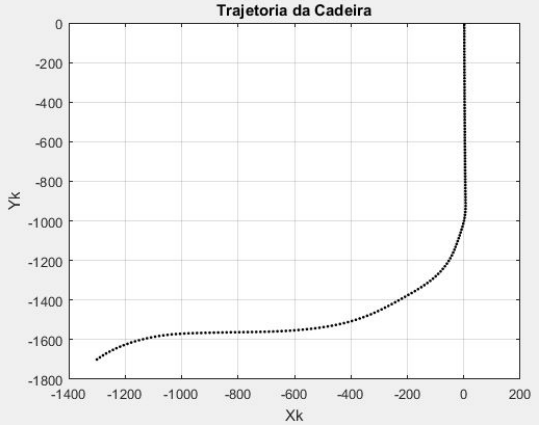
$$y_{k+1} = y_k + T\, v\sin(\theta_k) \tag{19}$$

$$\theta_{k+1} = \theta_k + T\, w \tag{20}$$

Onde x_k, y_k e θ_k representam a posição e angulação atual da cadeira e x_{k+1}, y_{k+1} e θ_{k+1} são a estimativa da posição e angulação depois de um instante de tempo, representado por *T.*

As funções de pertinência e regras de inferência foram implementadas no *script* do MatLab, gerando um conjunto de pontos que representam o deslocamento da cadeira nos eixos x e y. A Figura 30 apresenta o exemplo de uma trajetória simulada da cadeira de rodas com as posições iniciais (-1300,-1700)mm e um ângulo de orientação de 45 graus.

Figura 30 - Simulação da trajetória



Autor: O autor (2019).

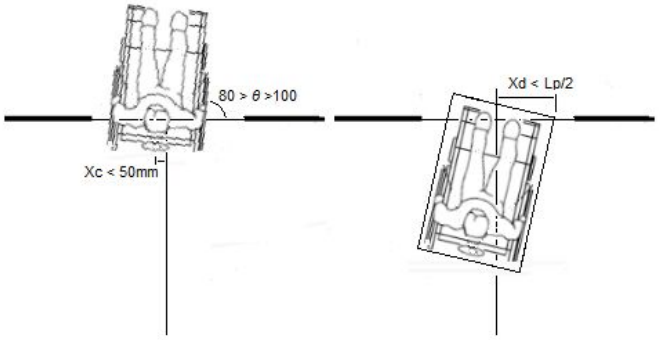
A partir disso, foi determinado quais condições iniciais podem ou não resultar em uma correta trajetória, atingindo o ponto de destino sem que haja colisão. Para isso, os dados de entrada da posição y foram mantidos em -1500mm, variando a posição inicial da cadeira em x de -2000 a 2000mm, com passo de 100mm e o ângulo de orientação de 0 a 180 graus, com um passo de 4,5 graus.

A Figura 31 ilustra as condições que devem ser satisfeitas para que a cadeira possa executar o movimento livre de colisões. A cadeira atinge o objetivo de forma correta quando Y_r >= 0 e X_r apresenta um erro menor que 50mm com um ângulo de orientação entre 80 e 100 graus, como mostra a Figura 31-b.

Devido a posição da cadeira ser dada no centro do eixo das rodas tracionadas, deve-se garantir que suas extremidades não colidam com a porta.

Quando alguma das extremidades atinge o ponto 0 do eixo Y, deve-se observar que a posição em X deve ser menor que a metade da largura da porta, como pode ser visto na Figura 31-a. Caso contrário, a extremidade da cadeira iria colidir no canto da porta.

Figura 31 - Condições para conclusão da trajetória



(a) (b)
Autor: O autor (2019).

Algumas outras condições também podem ser observadas para evitar alguma situação que não atenda a execução correta da trajetória, como uma possível colisão lateral, porém neste trabalho foram somente consideradas as condições apresentadas na Figura 31.

4.2.2 Implementação do controle

O código responsável pelo funcionamento do controlador fuzzy foi implementado em linguagem C, e embarcado em uma plataforma ESP32. O algoritmo recebe a posição inicial determinada pelo processo de localização, por meio de uma comunicação serial com um *notebook*. E então executa o movimento por meio do controlador projetado, como também estima sua posição ao longo de trajetórias por meio de sensores capazes de medir a rotação das rodas.

O funcionamento do controle ao longo da trajetória necessita da sua posição atual. Para tal, foi utilizado o método de localização relativa, que permite estimar sua posição atual através de sua posição anterior. Conhecendo previamente a

velocidade da cadeira, é possível determinar sua posição e orientação (CASTRO, 2017).

A técnica utilizada para determinar a localização relativa foi a odometria, que permite determinar a posição e orientação da cadeira por meio do deslocamento incremental de suas rodas ao longo do tempo. A aplicação desta técnica se mostra adequada em trajetórias de curtas distâncias (CASTRO, 2017).

A medição da rotação das rodas foi realizada por meio do uso de dois *encoders*. Inicialmente, foi necessário verificar quantos pulsos de saída representam um determinado deslocamento nas rodas cadeira. Conhecendo o número de pulsos por volta do eixo do *encoder* (pps), a redução que acopla o motor a roda (*Red*) e o diâmetro das rodas (*D*), podemos determinar o número de pulsos do encoder para o deslocamento de 1mm (P_{1mm}), por meio da Equação 21.

$$P_{1mm} = \frac{pps * Red}{D * \pi} \tag{21}$$

Desta forma, é possível determinar o deslocamento das rodas em milímetros com base no número de pulsos lidos pela ESP-32. As entradas da plataforma, conectadas aos pulsos de saída dos *encoders*, foram configuradas para uma interrupção de borda de subida, contando número de pulsos em um tempo de amostragem determinado de 50ms. Este tempo foi estipulado com base na velocidade de movimento da cadeira e no tempo de resposta do controle da mesma.

O deslocamento linear (Δs) e o angular ($\Delta\Theta$) da cadeira podem ser descritos pela Equação 22 e 23. Onde *Npd* é o número de pulsos da roda direita no tempo de amostragem *T* de 50ms e *Npe* é o número de pulsos da roda esquerda.

$$\Delta s = Tv = \frac{Npd + Npe}{2P_{1mm}} \tag{22}$$

$$\Delta\theta = Tw = \frac{Npd - Npe}{LP_{1mm}} \tag{23}$$

Da mesma forma que nas simulações, para algoritmo determinar a posição da cadeira por meio da odometria, utiliza-se o modelo cinemático de robôs diferenciais. Entretanto, na odometria a posição e orientação atual, x_k, y_k e θ_k, é estimada com base na posição e orientação anterior x_{k-1}, y_{k-1} e θ_{k-1}, diferente da aplicada na simulação. O conjunto de equações a seguir descreve a posição e orientação da cadeira ao longo da trajetória (RODRIGUES, 2017).

$$x_k = x_{k-1} + \Delta s \cos(\theta_{k-1}) \tag{24}$$

$$y_k = y_{k-1} + \Delta s \sin(\theta_{k-1}) \tag{25}$$

$$\theta_k = \theta_{k-1} + \Delta\theta \tag{26}$$

Neste trabalho não foi executado nenhum processo para estimar os erros e determinar a precisão da odometria, apenas foram efetuados testes para verificar se a distância percorrida correspondia com a medida realizada pela odometria.

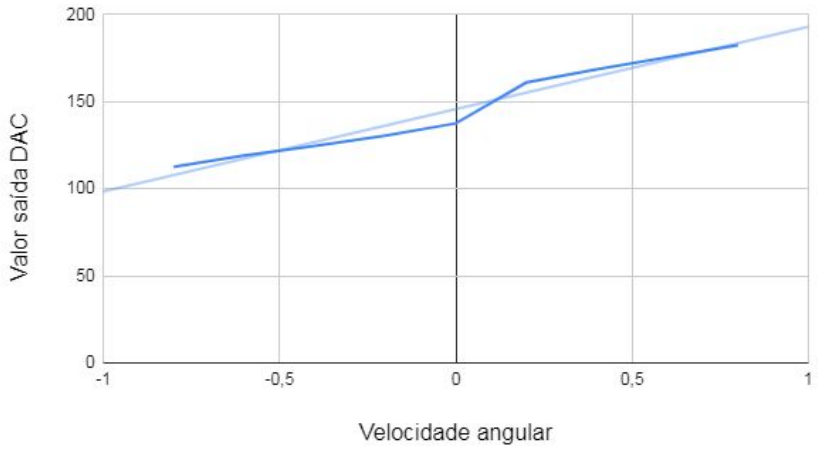
Conhecendo a posição e orientação atual ao longo da trajetória, o controlador fuzzy responde com a velocidade angular necessária para atingir o ponto objetivo de acordo com as regras de inferência estabelecidas na secção 4.2.1.

A plataforma ESP-32 contém duas saídas DAC de 8-bits, que podem receber valores de 0 a 256, proporcionais as saídas analógicas de 0 a 3,3V. Estas saídas são responsáveis por enviar os comandos de velocidade para o controle da cadeira. Desta forma, foi determinado quais valores de 0-256 condizem com a velocidade angular da cadeira em rad/s. Este procedimento foi realizado atribuído valor para a saída do DAC e medindo sua velocidade angular por meio da odometria implementada.

Foi constatado que ao aplicar uma referência de velocidade para a cadeira, ela tem uma resposta diferente, para sua velocidade nas rodas, quando está acelerado do que quando está desacelerando. Isso ocorre devido ao comportamento específico do controle da própria cadeira rodas. Com o objetivo de minimizar essa diferença, foi calculada a média para os valores com a cadeira acelerando e , gerando o gráfico apresentado na Figura 32, que demonstra a relação entre ao valor de saída aplicado ao DAC e a velocidade angular da cadeira. O algoritmo obtém a

velocidade angular pelo controlador fuzzy na faixa de -0,8 a 0,8 rad/s, e converte para um valor a ser enviado para o DAC na faixa de 112 a 182.

Figura 32 - Valor Saída DAC



Autor: O autor (2019).

Com o algoritmo obtendo a posição e orientação inicial da cadeira, assim como sua posição e orientação atual ao longo da trajetória. Por meio do controlador Fuzzy, uma resposta para a velocidade angular é enviada para o controle da cadeira rodas realizar o movimento pretendido.

5 RESULTADOS E DISCUSSÕES

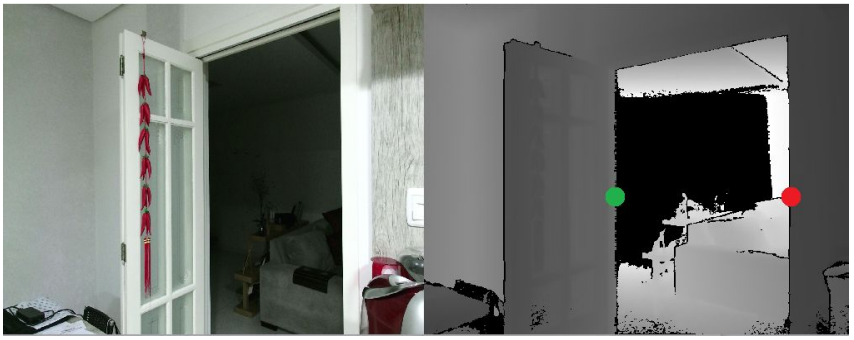
Neste capítulo são descritos os ensaios realizados a fim de verificar e validar os métodos estudados e implementados, apresentado os resultados e suas respectivas análises. Os testes iniciais incidem sobre a implementação do algoritmo de detecção das portas. Posteriormente, são apresentados resultados das simulações do controlador Fuzzy para diferentes condições iniciais. E, por fim, foram realizados testes do sistema implementado na cadeira, extraído sua posição em relação à porta e executando o movimento, e, na sequência, comparando os resultados obtidos com as simulações.

5.1 ENSAIOS DETECÇÃO DE PORTAS

Nesta seção apresentados os resultados para o algoritmo que realiza a detecção de portas pelos métodos descritos na secção 4.1.2. O sensor Kinect foi disposto em diversas posições, em relação às portas, e executado o algoritmo para realizar a detecção em cada situação. Neste momento não é avaliado a precisão nas medidas e sim se o sistema foi capaz de detectar a porta.

Os testes foram realizados com um padrão de porta de 0,8m a 0,9m. O primeiro caso em que o algoritmo foi submetido, é o qual, pelo ponto de vista do Kinect, é possível ser observado dois pontos de descontinuidade, pertencentes aos extremos esquerdo e direito da porta. Conforme ilustra a Figura 33.

Figura 33 - Porta com dois pontos de descontinuidade



Autor: O autor (2019).

Os pontos verde e vermelho indicam onde ocorre a descontinuidade na imagem, ou seja, quando a diferença de distância é positiva a borda é pertencente ao lado direito (vermelho) e quando essa diferença é negativa a borda pertence ao lado esquerdo da porta (verde).

. Quando mais de um par de bordas atendem as condições para ser considerado uma porta, o algoritmo considera como porta os pares que apresentam a menor abertura, como no caso mostrado na Figura 34, em que a porta identificada está representada pelos pontos maiores.

Figura 34 - Porta com quatro pontos de descontinuidade

Autor: O autor (2019).

Em situações que o sensor Kinect não pode observar dois pontos de descontinuidade pertencentes a uma porta, identificando apenas um. Uma sub-rotina no algoritmo, que objetiva encontrar o outro extremo da porta, é executada nestes casos, conforme o método descrito na seção 4.1.2. A Figura 35 mostra a identificação de uma porta para essa situação.

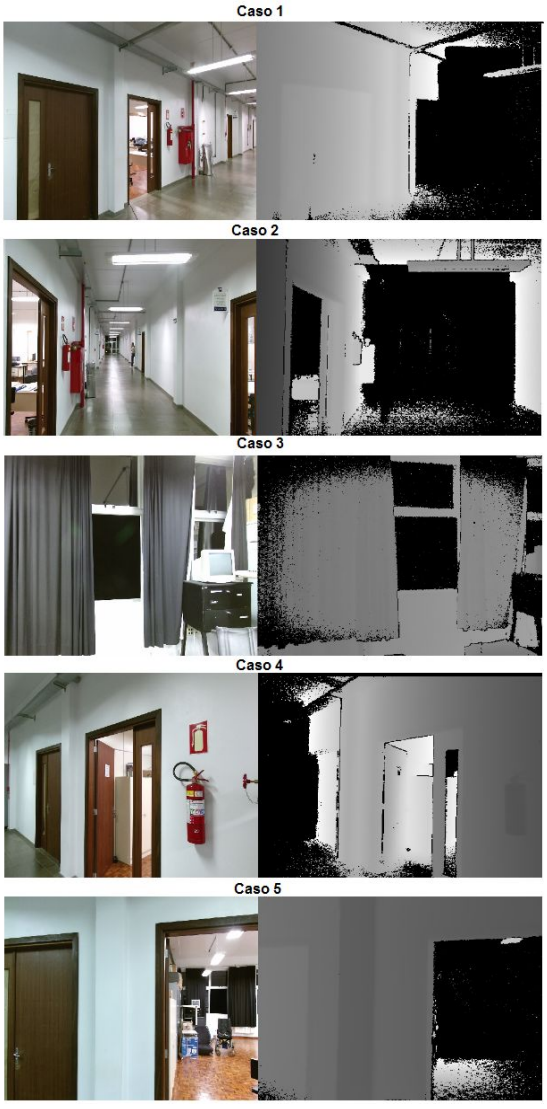
Figura 35 - Com um ponto de descontinuidade

Autor: O autor (2019).

É possível observar pela imagem que somente um ponto de descontinuidade da porta pode ser observado pela varredura do Kinect, o pertencente ao seu extremo direito. O outro externo da porta, marcado em verde, é o ponto que a apresenta a menor distância calculada entre todos os outros da imagem a partir do lado direito da porta, logo este será o extremo esquerdo na porta. Diferente do método de dois pontos de descontinuidade, nessas situações se mais de um salto de profundidade for observado pelo sensor, pode ocorrer uma detecção equivocada.

A Figura 36 apresenta cinco casos em que o algoritmo não detectou a porta ou detectou de forma equivocada.

Figura 36 - Casos de falha na detecção



Autor: O autor (2019).

No primeiro caso, a porta se encontra muito distante do sensor Kinect, além dos limites do sensor. Logo, as bordas estão mais distantes que os limites estabelecidos pelo algoritmo.

No caso 2, o sensor se encontra muito próximo a parede, e neste caso o algoritmo é capaz de identificar as bordas, porém as descarta por não atenderem a condição de angulação necessária de 10 graus entre as bordas da porta.

O caso 3 mostra a detecção equivocada de uma janela, pelo fato do algoritmo se basear apenas na diferença de profundidade para a detecção. Desta forma, aberturas como janelas podem ser consideradas, de maneira errônea, como portas.

Já o quarto caso apresenta uma situação, em que na porta, somente um ponto de descontinuidade pode ser observado. No entanto há a existência de uma janela em sua lateral, que ocasiona uma leitura incorreta de qual é a verdadeira borda pertencente a porta. Pelo método de apenas um ponto de descontinuidade o algoritmo estima a porta de forma incorreta, pois considera a janela como um dos extremos da porta.

Por último, o caso 5 mostra uma situação em que a borda direita da porta se encontra fora do campo de visão do sensor, não sendo possível detectá-la.

Considerando os testes realizados, o algoritmo apresentou um desempenho satisfatório para detectar as portas, principalmente quando é possível observar dois pontos de descontinuidade em seus extremos, falhando apenas nas situações mais desfavoráveis em relação às condições de uso do sensor Kinect.

Quando apenas há a existência de um ponto de descontinuidade na porta, observado ponto de vista do sensor, o algoritmo busca encontrar o outro extremo, por meio da menor medida entre os pixels que se seguem. Entretanto, para este método ser eficiente, o algoritmo precisa encontrar apenas um ponto de descontinuidade. Logo, se em toda a imagem ele encontrar algum outro ponto, uma identificação errônea pode ocorrer, devido a algum ruído ou outro objeto na imagem.

5.2 RESULTADOS SIMULAÇÕES

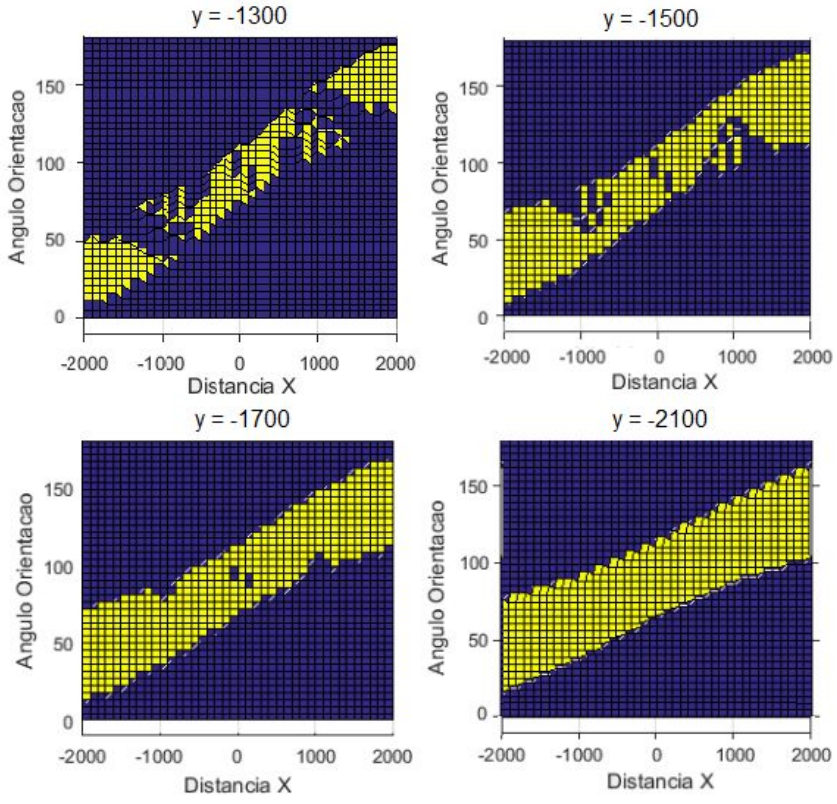
Seguindo a metodologia apresentada na secção 4.2.1.1, os resultados do controlador Fuzzy são apresentados para diferentes posições iniciais da cadeira de

rodas, verificando se sua trajetória atinge o ponto objetivo nas condições estabelecidas.

A partir da posição inicial, o sistema deve conseguir identificar a porta, sempre considerando as limitações do ângulo de visão do sensor Kinect de 70 graus, e observando as dimensões da cadeira a partir do centro das rodas até os pontos extremos para evitar colisões.

Com isso, a Figura 37 apresenta os resultados para as simulações em quatro posições iniciais fixas em y -1300, -1500, -1700 e -2100mm, com a posição em x variando de -2000 a 2000mm e o ângulo de orientação de 0 a 180 graus.

Figura 37 - Simulação para diferentes posições iniciais



Autor: O autor (2019).

O gráfico mostra, nos pontos em amarelo, onde o controle teoricamente é capaz de identificar a porta e realizar a trajetória. Já os pontos em azul são as situações para a posição inicial em que o controle não consegue identificar é nem atingir o ponto de passagem. O padrão no gráfico que forma a curva, em amarelo se

dá pelas limitações de visão do sensor Kinect. Os pontos em azul que invadem a essa curva, são as situações limitadas pelo controlador fuzzy, onde não é capaz de realizar a trajetória de forma correta.

O controlador fuzzy foi projetado para ser capaz de executar a trajetória quando a distância mínima em y da porta é de 1500mm, porém ao levar em conta as dimensões físicas da cadeira e as limitações visuais do sensor, o movimento da cadeira fica restrito aos pontos em amarelo. Quando a cadeira de rodas se encontra mais próxima a porta as posições iniciais onde o controle não consegue realizar o movimento são cada vez mais expressivas.

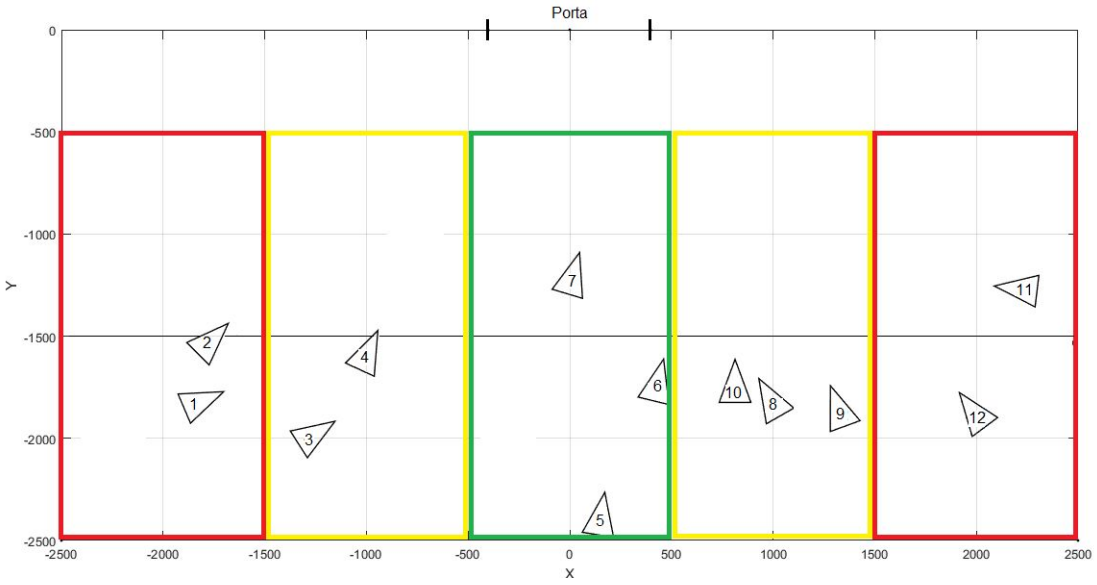
Conforme a distância aumenta em y a limitação do controle tende a diminuir, até que as limitações do sistema se deem apenas pelo Kinect. Pode-se observar que a partir de -2100mm o movimento fica restrito apenas pelas limitações de visão do Kinect.

5.3 TESTE DE DETECÇÃO E MOVIMENTO

Com todo o sistema implementado na cadeira, foram realizados testes do algoritmo de identificação da porta e controle do movimento, com a finalidade de realizar a prova de conceito do sistema proposto. Para isso, a cadeira foi disposta em diversas posições e orientações em frente a uma porta (largura padrão de 0,9m), extraído sua localização e posteriormente executando o movimento da cadeira de rodas com o intuito de atingir o ponto de passagem no centro da porta.

A Figura 38 apresenta as posições em que a cadeira de rodas foi submetida para que os testes fossem executados. Os testes foram divididos em três regiões, considerados como “distância longa” (vermelha), “distância média” (amarelo) e “distância próxima” (verde). Ao todo foram realizados 12 experimentos, conforme numeração vista nos triângulos da Figura 38. Entretanto foram selecionados sete que apresentaram um resultado mais relevante, que são discutidos neste capítulo e os demais resultados se encontram no (Apêndice A).

Figura 38 - Disposições da cadeira nos ensaios



Autor: O autor (2019).

Cada posição inicial extraída pelo algoritmo foi conferida com uma trena laser marca BOSCH, modelo GLM 40 Professional, e assim determinado seu erro. A trajetória para cada posição inicial também foi comparada com a simulada, salientando suas diferenças e indicando se a cadeira conseguiu concluir o movimento livre de colisões.

5.3.1 Distância longe

No quadro onde a cadeira se encontra a longas distância da porta, são analisadas as situações 1 e 11. A Tabela 4 apresenta o resultado para as medidas das bordas, junto com a posição inicial calculada, a posição final atingida pelo controlador Fuzzy, lida pela odometria, e a posição final em x, medida pela trena.

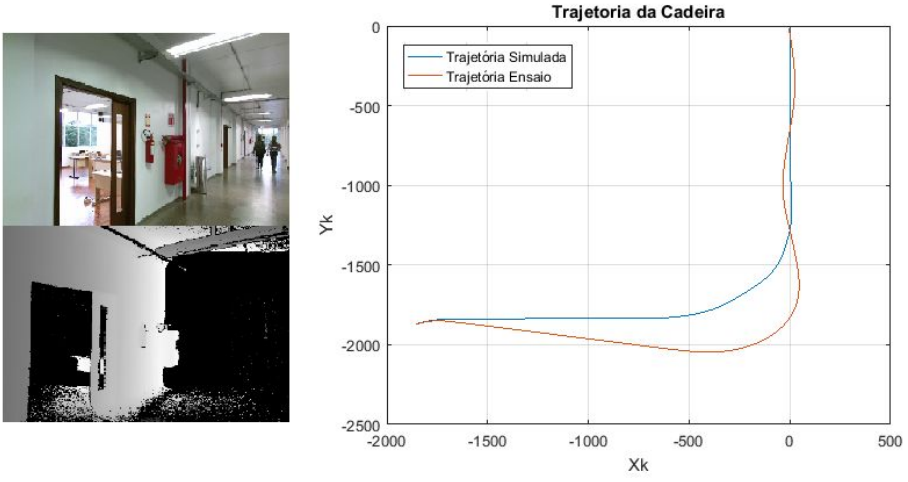
Tabela 4 - Medidas ensaios situação 1

Situação 1		
Borda da Porta		
	Kinect	Trena Laser
Direita [mm]	3170	3150
Esquerda [mm]	2510	2460
Largura Porta	917	910
Cadeira de Rodas		
	Posição Inicial	Posição Final
X [mm]	-1857	-4,94
Y [mm]	-1872	-8,64
θ	22,4	96,2

Autor: O autor (2019).

A Figura 39 mostra a porta identificada e a trajetória realizada pelo controlador fuzzy implementado (laranja), junto com a trajetória simulada no MatLab (azul).

Figura 39 - Trajetória situação 1



Autor: O autor (2019).

Pode-se observar pelo gráfico que a cadeira se deslocou para a direita enquanto se aproximava da porta. Isso ocorre pelo fato de a cadeira de rodas, mesmo com velocidade somente linear, não andar em linha reta. Como o controlador desenvolvido apenas tem como variável de entrada sua posição em *x* e a orientação da cadeira apenas com funções de pertinência para ângulos positivos, não foram realizadas regras para corrigir este deslocamento. Entretanto à medida que a

cadeira se aproxima da porta sua posição e orientação são corrigidas, atingindo o ponto alvo com posição (-4,9, -8,6)mm e orientação de 96 graus.

A Tabela 5 apresenta a situação 11 onde a cadeira se encontra muito próxima a parede em que está a porta, resultando em uma trajetória de colisão.

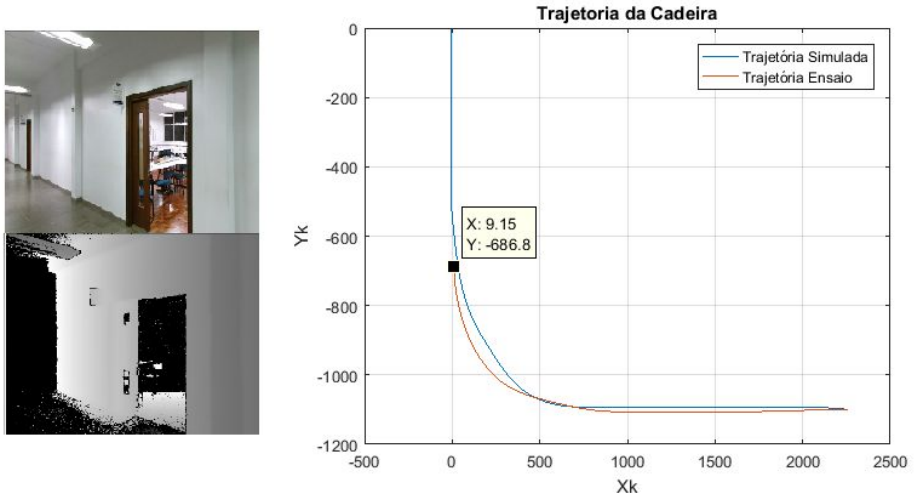
Tabela 5 - Medidas ensaio situação 11

Situação 11		
Borda da Porta		
	Algoritmo	Medido
Direita [mm]	3132	3170
Esquerda [mm]	2286	2330
Largura Porta	930	900
Cadeira de Rodas		
	Posição Inicial	Posição Final
X [mm]	2257	1,5
Y [mm]	-1101	-6,9
θ	173,6	88,2

Autor: O autor (2019).

A Figura 40 mostra tanto a porta identificada quanto a trajetória realizada pelo controlador fuzzy e a trajetória simulada para a situação .

Figura 40 - Trajetória situação 11



Autor: O autor (2019).

Da mesma foram que na situação anterior, a cadeira tendeu para a direita, porém neste caso o decréscimo no ângulo de orientação ficou dentro de uma regra inferência do Fuzzy, fazendo com que houvesse uma correção. Entretanto por a

cadeira estar muito próxima a parede, ela entrou em rota de colisão, sendo necessário a parada do movimento. O movimento foi interrompido no ponto marcado no gráfico em (9,17,-686)mm.

5.3.2 Distância média

No quadro onde a cadeira se encontra a um médio afastamento da porta, são analisadas as situações 4, 8 e 10. As Tabelas 6 e 7 mostram os valores para as medidas das bordas, posição inicial e posição final na situação 4 e 8 respectivamente.

Tabela 6 - Medidas ensaio situação 4

Situação 4		
Borda da Porta		
	Algoritmo	Medido
Direita [mm]	2423	2410
Esquerda [mm]	1954	1930
Largura Porta	898	910
Cadeira de Rodas		
	Posição Inicial	Posição Final
X [mm]	-1054	10,55
Y [mm]	-1648	-6,53
θ	64,23	84,22

Autor: O autor (2019).

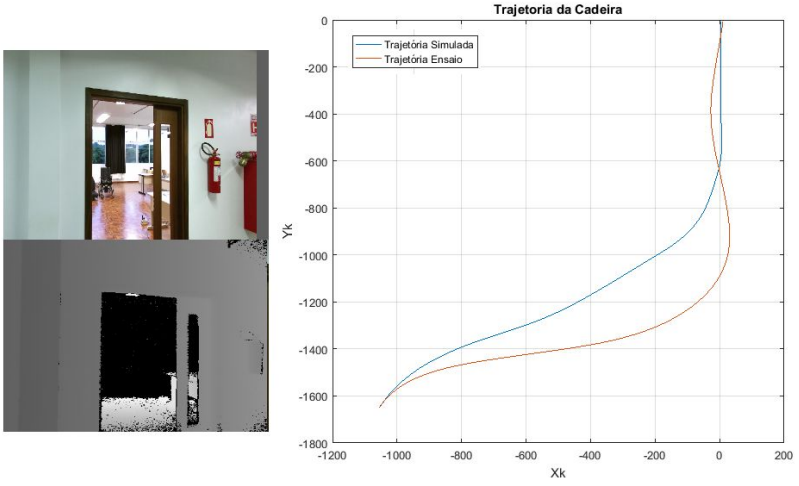
Tabela 7 - Medidas ensaio situação 8

Situação 8		
Borda da Porta		
	Algoritmo	Medido
Direita [mm]	2113	2150
Esquerda [mm]	2527	2450
Largura Porta	833	850
Cadeira de Rodas		
	Posição Inicial	Posição Final
X [mm]	1050	0
Y [mm]	-1809	-2,9
θ	120,6	90,5

Autor: O autor (2019).

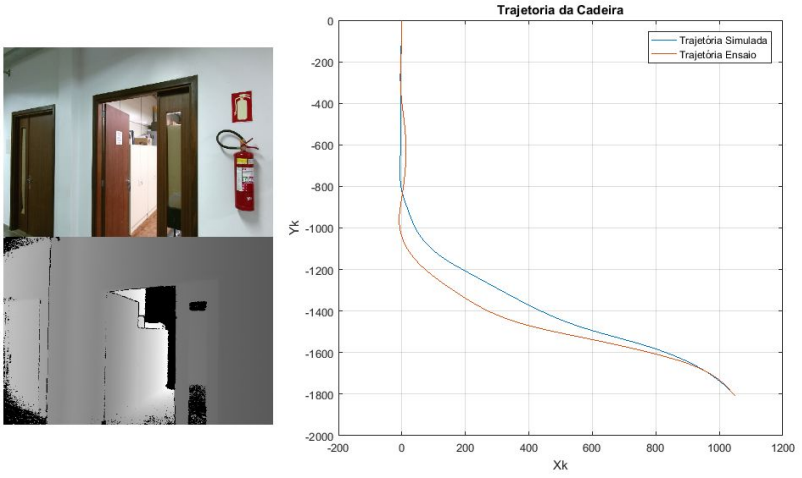
Na Figura 41 e 42 são ilustradas a imagem da identificada e os gráficos da trajetória para a situação 4 e 8.

Figura 41 - Trajetória situação 4



Autor: O autor (2019).

Figura 42 - Trajetória situação 8



Autor: O autor (2019).

Pode-se observar nos gráficos que quando a cadeira precisa realizar curvas para a direita há um desvio maior em relação a trajetória simulada, e quando a cadeira realiza curvas para a esquerda um desvio menor. Essa diferença ocorre devido a cadeira de rodas realizar uma leve curva para direita, mesmo recebendo do controlador unicamente velocidade linear. Contudo nessas situações isso apenas

interfere no trajeto da cadeira, visto que o controle consegue corrigir e atingir o ponto de passagem, livre de colisões.

Outra situação analisada é a de número 10, onde não foi possível realizar o movimento livre colisão. A Tabela 8 apresenta as medidas para esta situação.

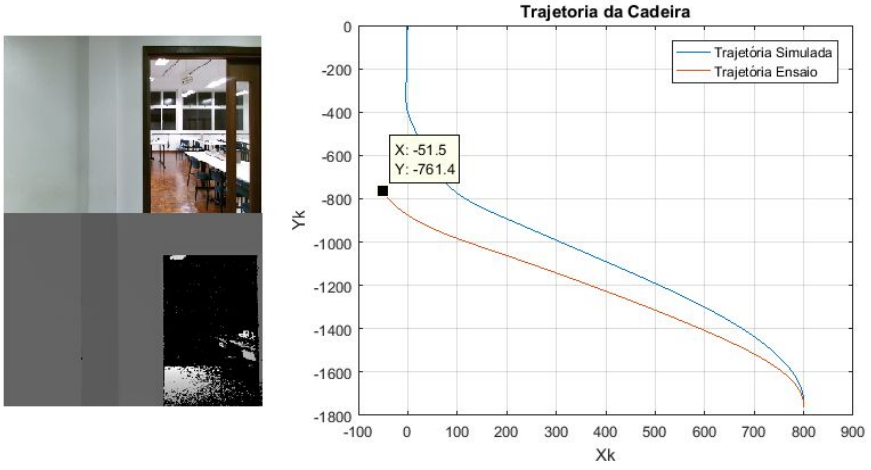
Tabela 8 - Medidas ensaio situação 10

Situação 10		
Borda da Porta		
	Algoritmo	Medido
Direita [mm]	1944	1930
Esquerda [mm]	2272	2250
Largura Porta	875	900
Cadeira de Rodas		
	Posição Inicial	Posição Final
X [mm]	800	-
Y [mm]	-1759	-
θ	89.4	-

Autor: O autor (2019).

A Figura 43 ilustra a porta identificada em conjunto com a trajetória realizada pelo controlador e a trajetória simulada para a situação 10.

Figura 43 - Trajetória situação 10



Autor: O autor (2019).

Nesta situação a cadeira realiza a curva à direita passando um pouco do eixo central da porta e no momento que começa a corrigir sua a posição e orientação, suas extremidades já se encontram próximas demais da porta, resultando em uma

colisão. Essa situação pode se enquadrar na região azul da Figura 38 na qual não é possível realizar o movimento.

5.3.3 Distância próxima

Nas situações onde a cadeira se encontra a distâncias próximas da porta, são apresentados os resultados para as situações 6 e 7. A Tabela 9 e 10 mostra os valores para as medidas das bordas, posição inicial e posição final respectivamente.

Tabela 9 - Medidas ensaio situação 6

Situação 6		
Borda da Porta		
	Algoritmo	Medido
Direita [mm]	2398	2370
Esquerda [mm]	2526	2510
Largura Porta	858	855
Cadeira de Rodas		
	Posição Inicial	Posição Final
X [mm]	413	5,5
Y [mm]	-2203	-11,4
θ	76,3	88

Autor: O autor (2019).

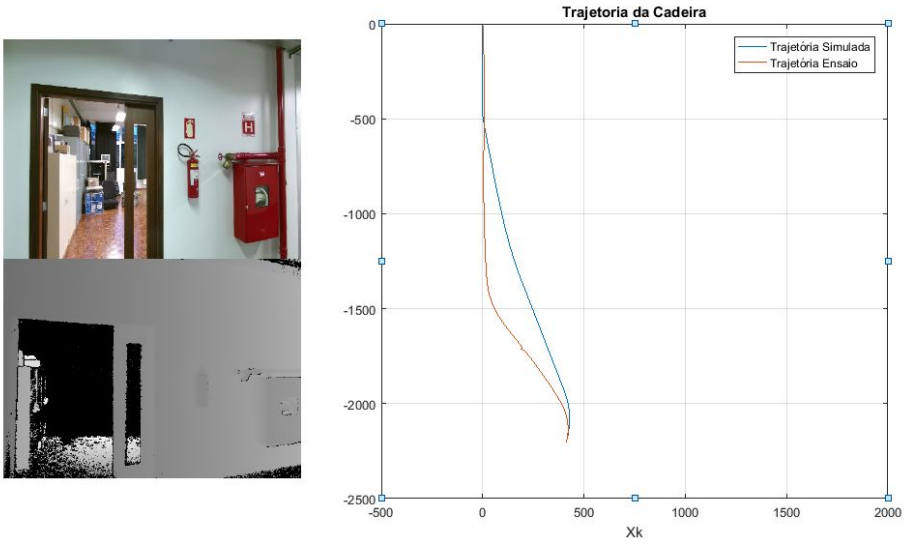
Tabela 10 - Medidas ensaio situação 7

Situação 7		
Borda da Porta		
	Algoritmo	Medido
Direita [mm]	1549	1520
Esquerda [mm]	1528	1530
Largura Porta	823	855
Cadeira de Rodas		
	Posição Inicial	Posição Final
X [mm]	23	19,72
Y [mm]	-1287	-5,23
θ	72,2	90,5

Autor: O autor (2019).

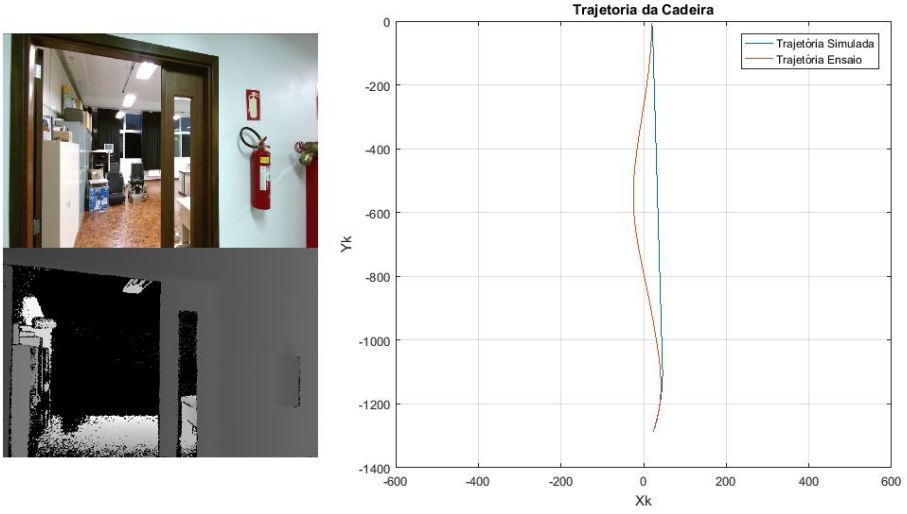
Na Figura 44 e 45 são apresentados os gráficos da trajetória para a situação 6 e 7.

Figura 44 - Trajetória situação 6



Autor: O autor (2019).

Figura 45 - Trajetória situação 7



Autor: O autor (2019).

Nas duas trajetórias a cadeira parte com um ângulo de orientação parecido, o que difere é a distância em relação a porta. Na situação 6 a cadeira precisa realizar uma curva para chegar mais próxima ao centro da porta. Já na situação 7 a cadeira já se encontra em cima do eixo x, porém com um ângulo de orientação de 72 graus, para corrigir a orientação a cadeira gira, no entanto passa da posição e tenta compensar em movimentos oscilatórios até cruzar a porta, isso não ocasiona nenhuma colisão ou erro ao atingir o ponto de passagem, apenas não se fixa em

uma linha reta. isso já não ocorre na situação 6, visto que para corrigir sua orientação a cadeira realiza uma curva mais suave, já atingindo o ponto central com a orientação desejada e a partir deste apenas se movimentando em linha reta.

5.3.4 Análise dos resultados

Com base nos resultados obtidos foi possível observar uma diferença entre a trajetória em situações que o controle necessita realizar uma rotação para a esquerda do que quando realiza para a direita. Dois fatores podem contribuir para que isso ocorra.

O primeiro fator ocorre devido a cadeira de rodas não se deslocar em linha reta quando recebe apenas comandos de velocidade linear, realizando uma leve curva para direita.

O segundo fator que pode influenciar está nas tensões de saída, que são enviadas para o controle da cadeira, que representam as velocidades angulares. A saída necessária para atingir determinada velocidade é diferente quando a cadeira está acelerado do que quando está desacelerando, isso ocorre devido ao comportamento específico do controle da própria cadeira.

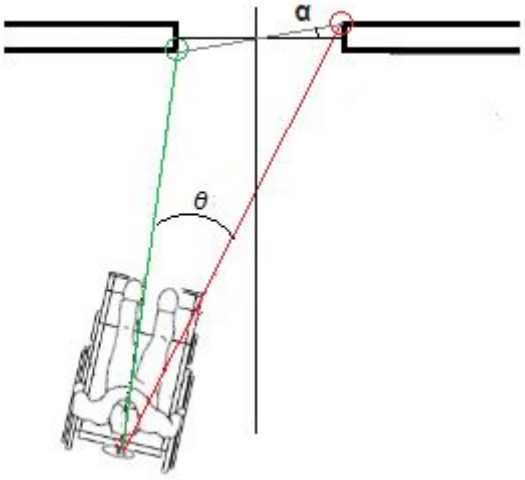
Ainda que isso cause uma pequena modificação na trajetória o controlador Fuzzy projetado é capaz de corrigir o movimento da cadeira, não tendo um impacto significativo no resultado final, desde que haja espaço suficiente para a manobra da cadeira.

Ao concluir o movimento atingindo a posição no ponto de passagem a cadeira parou deslocada, em seu eixo x, com distâncias que variam de 10mm até 80mm, em relação da indicada pelas leituras da odometria, essa diferença pode ser resultado de erros associados a etapa de odometria, visto que não foram realizados calibrações e estimativas para erros para teste método de localização.

Outro ponto importante observado foi que ao concluir o movimento, mesmo com as leitura indicando um ângulo de orientação de 90 graus, a cadeira parou levemente oblíqua, isso ocorre devido a uma erro de estimação do ângulo de orientação da cadeira em relacao a porta, esse erro deve-se pelo algoritmo detectar o ponto de descontinuidade, ou pelo lado de fora ou de dentro do marco da porta.

Quanto mais largo o marco maior o erro na estimação do ângulo. Conforme ilustrado na Figura 46, onde θ é o ângulo de orientação estimado é α o erro da orientação.

Figura 46 - Estimativa de orientacao porta



Autor: O autor (2019).

6 CONCLUSÕES

A melhora na qualidade de vida de deficientes físicos é um importante aspecto na evolução da sociedade. É benéfico o desenvolvimento de tecnologias capazes de proporcionar a estes indivíduos uma maior autonomia e independência nas interações sociais e familiares, igualmente facilitando a inclusão ao mercado de trabalho e educação. Entre as tecnologias assistivas estão as cadeiras de rodas inteligentes.

As cadeiras de rodas inteligentes são projetadas para auxiliar na navegação dos usuários de diversas formas, como evitando colisões, realizando auxílio na condução por passagens estreitas e até mesmo ser capaz de realizar o transporte autônomo. Estas cadeiras estão atreladas a sensores que devem ser capazes de perceber o ambiente ao seu redor, planejar e executar ações.

Diante deste contexto, este trabalho objetivou realizar a prova de conceito de um sistema de navegação assistiva para cadeira de rodas motorizada, capaz de identificar portas e realizar a passagem da mesma de maneira autônoma. Partindo da escolha da técnicas para identificar e localizar portas através de um sensor de profundidade Kinect v2, e o desenvolvimento do sistema de controle para executar o movimento.

O desenvolvimento do projeto iniciou com a escolha da técnica de identificação da porta, com base na diferença de profundidade. O próximo passo foi o projeto do controlador Fuzzy a partir do comportamento da cadeira de rodas, junto com o seu modelo cinemático. Por fim realizaram-se os testes do algoritmo de identificação e localização das portas, bem como as simulações e testes do controle da trajetória.

Os resultados obtidos para algoritmo de detecção e localização das portas foram satisfatórios, alcançando bons níveis de precisão das medidas e sendo muito eficiente na detecção de portas em diversas condições, principalmente quando é possível observar dois pontos de descontinuidade na porta. Para situações onde apenas um ponto de descontinuidade da porta pode ser observado, condições mais favoráveis são necessárias, visto que é mais perceptível a ruídos, resultando em detecções equivocadas.

O controlador fuzzy projetado, obteve a capacidade para executar o movimento da cadeira, partindo da posição e orientação inicial, atingindo o ponto de passagem no centro da porta. O controlador teve um comportamento dentro do esperado, realizando o controle do movimento com características próximas as simulações. Apesar de alguns divergências terem sidos constatadas por características do controle da própria cadeira e possíveis erros relativos a odometria, o resultado final se mostrou satisfatório, com o controle atravessando a porta de maneira adequada.

Por fim, pode-se concluir que o objetivo principal proposto foi alcançado, visto que encontram-se os resultados que comprovam que o sistema testado pode ser aplicado para suprir as necessidades de deficientes físicos usuários de cadeira de rodas.

Com o intuito de dar continuidade a este trabalho, sugere-se como trabalhos futuros a implementação de pré-processamento na imagem para reduzir ruídos melhorando a eficiência do algoritmo de detecção das portas. A integração do sistema em um plataforma embarcada ao invés de utilizar um notebook, a Raspberry PI 4 possui as especificações necessários para o algoritmo desenvolvido.

A Implementação de um sistema Neuro-Fuzzy para o controle do movimento, pode ser apontado como um trabalho futuro, onde deve ser realizado um estudo mais aprofundado para que o sistema realize ajustes, levando em conta a posição inicial da cadeira de rodas. Outro projeto pode ser a adição de um controle em malha fechada para a velocidade da cadeira.

Pode-se também ser implementado o Filtro de Kalman, para realizar a fusão dos dados extraídos pelo sensor Kinect com os dados da odometria, a fim de aumentar a precisão da estimativa de posicionamento da cadeira ao longo da trajetória.

REFERÊNCIAS BIBLIOGRÁFICAS

ABNT. **NBR 9050** – Acessibilidade a edificações, mobiliário, espaços e equipamentos urbanos, 2004.

ALENYA, G.; FOIX, S.; TORRAS C. Using ToF and RGBD cameras for 3D robot perception and manipulation in human environments. **Intelligent Service Robotics**. v. 7, p 211-220. 2014. Disponivel em: https://link.springer.com/article/10.1007/s11370-014-0159-5. Acesso em: 5 jun. 2019.

ALVES, Tiago Giacomeeli. **Sistema de Controle de Pose para uma Cadeira de Rodas Inteligente.** 2018. f 143. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2018.

ASUSTeK Computer Inc. **Xtion Pro Live Specifications**. Disponível em: https://www.asus.com/3D-Sensor/Xtion_PRO_LIVE/specifications/. Acesso em: 14 maio 2019.

AVELL. **Site Avell**. Disponível em: https://avell.com.br/?gclid=EAIaIQobChMIv6Cf-LL25QIVCxGRCh0YkAIcEAAYASAAEgKcEPD_BwE. Acessado em: 15 Out. 2019.

BARROS FILHO, Emanuel Guerra de. **Controle Inteligente Aplicado a uma Mesa de Coordenadas de Dois Graus de Liberdade.** 2011, f 106. Dissertação (Mestre em Ciências de Engenharia Elétrica e Computação) - Universidade Federal do Rio Grande do Norte, Natal, 2011.

BELO, F. A. W. **Desenvolvimento de Algoritmos de Exploração e Mapeamento Visual para Robôs Móveis de Baixo Custo.** 2006. f 276. Dissertação (Mestrado em Engenharia Elétrica) - Pontifícia Universidade Católica do Rio de Janeiro - Rio de Janeiro, 2006.

BECKER, Marcel. **Aplicação de Tecnologias Assistivas e Técnicas de Controle em Cadeira de Rodas Inteligentes.** 2000. f 192. Tese (Doutorado em Engenharia Mecânica) - Universidade Estadual de Campinas, Campinas, 2000.

BEZZERA, Glauber Gomes. **Localização de um Robô Móvel Usando Odometria é Marcos Naturais.** 2004. f 112. Dissertação (Mestrado em Ciências) - Universidade Federal do Rio Grande do Norte, Natal, 2004.

BITTENCOURT, Z. L. C. el. al. Expectativas quanto ao uso de Tecnologia Assistiva. **Journal of Research in Special Educational Needs.** São Paulo, v. 16, n. s1, p. 492-496, 2016.

BORENSTEIN, Johann, FENG Liqiang. Measurement and Correction of Systematic Odometry Errors in Mobile Robots. **IEEE Transactions on Robotics and Automation.** Out. 1996.

BOSCH, **GLM 40 Professional:** Manual Original, 2015. Disponível em: https://www.bosch-professional.com/binary/ocsmedia/optimized/full/o234011v21_1_609_92A_23E_1606.pdf Acessado em: 05 de Nov. 2019.

BRAGA, Rodrigo Antonio Marques. **Plataforma de Desenvolvimento de Cadeiras de Rodas Inteligentes.** 2010. 246 f. Dissertação (Doutorado em Engenharia Informática) - Universidade do Porto. Porto, 2010.

BURHANPURKAR Maya et. al. Cheap or Robust? The Practical Realization of Self-Driving Wheelchair Technology. **2017 International Conference on Rehabilitation Robotics.** Londres, Inglaterra, jul. 2017.

CASTRO, André Luiz Figueiredo. **Uma Nova Abordagem para Identificação e Reconhecimento de Marcos Naturais Utilizando Sensores RGB-D.** 2017. f 99. Dissertação (Pós-Graduação em informática) - Universidade Federal da Paraíba, João Pessoa, 2017.

CAVALCANTI, José Homero Feitosa; CAVALCANTI, Mônica Tejo; SOUTO, Cícero Rocha; MELO, Hiran de. **Lógica Fuzzy aplicada às engenharias**. João Pessoa - PB: 2014.

DAI, DaWei et. al. Detecting, Locating and Crossing a Door for a Wide Indoor Surveillance Robot. **2013 IEEE International Conference on Robotics and Biomimetics.** Shenzhen, China, mar. 2013.

DESAI, S.; MANTHA, S. S.; PHALLE, V. M. Advances in Smart Wheelchair Technology. **2017 International Conference on Nascent Technologies in the Engineering Field.** Mumbai, 2017.

FREEDOM. **Site Freedom Produto**. Disponível em: http://www.freedom.ind.br/produto/saude/cadeiras-de-rodas-motorizadas/freedom-compact-13/. Acessado em: 07 jul. 2019.

INTEL Corporation. **RealSense**. Disponível em: https://www.intelrealsense.com/stereo-depth/?utm_source=intelcom_website&utm_medium=button&utm_campaign=day-to-day&utm_content=D400_learn-more_button&_ga=2.84580185.1119786037.1560889531-262148928.1555369181. Acesso em: 8 jul. 2019.

JUNG, Changbae; CHUNG, Woojin. Accurate Calibration of Two Wheel Differential Mobile Robots by Using Experimental Heading Errors. **2012 IEEE International Conference on Robotics and Automation.** Saint Paul, USA, jul. 2012.

KAKILLIOGLU, B.; OZCAN, K.; VELIPASALAR, S. Doorway Detection for Autonomous Indoor Navigation of Unmanned Vehicles. **2016 IEEE International Conference on Image Processing.** Phoenix, USA, Set. 2016.

KHOSHELHAM, Kourosh; ELBERINK, Sander Oude. Accuracy and resolution of kinect depth data for indoor mapping applications. **Sensors**, v. 12, n. 2, p.

1437-1454, 2012. Disponível em: https://www.mdpi.com/1424-8220/15/11/27569. Acesso em: 6 jun 2019.

LOPES, Isaia Lima; OLIVEIRA, Flavia Aparecida; PINHEIRO, Carlos Alberto Marari. **Inteligência artificial.** 1. ed. Rio de Janeiro . Elsevier, 2014.

MACIEL, Guilherme Marins. **Sistemas de Navegação Semi Autônomo para Robótica Móvel Assistiva Baseado em Movimentos de Cabeça e Comandos de Voz.** 2018. f 100. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal de Juiz de Fora. Juiz de Fora, 2018.

MAGALHÃES, Sandro Augusto Costa. **Controlo das trajetórias de um robô móvel de alto desempenho.** 2017. f 124. Dissertação (Mestrado em Engenharia Electrotécnica e Computadores) - Faculdade de Engenharia Universidade do Porto, Porto, Portugal, 2017.

MARCHI, Jeruza. **Navegação de Robôs Móveis Autônomos:** Estudo e Implementação. 2001. f 132. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal de Santa Catarina. Florianópolis, 2001.

NASCIMENTO JUNIOR, Amadeu do. **Robotização de uma cadeira de rodas motorizada: arquitetura, modelos, controle e aplicações.** 2016. f 122. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal de Campinas, Campinas, 2016.

OMRANE, H.; MASMOUDI, M. S.; MASMOUDI, M. Fuzzy Logic Based Control for Autonomous Mobile Robot Navigation. **Hindawi Publishing Corporation.** maio 2016.

ONG, K. W.; SEET, G.; SIM, S. K. Sharing and Trading in a Human-Robot System. **INTECH Open Access Publisher.** 2005. Disponivel em: https://www.intechopen.com /books/cutting_edge_robotics/sharing_and_trading_in_a_human-robot_system. Acesso em: 6 maio 2019.

OSMAN, Husna Izzati et. al. Entryway Detection Algorithm using Kinect's Depth Camera for UAV Application. **2017 IEEE 8th Control and System Graduate Research Colloquium.** Shah Alam, Malaysia, Ago. 2017.

PAGLIARI, Diana; PINTO, Livio. Calibration of kinect for xbox one and comparison between the two generations of Microsoft sensors. **Sensors**, v. 15, n. 11, p. 27569-27589, 2015. Disponível em: https://www.mdpi.com/1424-8220/15/11/27569. Acesso em: 5 jun. 2019.

PERDIGÃO, Jorge da Silva. **Collaborative-Control based Navigation of Mobile Human-Centered Robots.** 2017, Dissertação (Mestrado em Engenharia Elétrica e Computação) - Universidade de Coimbra, Coimbra, Portugal, 2014.

PIZZETTA, Augusto Boff. I**mplementação de um Sistema Auxiliar de Mobilidade para Deficientes Visuais Baseado em Câmera RGB-D.** 2016. f 128. Trabalho de Conclusão (Engenharia Elétrica) - Universidade de Caxias do Sul, Bento Gonçalves, 2016.

PRADO, Marcos Gomes. **Planejamento de Trajetória para Estacionamento de Veículos Autônomos.** 2013. f 84. Dissertação (Mestrado em Ciências de Computação e Matemática) - Universidade de São Paulo, São Paulo, 2013.

QUINTANA, B. et. al. Door detection in 3D coloured point clouds of indoor environments. **Automation in Construction 85**. p. 146-166, 2018.+

RANSAN, Bernardo Ernesto Moro. **Bypass para uma Cadeira de Rodas Motorizada**. 2019. Universidade de Caxias do Sul, Bento Gonçalves, 2019.

RASPBERRY. **Site da Raspberry Pi**. Disponível em: https://www.raspberrypi.org/. Acesso em: 08 jul. 2019

RODRIGUES, Ana Rita Marques. **Planeamento de Trajetórias e Controlo de um Robô Omnidirecional.** 2017. f 101. Dissertação (Mestrado em Engenharia Electrotécnica e Computadores) - Faculdade de Engenharia Universidade do Porto, Porto, Portugal, 2017.

SANDEEP, B.S.;SUPRIYA, P. Analysis of Fuzzy Rules for Robot Path Planning. **2016 Intl. Conference on Advances in Computing, Communications and Informatics.** Jaipur, India, set. 2016. Disponivel em: https://ieeexplore.ieee.org/document/7732065. Acesso em: 12 jun. 2019.

SILVA, Luciano. **Segmentação de Imagens de Profundidade por Detecção de Bordas.** 2000. f 113. Dissertação (Mestrado em Informática) - Universidade Federal do Paraná, Curitiba, 2000.

SIMÕES, Marcelo Godoy; SHAW, Ian S. **Controle e Modelagem Fuzzy.** 2. ed. São Paulo. Edgard Blucher, 2014.

SOUZA, Stephanie Kamarry Alves. **Planejamento de Movimento para Robôs Móveis Baseado em uma Representação Compacta da Rapidly-Exploring Random tree (RRT).** 2017. f. 67. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Sergipe, São Cristóvão, 2017.

TORRES, João Luís Vila Chã. **Sistema de Localização Indoor para o robô telemóvel "Robobo".** 2018. f 70. Dissertação (Mestrado em Engenharia Electrotécnica e Computadores) - Faculdade de Engenharia Universidade do Porto, Porto, Portugal, 2018.

WANG, Kaiyu. Research and Implementation of Automatic Fuzzy Garage Parking System Based on FPGA. **MATEC Web of Conferences.** 2016.

WORLD HEALTH ORGANIZATION. **Relatório mundial sobre a deficiência**; tradução Lexicus Serviços Lingüísticos. São Paulo: SEDPcD, 2012. 334 p. Disponível em: https://apps.who.int/iris/bitstream/handle/10665/44575/9788564047020_por.pdf;jsessionid=6C99C402A537DE2926F25B3B5A7F04FE?sequence=4. Acesso em: 12 abr. 2019.

YUAN, T. H. et. al. An Automated 3D Scanning Algorithm using Depth Cameras for Door Detection. **2015 International Electronics Symposium.** 2015.

ZHOU, Y. et. al. Kinect Depth Image Based Door Detection for Autonomous Indoor Navigation. **The 23rd IEEE International Symposium on Robot and Human Interactive Communication.** Edinburgh, Escócia, Ago. 2014.

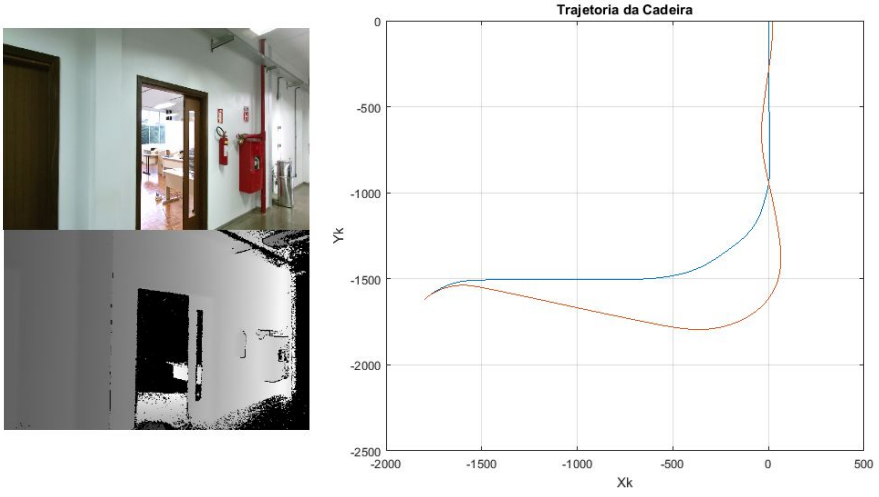
APÊNDICE A - RESULTADOS DE DETECÇÃO E TRAJETÓRIA

Tabela 11 - Medidas ensaio situação 2

Situação 2		
Borda da Porta		
	Algoritmo	Medido
Direita [mm]	2969	2940
Esquerda [mm]	2313	2260
Largura Porta	892	910
Cadeira de Rodas		
	Posição Inicial	Posição Final
X [mm]	-1801	21,93
Y [mm]	-1622	-5,97
θ	44,8	90,5

Autor: O autor (2019).

Figura 47 - Trajetória situação 2



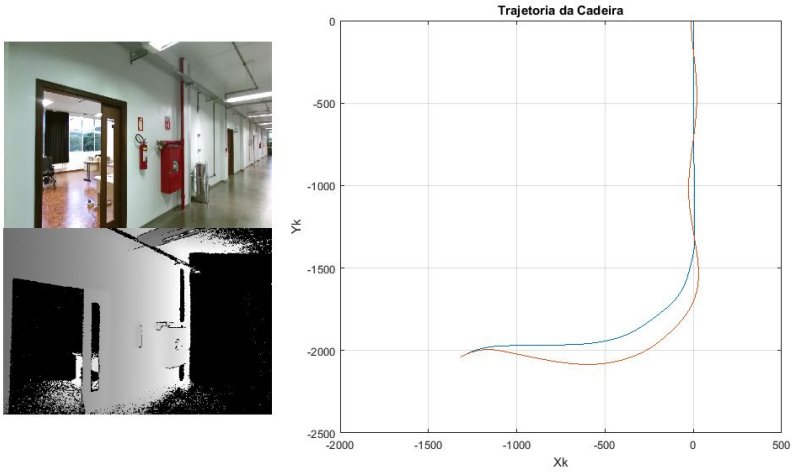
Autor: O autor (2019).

Tabela 12 - Medidas ensaio situação 3

Situação 3		
Borda da Porta		
	Algoritmo	Medido
Direita [mm]	2893	2850
Esquerda [mm]	2383	2320
Largura Porta	917	910
Cadeira de Rodas		
	Posição Inicial	Posição Final
X [mm]	-1317	-12,35
Y [mm]	-2040	-4,65
θ	32,5	90.53

Autor: O autor (2019).

Figura 48 - Trajetória situação 3



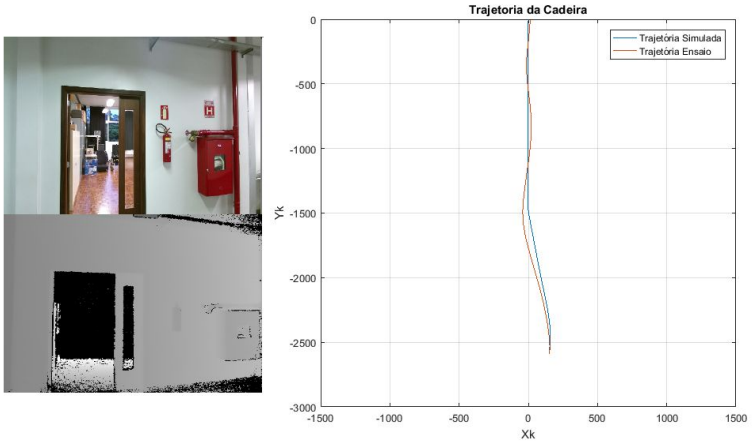
Autor: O autor (2019).

Tabela 13 - Medidas ensaio situação 5

Situação 5		
Borda da Porta		
	Algoritmo	Medido
Direita [mm]	2806	1870
Esquerda [mm]	2841	1290
Largura Porta	855	910
Cadeira de Rodas		
	Posição Inicial	Posição Final
X [mm]	150	13,3
Y [mm]	-2591	-4,47
θ	80,7	87,7

Autor: O autor (2019).

Figura 49 - Trajetória situação 5



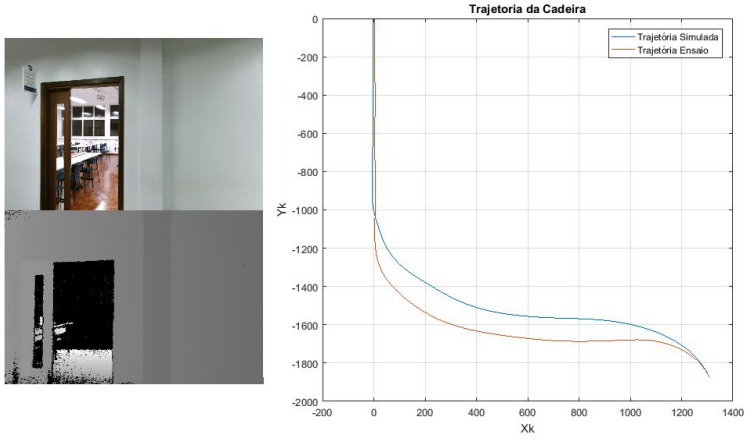
Autor: O autor (2019).

Tabela 14 - Medidas ensaio situação 9

Situação 9		
Borda da Porta		
	Algoritmo	Medido
Direita [mm]	2263	2210
Esquerda [mm]	2744	2730
Largura Porta	875	850
Cadeira de Rodas		
	Posição Inicial	Posição Final
X [mm]	1307	4,4
Y [mm]	-1872	-3
θ	109,4	90

Autor: O autor (2019).

Figura 50 - Trajetória situação 9



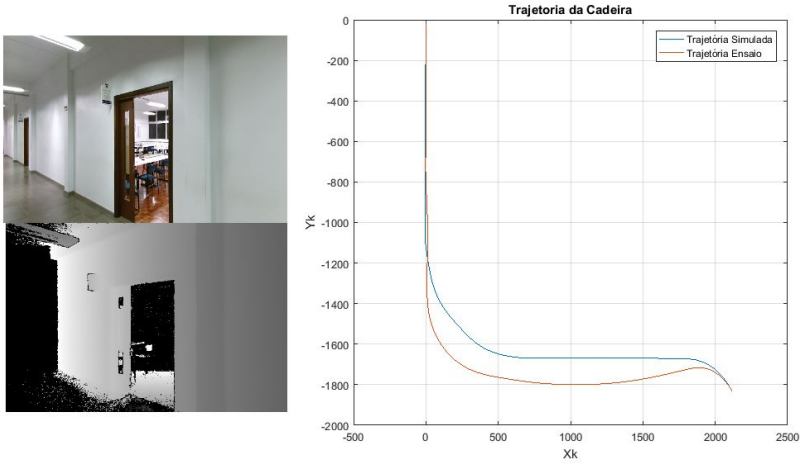
Autor: O autor (2019).

Tabela 15 - Medidas ensaio situação 12

Situação 12		
Borda da Porta		
	Algoritmo	Medido
Direita [mm]	3301	3280
Esquerda [mm]	2618	2600
Largura Porta	933	900
Cadeira de Rodas		
	Posição Inicial	Posição Final
X [mm]	2114	3,4
Y [mm]	-1830	-7,6
θ	127.2	90,5

Autor: O autor (2019).

Figura 51 - Trajetória situação 12



Autor: O autor (2019).

ANEXO A - DADOS TÉCNICOS TRENA LASER

Tabela 16 - Dados técnicos trena laser

Medidor de distâncias digital laser	GLM 40
Faixa de medição (tipicamente)	0,15 – 40 m [A)]
Faixa de medição (tipicamente, condições desfavoráveis)	20 m [B)]
Exatidão de medição (tipicamente)	± 1,5 mm [A)]
Precisão de medição (tipicamente, condições desfavoráveis)	± 3,0 mm [B)]
Mínima unidade de indicação	1 mm
Temperatura de funcionamento	– 10 °C ... + 45 °C
Temperatura de armazenamento	– 20 °C ... + 70 °C
Máx. humidade relativa do ar	90 %
Classe de laser	2
Tipo de laser	635 nm, < 1 mW
Diâmetro do raio laser (a 25 °C) aprox. – a uma distância de 10 m – a uma distância de 40 m	 9 mm [C)] 36 mm [C)]
Desligamento automático após aprox. – Laser – Ferramenta de medição (sem medição)	 20 s 5 min
Peso conforme EPTA-Procedure 01:2014	0,09 kg
Dimensões	105 x 41 x 24 mm
Tipo de proteção	IP 54 (protegido contra pó e projeção de água) [D)]
Pilhas	2 x 1,5 V LR03 (AAA)
Pilhas recarregáveis	2 x 1,2 V HR03 (AAA)
Medições individuais por conjunto de pilhas	5000
Ajuste da unidade de medida	m, ft, in

Autor: Adaptado Bosch (2015).

Figura 52 - Trena laser Bosch

Autor: Adaptado Bosch (2015).