UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

JURASILDO OLIVEIRA REINALDO

RECONHECIMENTO DE OBJETOS NO DESENVOLVIMENTO DE UM SISTEMA DE NAVEGAÇÃO INTELIGENTE PARA ROBÔS MÓVEIS

MOSSORÓ - RN
2015

JURASILDO OLIVEIRA REINALDO

RECONHECIMENTO DE OBJETOS NO DESENVOLVIMENTO DE UM SISTEMA DE NAVEGAÇÃO INTELIGENTE PARA ROBÔS MÓVEIS

Orientador: Profº Dr. Anderson Abner de Santana Sousa

MOSSORÓ - RN
2015

Catalogação da Publicação na Fonte.
Universidade do Estado do Rio Grande do Norte.

Reinaldo**,** Jurasildo Oliveira
Reconhecimento de objetos no desenvolvimento de um sistema de navegação inteligente para robôs móveis **/** Jurasildo Oliveira Reinaldo – Mossoró, RN, 2015.
63 f.

Orientador(a): Prof. Dr. Anderson Abner de Santana Sousa

Dissertação (Mestrado). Universidade do Estado do Rio Grande do Norte. Programa de Pós Graduação em Ciência da Computação

1. Sistema de Navegação. 2. Reconhecimento de Objetos. 3. Ontologias. I. Sousa, Anderson Abner de Santana. II. Universidade do Estado do Rio Grande do Norte. III.Título.

UERN/ BC CDD 004

Bibliotecário: Sebastião Lopes Galvão Neto – CRB - 15/486

JURASILDO OLIVEIRA REINALDO

RECONHECIMENTO DE OBJETOS NO DESENVOLVIMENTO DE UM SISTEMA DE NAVEGAÇÃO INTELIGENTE PARA ROBÔS MÓVEIS

APROVADA EM: 28 de janeiro de 2015.

BANCA EXAMINADORA

Prof. Dr. Anderson Abner de Santana Sousa
Orientador

Prof. Dr. Luiz Marcos Garcia Gonçalves.
Avaliador Externo

Profa. Dra. Rosiery da Silva Maia.
Coorientadora

Prof. Dr. Wilfredo Blanco Figuerola.
Avaliador do Programa

Dedico esse trabalho a toda minha família e aos amigos que fiz durante toda essa jornada.

″Nas grandes batalhas da vida, o primeiro passo para a vitória é o desejo de vencer!″

Mahatma Gandhi

AGRADECIMENTOS

A Deus, que de uma forma inexplicável conseguiu trilhar caminhos que me permitiram chegar até aqui;

À minha família, por todo o apoio e carinho;

Aos amigos que fiz durante esses dois anos de pós-graduação, em especial a meu amigo Carlos Eduardo (Novinho) e a turma do condomínio ;

Ao Professor Anderson Abner, pela paciência e dedicação com que me orientou durante todo esse tempo;

Aos Professores da banca;

A CAPES e a FAPERN pelo apoio financeiro.

RESUMO

O problema da navegação no contexto de robôs móveis pode ser considerado um dos maiores desafios relacionado ao campo da robótica. Tornar um robô independente, capaz de se locomover por ambientes desconhecidos e tomar decisões em meio a situações inesperadas, são funções de um sistema de navegação ideal para ser utilizado em robôs móveis. Sabendo das dificuldades encontradas nessa área, resolveu-se nesse trabalho, desenvolver um sistema de navegação que permitisse ao robô tomar decisões e consequentemente mudar seu comportamento, de acordo com o(s) objeto(s) reconhecido(s). Ao contrário da maioria dos sistemas desenvolvidos, aonde os comportamentos de um robô são assumidos com base apenas em informações de distância, o sistema apresentado neste trabalho, utiliza imagens capturadas por um sensor RGB-D e que posteriormente são processadas pelo algoritmo SURF, para que ao reconhecer um objeto previamente cadastrado, o robô consiga saber suas características e como se comportar a partir daquele momento. Cada objeto que o sistema é capaz de reconhecer foi descrito formalmente utilizando o conceito de ontologias, o que possibilitou que fosse criada uma base de conhecimentos aonde diferentes instâncias de um mesmo objeto pudessem ser armazenadas. Para validação do trabalho, foram realizados alguns experimentos envolvendo comportamentos reativos e deliberativos, assim como o método de reconhecimento de objetos, a fim de se verificar o funcionamento dos módulos do sistema, de forma independente. O primeiro teste levou em consideração o processo de reconhecimento de objetos, quanto a sua eficiência. Já o segundo, buscou verificar o grau de reação do robô ao receber estímulos provenientes do ambiente, e assim, validar os comportamentos reativos. O terceiro experimento buscou testar o comportamento do robô ao reconhecer dois objetos com diferentes propriedades, um estático e o outro dinâmico. O ultimo teste foi realizado para verificar como o sistema se comporta de forma geral, adotando tanto o reconhecimento de objetos, quanto os comportamentos reativos e deliberativos. No geral, com o resultado dos testes, pode-se concluir que o sistema atendeu as expectativas, modificando o comportamento do robô à medida que ele reconhece um determinado objeto e conseguindo reagir espontaneamente a obstáculos inesperados.

Palavras-chave: Sistema de Navegação, Comportamentos Adaptativos, Reconhecimento de Objetos e Ontologias.

ABSTRACT

The navigation problem in the mobile robotics context is one of the most difficult challenges in Robotics area. To make a robot totally independent, able to motion trought an unknown enviroment and to take decisions in some unexpected situations, are functions of a ideal navigation system for mobile robots. Knowing the difficults found in this area, decided up in this work, to develop a navigation system able to enable the robot to take decisions and consequently change its behavior, in accordance with the recognize objects. In contrast to the most developed systems, where the behavior of a robot are assumed on the basis of distance information, the system showed in this work, use captured images by a RGB-D sensor that later are processed by the SURF algorithm, where after recognize an object that was previously registered, the robot being able to identify the object's characteristics to behavior in according from that moment. Each object that the system is able to regonize was formally described using the ontologies concepts, it allowed create a knowledge basis where different instances of the same object could be storage. To validate this work, it was made some experiments involving reactive and deliberative behavior, as well as the object recognition method, in order to verify the system modules working. The first experiment evaluated the object recognition efficiency. The second one verified the reaction level of the robot when it recieves stimulus coming from the environment in order to validate the reactives behaviors. The third experiment tried to check the robot behavior after it recognized two objects, one static and another one dynamic, with different properties. And the last one was executed to verify the system behavior in the general, activating the object recognition, and reactive and deliberative behaviors. On the whole, with the result of the tests, we can conclude that the system attempted our expectations, modifying the robot's behavior according to a recognized object and reacting spontaneously to unexpected. obstacle.

Key-words: Navigation system, Adaptive behavior, Objects recognize and Ontologies.

LISTA DE ILUSTRAÇÕES

Figura 1 – Etapas de um processo de navegação robótico. 14
Figura 2 – Predição de movimentos aplicado à navegação de robôs. 16
Figura 3 – Representação estrutural de um veleiro. 24
Figura 4 – Mapeamento por sonar utilizando uma grade de ocupação. 26
Figura 5 – Mapeamento Topológico. 27
Figura 6 – Localização relativa. 30
Figura 7 – Decomposição celular exata. (a) decomposição do ambiente em células; (b) grafo de conectividade construído a partir da divisão do ambiente. 32
Figura 8 – Decomposição celular aproximada. (a) configuração do ambiente; (b) decomposição do ambiente em células; (c) aproximações realizadas pela discretização. 33
Figura 9 – Campo potencial. 34
Figura 10 – Paradigmas robóticos. (a) Paradigma hierárquico. (b) Paradigma reativo. (c) Paradigma Híbrido (hierárquico/reativo). 36
Figura 11 – DAMN Árbitro e Comportamentos. 38
Figura 12 – Fluxo de execução do sistema 39
Figura 13 – Arquitetura do sistema 40
Figura 14 – Função de navegação de Manhattan. 45
Figura 15 – Plataforma robótica utilizada. 47
Figura 16 – Ambiente de testes 48
Figura 17 – Mapa em grade de ocupação 48
Figura 18 – Interface gráfica do reconhecimento de objetos. (a) Objeto cadeira, (b) Objeto robô X80 49
Figura 19 – Comportamentos reativos - Odometria, Mapeamento e Planejamento 50
Figura 20 – Comportamentos deliberativos - Odometria, Mapeamento e Planejamento (Cadeira). 51
Figura 21 – Comportamentos deliberativos - Variação da velocidade (Cadeira). . 52
Figura 22 – Comportamentos deliberativos - Odometria, Mapeamento e Planejamento (Robô X80). 53
Figura 23 – Comportamentos deliberativos - Velocidade (Robô X80). 54
Figura 24 – Comportamentos deliberativos e reativos - Odometria, Mapeamento e Planejamento. 55
Figura 25 – Comportamentos deliberativos e reativos - Velocidade. 56
Figura 26 – Ontologia geral do sistema 62

LISTA DE TABELAS

Tabela 1 – Vantagens e desvantagens da abordagem métrica 28
Tabela 2 – Vantagens e desvantagens da abordagem topológica 29
Tabela 3 – Atributos para uma instância de Cadeira 41
Tabela 4 – Descrições e comportamentos dos objetos reconhecidos 49

LISTA DE ABREVIATURAS E SIGLAS

SGM	Semantic Grid Map
HMM	Hidden Markov Model
SVM	Support Vector Machine
SLAM	Simultaneous Localization and Mapping
RGB-D	Red Green Blue e Depth
DAMN	Distributed Architecture for Mobile Navegation
SURF	Speeded-Up Robust Feature
OPENCV	Open Source Computer Vision Library
FLANN	Fast Library for Approximate Nearest Neighbors

SUMÁRIO

1 INTRODUÇÃO 14
1.1 MOTIVAÇÃO E JUSTIFICATIVA 15
1.2 OBJETIVO GERAL 17
1.3 OBJETIVOS ESPECÍFICOS 17
1.4 CONTRIBUIÇÕES 17
1.5 ORGANIZAÇÃO DA DISSERTAÇÃO 18

2 TRABALHOS RELACIONADOS 19
2.1 CLASSIFICAÇÃO DE AMBIENTES E CONSTRUÇÃO DE MAPAS POR GRADE SEMÂNTICA, BASEADOS EM DADOS DE *LASER* 19
2.2 MAPEAMENTO SEMÂNTICO USANDO ROBÔS MÓVEIS 20
2.3 EXPLORAÇÃO SEMÂNTICA ONLINE DE MAPAS INTERNOS 20
2.4 MAPEAMENTO SEMÂNTICO DE UM ROBÔ ATRAVÉS DO RECONHECIMENTO DE ATIVIDADES HUMANA . . . 21
2.5 MAPAS SEMÂNTICOS DE OBJETOS PARA ATIVIDADES DOMÉSTICAS ROBOTIZADAS – REPRESENTAÇÃO, AQUISIÇÃO E USO. 22

3 FUNDAMENTAÇÃO TEÓRICA 23
3.1 RECONHECIMENTO DE OBJETOS 23
3.1.1 Reconhecimento de padrões 23
3.2 ONTOLOGIAS 25
3.3 MAPEAMENTO 26
3.3.1 Mapeamento Métrico 26
3.3.2 Mapeamento Topológico 27
3.3.3 Mapas Métricos vs. Mapas Topológicos 28
3.4 LOCALIZAÇÃO RELATIVA 29
3.5 PLANEJAMENTO DE CAMINHOS 31
3.5.1 Decomposição Celular 31
3.5.1.1 Decomposição celular exata 32
3.5.1.2 Decomposição celular aproximada 33
3.5.2 Campo Potencial 34
3.6 ARQUITETURA DE ROBÔS MÓVEIS 35
3.6.1 Paradigmas Robóticos 35

3.6.1.1 Paradigma hierárquico 36
3.6.1.2 Paradigma reativo 36
3.6.1.3 Paradigma híbrido 37
3.6.2 Arquitetura Híbrida DAMN 37

4 SISTEMA DE NAVEGAÇÃO BASEADO EM COMPORTAMENTOS 39
4.1 PRINCÍPIO DE FUNCIONAMENTO 39
4.2 BLOCO BANCO DE ONTOLOGIAS 41
4.3 BLOCO BANCO DE OBJETOS 42
4.4 BLOCO MAPA DO AMBIENTE 42
4.5 BLOCO SENSORES 43
4.6 BLOCO ATUADORES 43
4.7 BLOCO COMPORTAMENTOS REATIVOS 44
4.8 BLOCO COMPORTAMENTOS DELIBERATIVOS 44
4.9 BLOCO GERENCIADOR DE COMPORTAMENTOS 46

5 EXPERIMENTOS 47
5.1 AMBIENTE DE TESTES 47
5.2 EXPERIMENTOS DO RECONHECIMENTO DE OBJETOS 49
5.3 EXPERIMENTOS DOS COMPORTAMENTOS REATIVOS . 50
5.4 EXPERIMENTOS DOS COMPORTAMENTOS DELIBERATIVOS 51
5.4.1 Objetos estáticos 51
5.4.2 Objetos dinâmicos 53
5.5 EXPERIMENTO UTILIZANDO COMPORTAMENTOS REATIVOS E DELIBERATIVOS 55

6 CONCLUSÃO E TRABALHOS FUTUROS 58

REFERÊNCIAS 60

7 ONTOLOGIA DO SISTEMA 62

1 INTRODUÇÃO

A capacidade do ser humano de navegar em ambientes conhecidos ou desconhecidos é vista, em muitos casos, como intuitiva ou racional. Intuitiva no sentido de ser realizada de forma espontânea, causada principalmente pela rotina diária, como realizar o percurso de casa para o trabalho ou simplesmente o ato de deixar as crianças no colégio. E racional, quando um maior esforço mental é necessário, em atividades executadas pela primeira vez.

No contexto dos robôs móveis, a ”inteligência” é organizada através de paradigmas que podem ser descritos segundo três primitivas: Sentir, Planejar e Agir(MURPHY, 2000). De acordo com o relacionamento entre essas primitivas, é possível estabelecer um comportamento "inteligente"para um robô móvel dentro do contexto da navegação autônoma, ou seja, a navegação sem a interferência de seres humanos. Com isso, é permitido ao robô sair de uma posição inicial até uma posição final, desviando-se de obstáculos e percorrendo sua trajetória de forma segura, estando preparado para contornar situações inesperadas que possam impedir o cumprimento de sua missão.

Figura 1 – Etapas de um processo de navegação robótico.

Fonte: Santana (2011)

O problema da navegação aplicado à robótica móvel pode ser dividido em cinco subproblemas menores: mapeamento de ambientes, localização, planejamento de caminho, geração de trajetória e execução de trajetória (SANTANA, 2011). A Figura 1 mostra a sequência de etapas que constitui o processo de navegação robótico.

- Mapeamento de Ambiente

  Possibilita a criação de modelos computacionais do ambiente, utilizando informações sensoriais captadas pelo robô, que podem ser representadas em forma de mapa.

- Localização

  Consiste na capacidade do robô determinar sua pose (posição e orientação) no seu espaço de trabalho, utilizando apenas informações sensoriais e algum conhecimento acumulado previamente do ambiente.

- Planejamento de Caminho

Consiste em algoritmos que utilizam o mapa do ambiente, para calcular o melhor caminho entre dois pontos distintos, representados pela posição inicial e final do robô.

- Geração de Trajetória

  Possibilita adaptar o percurso calculado pelo planejador de caminhos, para uma forma, que respeite as restrições cinemáticas e temporais do robô.

- Execução da Trajetória

  Garante que os atuadores do robô (ex: motores) executem a trajetória gerada de forma mais fiel possível, respeitando as restrições cinemáticas e temporais impostas.

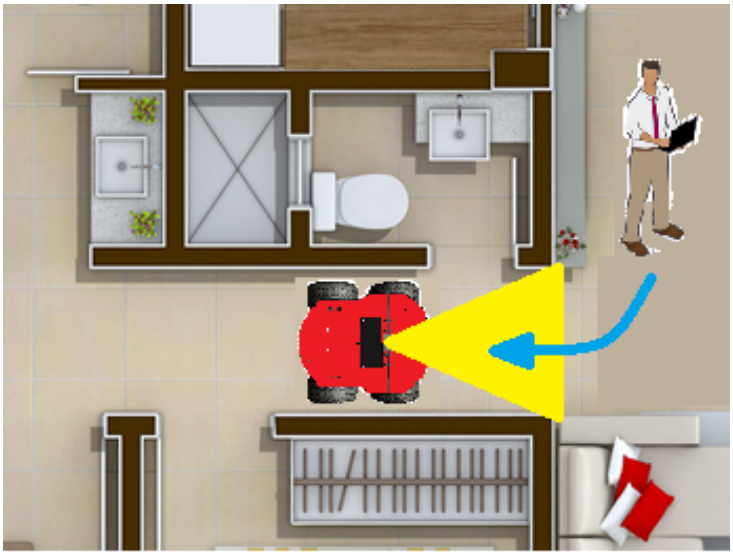
1.1 MOTIVAÇÃO E JUSTIFICATIVA

O processo de navegação, no contexto de robôs móveis, é sem dúvida um dos maiores desafios relacionado ao campo da robótica. Dessa forma, tornar um robô independente, capaz de se locomover por diferentes tipos de ambientes e interagir com obstáculos estáticos e dinâmicos, seria a função de um sistema de navegação considerado ideal para ser utilizado em robôs móveis.

A navegação em ambientes estáticos é a forma mais simples para um robô conseguir atingir uma localização específica. Quando um ambiente é estático, significa que os objetos não mudam de posição com o passar do tempo. O robô segue apenas as medições presentes em um mapa, sem se preocupar com o surgimento de obstáculos imprevistos, que possam aparecer em sua trajetória.

Quando se trata de colocar um robô em um ambiente desconhecido e dinâmico, sujeito a inúmeras situações inesperadas, o processo de navegação tem que levar em consideração diversos fatores ao se planejar as ações a serem executadas. Inicialmente, objetos dinâmicos podem se locomover a diferentes velocidades, o que pode dificultar a previsão do tempo de reação entre o robô receber um estímulo proveniente do ambiente e agir adequadamente a ele. Em alguns casos, objetos dinâmicos podem se passar por objetos estáticos. Por exemplo, uma pessoa parada no meio de um corredor. Para um robô que planeja sua navegação baseando-se apenas em informações sensoriais de profundidade (ou distância), ele vai considerar uma pessoa que está parada, como um objeto estático (sem ocorrência de movimento). Consequentemente, o robô não tomará nenhuma precaução ao passar próximo dele, e assim, estará sujeito à ocorrência de uma possível colisão, caso a pessoa se mova de forma inesperada.

Figura 2 – Predição de movimentos aplicado à navegação de robôs.



Fonte: Própria autoria.

A Figura 2 mostra como a predição de movimentos pode ajudar a resolver alguns problemas causados pela limitação dos sensores. Inicialmente, o robô está inserido em um ambiente dinâmico, formado por diversas subdivisões. O triângulo amarelo indica a região de alcance do sensor, ou seja, o campo de visão do robô. Tudo que estiver fora dessa região e que não foi mapeada previamente fica invisível para ele. Nesse cenário, uma pessoa que está vindo de encontro ao robô, por uma trajetória representada pela seta azul, pode não perceber e colidir frontalmente com ele.

Essa situação poderia ser evitada se o sistema de navegação do robô identificasse que aquele é um local por onde passam pessoas constantemente, podendo ocasionar um risco colisão. Dessa forma, o robô pode adotar um comportamento de navegação como soar um alarme, diminuir a velocidade ou usar outro meio qualquer, que possa alertar sua presença.

Diante dos diversos fatores que dificultam a navegação de robôs móveis em ambientes dinâmicos, verificou-se a necessidade de dotar um robô com comportamentos "inteligentes", que possibilite que sua navegação seja adaptada, de acordo com os objetos presente no ambiente. Dessa forma, esse trabalho buscou desenvolver um sistema de navegação inteligente, que utiliza o reconhecimento de objetos para gerar comportamentos que estejam relacionados com as características dos objetos reconhecidos.

1.2 OBJETIVO GERAL

O objetivo geral desse trabalho consiste no desenvolvimento de um sistema de navegação inteligente para robôs móveis, que utiliza o reconhecimento de objetos, para gerar comportamentos que se adaptam ao tipo de objeto reconhecido. Entre alguns comportamentos que irão auxiliar o robô no processo de navegação por ambientes dinâmicos e desconhecidos, pode-se citar, a capacidade de diminuir sua velocidade e manter uma distância segura, dependendo do tipo de objeto reconhecido.

1.3 OBJETIVOS ESPECÍFICOS

- Desenvolvimento de um sistema de controle de movimento para um robô.
- Realização do mapeamento e do reconhecimento de objetos em tempo real.
- Desenvolvimento da arquitetura do sistema de navegação, baseando-se no paradigma híbrido (que visa a união de reações instantâneas com reações mais elaboradas).
- Utilização de técnicas semânticas na classificação de um objeto reconhecido.

1.4 CONTRIBUIÇÕES

A principal contribuição deste trabalho está relacionada ao desenvolvimento de um sistema de navegação para robôs móveis, que utiliza o reconhecimento de objetos como forma de auxiliar o robô no processo de tomada de decisão, criando comportamentos capazes de adaptar a navegação do robô ao tipo de objeto reconhecido. Na maioria das abordagens que tratam o problema da navegação, o robô somente altera sua navegação, caso encontre algum obstáculo que o force a desviar de sua trajetória. Além disso, geralmente consideram que os objetos são sempre estáticos. No trabalho proposto, o robô assumirá, durante o processo de navegação, certos comportamentos que se adequam aos objetos reconhecidos, sejam eles estáticos ou dinâmicos.

Esse trabalho busca tornar mais robusta a navegação de robôs móveis, utilizando o reconhecimento de objetos como forma de planejar seus comportamentos. Além disso,

essa abordagem de navegação baseada no reconhecimento de objetos pode ser utilizada como base para qualquer tipo de robô.

1.5 ORGANIZAÇÃO DA DISSERTAÇÃO

- O capítulo 2 aborda as pesquisas e trabalhos relacionados ao tema proposto.
- O capítulo 3 aborda a fundamentação teórica necessária para a realização desse trabalho, sendo abordados temas como localização, mapeamento, planejamento de caminhos, paradigmas robóticos, reconhecimento de objetos e ontologias.
- O capítulo 4 irá abordar o sistema de navegação desenvolvido.
- O capítulo 5 trata dos experimentos realizados.
- O capítulo 6 finaliza com as conclusões e trabalhos futuros.

2 TRABALHOS RELACIONADOS

Este capítulo aborda alguns trabalhos relacionados ao tema proposto, mostrando suas principais características e a relevância sobre a problemática estudada nesse trabalho.

2.1 CLASSIFICAÇÃO DE AMBIENTES E CONSTRUÇÃO DE MAPAS POR GRADE SEMÂNTICA, BASEADOS EM DADOS DE LASER

Neste trabalho, Lei, Kodagoda e Gamini (2010) tratam da construção de um *Semantic Grid Map* (SGM), ou mapa semântico baseado em grade, para representar um determinado ambiente. O SGM é similar a um mapa em grade de ocupação, com a exceção de possuir alto nível de informação sobre o ambiente (LEI; KODAGODA; GAMINI, 2010). Para a construção do mapa foi necessário capturar as principais características do ambiente através de sensores *lasers* e aplicá-las a um algoritmo de evolução diferencial (R.N; A.; A., 2008), para restringir a seleção do número de características a serem tratadas. Como resultado, os autores selecionaram quatro características consideradas dominantes quanto à performance e requisito computacional.

O processo de construção do SGM, começa com o robô explorando um ambiente desconhecido através de algum algoritmo de mapeamento, com o objetivo de criar um mapa em grade de ocupação. Com o mapa em mãos, aplica-se uma exploração virtual (LEI; KODAGODA; GAMINI, 2010) visitando todas as possíveis localizações do robô enquanto obtém as informações sensoriais. A partir da junção do mapa em grade de ocupação (SOUZA, 2008) com os dados sensoriais, consegue-se classificar o ambiente onde o robô está inserido e, consequentemente, gerar o mapa semântico. Os autores utilizam algumas heurísticas específicas para a classificação de objetos, como portas, corredores, salas de aula e escritório.

A proposta desse trabalho é bem interessante, por utilizar a classificação dos objetos e seu significado semântico, para criar mapas semânticos do ambiente. Utilizando essa abordagem, um robô poderia planejar suas ações e comportamentos de acordo com os objetos e ambiente onde ele se encontra, sabendo que o robô conhece a posição dos objetos e algumas informações semânticas deles.

2.2 MAPEAMENTO SEMÂNTICO USANDO ROBÔS MÓVEIS

Um conceito um tanto recente no campo do mapeamento robótico é o mapeamento semântico, onde informações semânticas podem ser usadas para o planejamento de caminhos e a tomada de decisões (WOLF; SUKHATME, 2008). Esse tipo de mapeamento considera, além das características métricas do ambiente, também informações semânticas sobre os dados. Nesse trabalho, Wolf e Sukhatme (2008) desenvolveram técnicas para construção de mapas de atividade e navegabilidade em ambientes, combinando técnicas de aprendizagem de máquina com algoritmos padrões de mapeamento. Eles apresentam sua proposta de mapeamento semântico, relacionando-a a dois problemas específicos: o mapeamento de terreno e o mapeamento baseado em atividade.

O mapeamento de terreno consiste em capturar dados tridimensionais da superfície através de *lasers* e construir um mapa com três dimensões do ambiente. Já o mapeamento baseado em atividade consiste em mapear o ambiente analisando o uso do espaço por entidades dinâmicas em função do tempo.

Esses dois problemas abordados com o mapeamento semântico foram tratados com base em dois modelos matemáticos: *Hidden Markov Model* (HMM) (RABINER, 1989) e o *Support Vector Machine* (SVM) (BURGES, 1998). A função básica desses dois modelos de probabilidade matemática é classificar os dados providos pelos sensores em classes específicas que representem padrões distintos. No mapeamento de terreno, o objetivo do HMM e do SVM é classificar as regiões do mapa em navegáveis e não navegáveis. Já no mapeamento baseado em atividades, o objetivo dos modelos matemáticos é classificar espaços de um ambiente urbano onde passam pessoas, carros, bicicletas e outros objetos móveis, em dois tipos específicos: ruas e calçadas.

Como resultado, os autores verificaram que ambos os modelos matemáticos se mantiveram equilibrados com valores eficazes de classificação. Sendo assim, pode-se concluir que o mapeamento semântico tem tudo para tornar a navegação de robôs móveis ainda mais robusta em atividades complexas, oferecendo informações de raciocínio que se assemelham muito aos seres humanos.

2.3 EXPLORAÇÃO SEMÂNTICA ONLINE DE MAPAS INTERNOS

Neste trabalho (LIU; CHEN; WICHERT, 2012), os autores propuseram um método que extrai características abstratas do ambiente, usando a técnica de raciocínio Bayesiano. Essas características representam conceitos abstratos que não aparecem nos mapas métricos tradicionais. Normalmente, em uma abordagem métrica, o mapa de um

ambiente consiste de espaços vazios, por onde o robô pode navegar e espaços ocupados como obstáculos e paredes. Através da exploração semântica, mais informações são incorporadas ao mapa para facilitar a localização do robô. Entre algumas dessas informações estão a classificação do ambiente como sala, cozinha, quarto e dos objetos dentro deles. Outra informação é a relação entre esses ambientes por exemplo, a cozinha fica no final de um corredor; as paredes dos quartos formam o corredor que vai para cozinha e a sala fica no inicio do corredor. Conclui-se que estando na sala, a única forma de ir para a cozinha é passando pelo corredor.

Em seu trabalho Liu, Chen e Wichert (2012) utilizam inicialmente um mapa em grade de ocupação pronto para o robô iniciar o processo de exploração semântico. O objetivo desse processo é fazer o robô conseguir construir um mapa abstrato do ambiente, que pode ser acoplado ao mapa em grade de ocupação, tornando mais robusta a etapa de navegação. Como resultado, eles conseguiram construir um mapa semântico classificando um ambiente composto por diversos quartos conectados por portas.

A abordagem dos autores é bem similar à proposta apresentada nesse trabalho. Principalmente, pelo fato deles levarem em consideração o conceito semântico das entidades dentro do ambiente.

2.4 MAPEAMENTO SEMÂNTICO DE UM ROBÔ ATRAVÉS DO RECONHECIMENTO DE ATIVIDADES HUMANA

Neste trabalho (LI et al., 2012), os autores propõem uma nova abordagem para construção de mapas semânticos, onde informações semânticas são inferidas a partir da interação do ser humano com os objetos presentes no ambiente. Para isso eles utilizam um robô dotado de vários sensores com a capacidade de se localizar e, ao mesmo tempo, mapear o ambiente por meio de um algoritmo de *Simultaneous Localization and Mapping* (SLAM). Como resultado obteve-se um mapa métrico interno do ambiente com duas dimensões.

A inserção dos dados semânticos no mapa métrico é resultado do mapeamento das atividades de uma pessoa, rastreadas por meio de dois sensores de movimento, colocados na cintura e na coxa. Com os movimentos do corpo humano interagindo com os objetos no ambiente, o robô consegue identificar padrões e relacionar determinadas atividades executadas pela pessoa a objetos específicos, atualizando, dessa forma, o mapa métrico com informações semânticas sobre o objeto.

O alto custo computacional relacionado ao processo de reconhecer um objeto por meio de técnicas de reconhecimento de padrões levou a construção de mapas semânticos através da análise das atividades de interação humano-ambiente (LI et al.,

2012). Portanto, diante do exposto pelos autores, fica a preocupação em analisar técnicas de reconhecimento de objetos que ofereçam bons resultados, levando em consideração sua eficácia e seu desempenho computacional.

2.5 MAPAS SEMÂNTICOS DE OBJETOS PARA ATIVIDADES DOMÉSTICAS ROBOTIZADAS – REPRESENTAÇÃO, AQUISIÇÃO E USO.

Nessa pesquisa (PANGERCIC et al., 2012), propuseram a criação de mapas semânticos de objetos levando em consideração a geometria tridimensional, a posição e a orientação deles. O ambiente abordado no trabalho consistiu em representar todas as entidades domésticas de uma cozinha, como: armário, dispositivos elétricos, mesas e balcões, assim como suas relações. A representação do mapa semântico como base de conhecimento simbólico foi construída utilizando o conceito de ontologias, o que permitiu a criação de hierarquias de conceitos e relações entre os diferentes tipos de objetos (PANGERCIC et al., 2012). Através das ontologias consegue-se, por exemplo, modelar a relação entre uma fechadura e uma porta, ou entre uma gaveta e um armário.

Com a construção do mapa semântico, o próximo passo foi identificar os objetos no contexto da cozinha. Para isso, os autores utilizaram o kinect para capturar uma nuvem de pontos tridimensional e aplicaram algumas técnicas para melhorar a qualidade da reconstrução da superfície dos objetos. A metodologia utilizada por eles consegue reduzir significativamente o volume de dados ocupado por uma nuvem de pontos. Com o processo de reconhecimento, eles conseguem associar o objeto a seu significado, possibilitando, por exemplo, reconhecer que um armário tem portas e que elas podem ser abertas ou fechadas; ou ainda que elas se abrem e se fecham de uma determinada forma. Para validar a proposta, em termos de detecção dos objetos e abertura dos armários, os autores utilizaram cinco cozinhas, onde o robô teria que detectar e abrir todos os 22 armários, três vezes. Como resultado, o teste mostrou que tanto na detecção quanto na abertura dos armários, a taxa de sucesso ficou maior ou igual a 85%.

Esse trabalho tem muito a oferecer para o desenvolvimento dessa dissertação. Primeiramente, porque eles utilizam o reconhecimento de objetos como base para a construção do mapa semântico e em segundo lugar, porque eles modelam a base de conhecimentos através de ontologias.

3 FUNDAMENTAÇÃO TEÓRICA

Para que um robô possa ser considerado autônomo é necessário que ele possua um sistema de navegação "inteligente". Para isso, é necessário que o robô seja capaz de mapear objetos e espaços no ambiente, deve ser capaz de se auto localizar e estar habilitado a planejar e decidir qual o melhor caminho a seguir, desviando-se de obstáculos e por fim, deve executar o caminho planejado.

3.1 RECONHECIMENTO DE OBJETOS

Reconhecer ou classificar um objeto é algo que o ser humano assim como outros animais realizam naturalmente. Determinadas coisas do dia a dia podem ser reconhecidas naturalmente como: reconhecer parentes, amigos e vizinhos. Enquanto outras precisam de um pouco mais de esforço, como diferenciar dois irmãos gêmeos ou rever um amigo de infância. Reconhecer objetos envolve descobrir características descriminantes que os descreva e utilizar tais descrições para compará-los com um conceito ideal (GRACIANO, 2012). Quando se conhece o conceito ideal de um objeto, significa que ele possui características que são únicas e que o caracteriza como pertencente a uma determinada classe. Isso explica, porquê se consegue classificar um objeto como sendo uma cadeira ou um cachorro, mesmo que nunca os tenha visto.

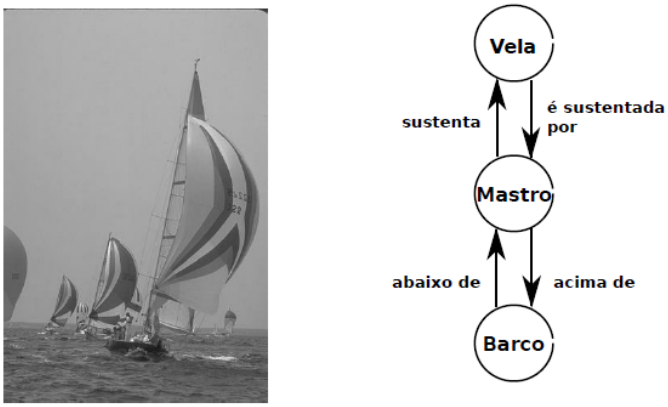
3.1.1 Reconhecimento de padrões

O reconhecimento de padrões pode ser definido como uma disciplina cientifica que objetiva a classificação de objetos em um número de categorias ou classes (THEODORIDIS; KOUTROUMBAS, 2009). Esses objetos podem ser ondas sonoras, animais, ações de valores ou qualquer outro tipo de medição que necessite ser classificada. Um dos meios mais comuns utilizados no reconhecimento e classificação de objetos são as imagens capturadas por sistemas de câmeras. Por meio da análise de imagens, consegue-se extrair informações que na maioria das vezes só o sistema visual humano consegue realizar. Com base em algoritmos de visão computacional e técnicas de processamento de imagens, aplicações conseguem identificar faces, placas de carros, reconhecer assinaturas digitais, obter a profundidade dos objetos entre outras inúmeras funcionalidades.

Existem diversas abordagens para se desenvolver um sistema de reconhecimento de objetos. Entre as principais podem-se citar as redes neurais (BISNETO, 2011), os descritores locais (MAIA, 2010) e o reconhecimento baseado em contexto (COZMAN, 2012). No trabalho de Graciano (2012), a autora considera três paradigmas de onde se originariam as principais técnicas de reconhecimento de objetos: o paradigma estatístico, o estruturado e o híbrido.

O paradigma estatístico é um dos mais antigos aplicados ao reconhecimento de objetos. Consiste em determinar um conjunto de características representadas em um espaço multidimensional e verificar se tais características estão presentes num determinado objeto pertencente a esse mesmo espaço multidimensional. Já o paradigma estruturado utiliza uma abordagem que representa um objeto por meio de seus elementos primitivos na forma de um grafo, permitindo que um objeto seja descrito não apenas por sua forma física, mais também pelas relações que seus elementos primitivos compartilham (GRACIANO, 2012). A Figura 3 mostra como o paradigma estruturado é utilizado no reconhecimento de um veleiro.

Figura 3 – Representação estrutural de um veleiro.



Fonte: Graciano (2012).

O veleiro estar sendo representado por um grafo orientado, onde cada vértice indica um elemento primitivo que o compõe e as arestas as relações entre esses elementos. Por exemplo, o elemento "Vela" mantém relações somente com o elemento "Mastro" do tipo "é sustentada por" e "sustenta". Esse paradigma permite que em muitos casos, o reconhecimento possa ser feito apenas utilizando as relações entre as primitivas, quando, por exemplo, não se tem acesso a forma física completa do objeto.

Quanto ao paradigma híbrido, ele consiste da união do paradigma estatístico com o estruturado, aproveitando o melhor de cada uma das abordagens.

Para a realização esse trabalho foi escolhido o paradigma estatístico, por ser amplamente utilizado na implementação de técnicas de reconhecimento de padrões aplicadas a imagens bidimensionais.

3.2 ONTOLOGIAS

Uma das formas de descrever o conhecimento adquirido a respeito de uma entidade, objeto ou agente, é através do uso de ontologias.

A palavra ontologia tem diferentes significados com base nas diferentes áreas do conhecimento. A diferença mais radical fica entre o conceito filosófico que marca a origem do termo ontologia e seu sentido no campo computacional que emergiu nos anos atuais. No primeiro caso, a palavra Ontologia (com inicial maiúscula) se refere a uma disciplina filosófica relacionando a natureza a estrutura da realidade. Aristóteles definia o conceito de Ontologia como a ciência do ser enquanto ser, focando na natureza e na estrutura das coisas, independente de sua existência atual (STAAB; STUDER, 2010). O segundo caso se refere a área de ciência da computação, onde a palavra ontologia (com inicial minúscula) representa uma forma de modelar formalmente a estrutura de um sistema computacional, através de observações úteis para serem utilizadas no sistema (STAAB; STUDER, 2010).

Uma ontologia pode ser definida como uma especificação formal de um conhecimento compartilhado relacionado a algum domínio de interesse (USCHOLD; GRUNINGER, 1996).

O ato de compartilhar um conhecimento, significa algo de senso comum que muitas pessoas concordam, e que deve está disponível para ser usado por outras entidades, que necessitem agrega-lo como parte na construção de um conhecimento mais específico ou mais geral. Esse conhecimento tem que ser formalizado de tal forma que evite ambiguidades na sua construção, ou seja, independente de onde ele seja aplicado seu significado deve ser exatamente o mesmo.

Uma forma de representar uma ontologia é através de um conjunto de classes e relações, que podem ser organizadas de forma hierárquica e que possibilitam a generalização ou especialização de um conceito.

As ontologias permitem identificar características semânticas envolvidas num contexto onde participam diferentes entidades e que mantêm diferentes relações entre si. Esse fato contribuiu para sua utilização nesse trabalho, pois através dos objetos reconhecidos será possível formalizar seus conceitos e relações existentes, resultando na modificação do comportamento de navegação do robô.

3.3 MAPEAMENTO

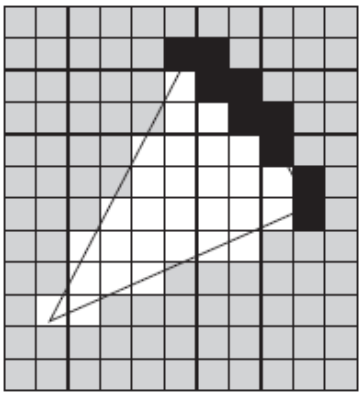
O problema do mapeamento é geralmente considerado um dos problemas mais importantes na busca de se construir robôs móveis autônomos. O mapeamento trata de adquirir modelos espaciais do ambiente físico através dos sensores dos robôs (THRUN, 2003). Historicamente, o campo de mapeamento robótico tem sido dividido em duas abordagens: o mapeamento métrico e o mapeamento topológico.

3.3.1 Mapeamento Métrico

Esse tipo de mapeamento consiste em algoritmos que conseguem capturar as propriedades métricas do ambiente (THRUN, 2003). Geralmente, essas informações são capturadas por meio de câmeras, sonares ou *lasers* e geralmente representadas por meio de coordenadas cartesianas.

Um dos primeiros algoritmos a utilizar essa abordagem foi o mapeamento baseado em grade de ocupação (THRUN, 2003). Esse algoritmo ainda é bastante utilizado nas aplicações atuais (SOUZA et al., 2013), (SANTANA, 2011). Seu funcionamento consiste em construir um mapa através de uma matriz de células, que podem estar ocupadas, vazias ou ainda não terem sido mapeadas (SOUZA, 2012). Na Figura 4, as células ocupadas (cor preta) indicam locais onde o robô não pode trafegar e as células vazias (cor branca) indicam lugares possíveis de navegação. As células restantes (cor cinza) são regiões ainda não mapeadas que podem assumir um dos estados citados anteriormente.

Figura 4 – Mapeamento por sonar utilizando uma grade de ocupação.



Fonte: Souza (2012).

A fórmula probabilística padrão usada para representar o mapeamento com

grade de ocupação é dada pela expressão (3.1) (SANTANA, 2011).

$$P(M|Z_{1:t}) \tag{3.1}$$

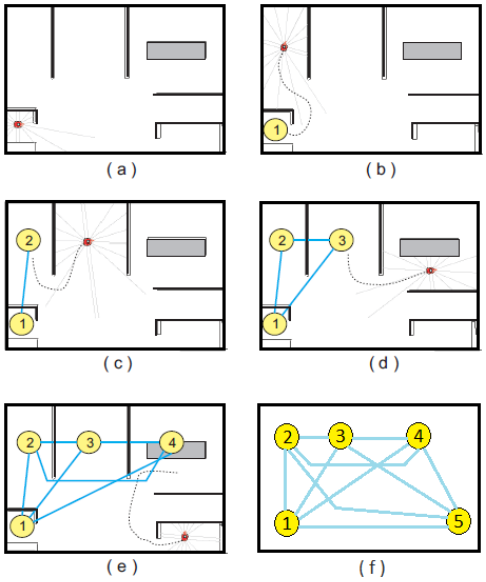
Onde M representa o mapa adquirido, ou seja, a grade de ocupação, criada pela discretização das dimensões do ambiente real e atualizada por $Z_{1:t}$, o conjunto de medições sensoriais obtidas até o instante de tempo t. O mapa M pode ser definido como um conjunto de células $m_{(x,y)}$ quando a grade é bidimensional ou $m_{(x,y,z)}$ quando tridimensional. Cada célula só pode assumir valores entre o intervalo 0 e 1. Uma célula com valor 0 significa que ela está vazia e com valor 1, ocupada.

Para usar essa formula é necessário que a posição e orientação do robô sejam conhecidas, pois todo o calculo está relacionado a distância entre a pose do robô e os dados captados pelos sensores.

3.3.2 Mapeamento Topológico

O mapeamento topológico surgiu como uma alternativa diferente do mapeamento métrico. Mapas topológicos representam ambientes como uma lista de lugares significativos conectados por arestas (THRUN, 2003). Um mapa topológico é representado na forma de um grafo, onde os lugares visitados são marcados como vértices e os caminhos possíveis para se chegar nesses lugares são marcados como arestas. Nesse tipo de mapeamento, o robô não possui uma informação precisa sobre sua pose no ambiente, apenas ele consegue identificar em qual parte do ambiente ele se encontra.

Figura 5 – Mapeamento Topológico.



Fonte: Adaptado de Souza (2012).

A Figura 5 mostra o processo de mapeamento topológico realizado por um robô. Inicialmente na Figura 5(a), o robô não possui nenhuma informação da estrutura do ambiente. Então, ele inicia o mapeamento, colocando sua posição atual como o primeiro local do ambiente, criando seu primeiro vértice no mapa. Na figura 5(b), o robô mapeou um primeiro compartimento e seguiu na tentativa de mapear os demais ainda não explorados. Na Figura 5(c), dois lugares já foram mapeados, criando respectivamente dois vértices, numerados de acordo com a ordem que o robô visitou. Esse processo continua até que o robô tenha visitado todos os lugares que compõe o ambiente (dependendo da estratégia de exploração utilizada). No final, tem-se um grafo composto por um conjunto de vértices e arestas, representado pela Figura 5(f) e que pode ser utilizado como mapa para uma navegação futura.

Mapas topológicos são mais eficientes por armazenar pouca informação com relação ao ambiente. Um robô que utiliza esse tipo de mapa não precisa saber exatamente sua pose em coordenadas geométricas, ele sabe apenas que está num dado local daquele ambiente. Do ponto de vista de tempo de processamento, os mapas topológicos são mais interessantes, pois armazenam menos informação, quando comparado aos mapas métricos. Por outro lado, eles perdem em precisão quando se deseja uma localização mais precisa.

3.3.3 Mapas Métricos vs. Mapas Topológicos

Cada tipo de mapeamento possui suas vantagens e desvantagens, fica a critério da aplicação, a abordagem a ser utilizada. No trabalho de Souza (2008), o autor relaciona respectivamente as principais vantagens e desvantagens do mapeamento métrico e topológico nas Tabelas 1 e 2.

Tabela 1 – Vantagens e desvantagens da abordagem métrica

Vantagens	Fácil de construir, representar e manter. Reconhecimento de lugares (baseado na geometria). Facilita a computação de caminhos curtos.
Desvantagens	Planejamento ineficiente, alto custo computacional. Requer a determinação precisa da pose do robô. Linguagem simbólica pobre para a solução de problemas.

Fonte: Souza (2008).

Tabela 2 – Vantagens e desvantagens da abordagem topológica

Vantagens	Permite planejamento eficiente e baixo custo computacional Não requer a determinação precisa da localização do robô Representação simbólica conveniente para a solução de problemas
Desvantagens	Dificuldade de construir e manter para ambientes grandes Dificuldade no reconhecimento de lugares Pode produzir caminhos sub ótimos.

Fonte: Souza (2008).

Para esse trabalho foi utilizada a abordagem métrica, mais especificamente o mapeamento baseado em grade de ocupação. Entre alguns motivos que levaram a sua escolha pode-se citar sua facilidade de representação e manipulação, além de poder oferecer informações geométricas do ambiente, importantes para o planejamento de caminhos e tomadas de decisões.

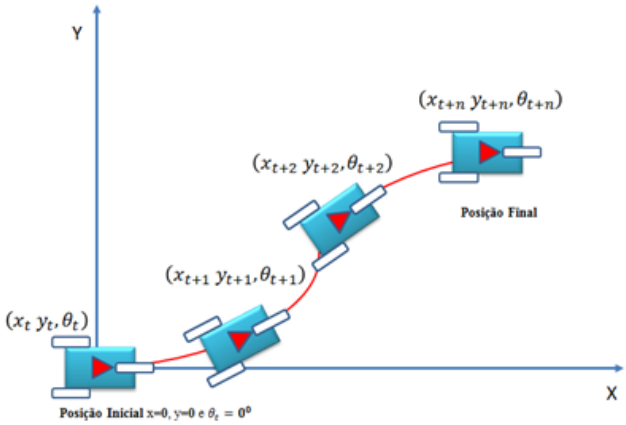
3.4 LOCALIZAÇÃO RELATIVA

Para se construir um sistema de navegação "inteligente" para um robô móvel é essencial saber sua localização física no ambiente. A localização de um robô móvel consiste no problema de determinar sua pose relativa a um dado referencial do ambiente (THRUN; BURGARD; FOX, 2005). A pose é formada pelas coordenadas x, y (e z para robôs aéreos e subaquáticos) e também por sua orientação em relação aos eixos do referencial assumido no ambiente.

Um dos métodos de localização utilizados para a navegação de robôs é a localização relativa. Esse tipo de localização também conhecida como *dead reckoning* é estimada baseando-se nas medidas das localizações dos instantes anteriores, onde o robô parte de uma posição inicial e à medida que ele se desloca, novas localizações irão surgindo com base nas localizações anteriores (SOUZA, 2008). Isso ocasiona um acumulo de erro à medida que o robô se desloca, não sendo um método muito eficiente para percorrer grandes distâncias, sem a utilização de algum método que modele tais erros.

A Figura 6 representa o método de localização relativa aplicado a um plano bidimensional. Nesse plano a pose do robô é representada por (x_t, y_t, θ_t), onde x_t e y_t são respectivamente as coordenadas em relação aos eixos X, Y do referencial adotado e θ_t a orientação do robô em relação ao eixo X. Partindo de uma posição inicial representada por sua pose $(x_t, y_t; \theta_t)$ até uma posição final $(x_{t+n}, y_{t+n}; \theta_{t+n})$, o robô calcula sua pose atual com base em sua pose anterior, levando em consideração o intervalo de tempo t entre uma pose e outra.

Figura 6 – Localização relativa.



Fonte: Próprio autor.

Esse tipo de localização geralmente utiliza *encoders* (sensores de rotação) que contam a quantidade de giros de cada roda, para que o robô consiga calcular a distância percorrida e sua orientação (SOUZA, 2008). Utilizando um modelo cinemático e as medições dos *encoders*, o robô consegue estimar sua pose no ambiente. Um modelo bastante utilizado é o modelo determinístico baseado em odometria (SIEGWART; NOURBAKHSH, 2004), representado pela equação 3.2 (SOUZA, 2012), (SIEGWART; NOURBAKHSH, 2004).

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} \Delta s \cos(\theta + \Delta\theta/2) \\ \Delta s \sin(\theta + \Delta\theta/2) \\ \Delta\theta \end{bmatrix} \tag{3.2}$$

$$\Delta s = \frac{\Delta s_r + \Delta s_l}{2} \tag{3.3}$$

$$\Delta\theta = \frac{\Delta s_r - \Delta s_l}{b} \tag{3.4}$$

$$\Delta s_l = \frac{2\pi.r.N_l}{N_{res}} \tag{3.5}$$

$$\Delta s_r = \frac{2\pi.r.N_r}{N_{res}} \tag{3.6}$$

Δs e $\Delta\theta$ representam, respectivamente, o deslocamento linear e o deslocamento angular realizado pelo robô. Esses deslocamentos dependem dos valores dos deslocamentos lineares de cada roda separadamente, sendo Δs_r para a roda direita e Δs_l para a roda esquerda, o b representa o tamanho do eixo entre rodas do robô. Esses deslocamentos independentes das rodas, por sua vez, são calculados em função da resolução máxima ou o número de pulsos correspondente a um giro completo da roda

N_{res}, dos pulsos retornados pelos *encoders* de ambas as rodas N_r e N_l e do raio das rodas r.

De posse de todos os valores citados acima, basta aplicá-los na equação 3.2 para se obter a pose atual do robô. Essa nova pose é resultado da soma dos valores da pose anterior com os valores das medições atuais.

A localização relativa foi escolhida nesse trabalho devido a sua simplicidade de implementação, mesmo estando ciente que a localização por odometria possui um acumulo de erro gradual à medida que o robô se desloca. Como o objetivo principal desse trabalho se trata da modelagem e implementação dos comportamentos adaptativos na navegação, a estimativa precisa da pose do robô não precisa ser muito rigorosa. Dessa forma, a modelagem e correção dos erros causados pela odometria, serão abordados em trabalhos futuros.

3.5 PLANEJAMENTO DE CAMINHOS

O primeiro passo para um sistema planejador de caminhos é transformar possíveis modelos de um ambiente continuo em um mapa discreto adequado para ser usado por um algoritmo de planejamento de caminhos (SIEGWART; NOURBAKHSH, 2004). Essa transformação é feita por decomposição. Segundo Siegwart e Nourbakhsh (2004), existem três estratégias básicas para se decompor um mapa baseado no modelo de um ambiente em um mapa discreto: *Road map* (mapas de rotas), decomposição celular e campo potencial. Nesse trabalho será abordado apenas os dois ultimos.

3.5.1 Decomposição Celular

Essa estratégia consiste em dividir o espaço livre por onde o robô consegue navegar em regiões chamadas células (SIEGWART; NOURBAKHSH, 2004), (OTTONI, 2000). Sobre essas células é construído um grafo de conectividade representando a relação de adjacência entre as mesmas. Com a construção desse grafo torna-se possível estabelecer um caminho entre duas configurações distintas do robô que ligue as posições inicial e final do percurso. Segundo Siegwart e Nourbakhsh (2004), o algoritmo de planejamento de caminhos para decomposição em células pode ser descrito através dos passos abaixo:

- Dividir o espaço livre em regiões simples chamadas células.

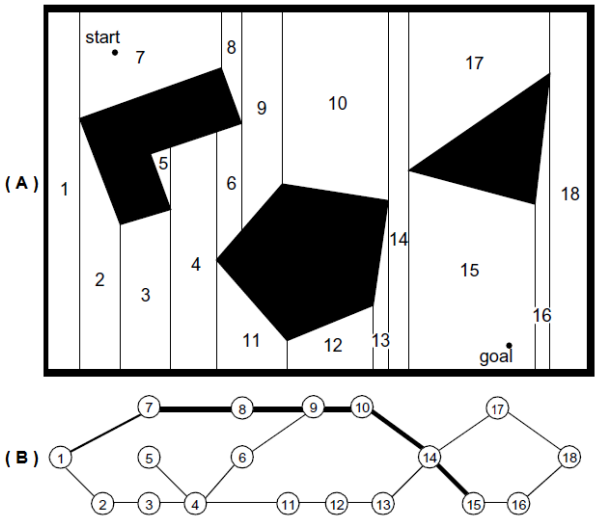
- Determinar quais células são adjacentes e construir um grafo de conectividade.
- Encontrar as células com as configurações inicial e final e depois encontrar um caminho no grafo de conectividade que ligue a célula com a configuração inicial a célula com a configuração final.
- Para a sequência de células encontradas por um algoritmo de busca apropriado, computar um caminho que passe pelo centro de cada célula, respeitando os seus limites.

A estratégia de decomposição celular pode ser implementada através de dois métodos (SIEGWART; NOURBAKHSH, 2004), (OTTONI, 2000): decomposição exata e decomposição aproximada.

3.5.1.1 Decomposição celular exata

Esse método de decomposição consiste em dividir o espaço livre do robô em células, de tal forma que a união delas represente fielmente as características do ambiente sem perda de informação. A Figura 7(a) mostra um mapa onde foi aplicado a decomposição celular exata em células na vertical. Já a Figura 7(b) representa o grafo de conectividade entre as células, especificando através de uma linha mais cheia, o caminho entre a configuração inicial e final do robô.

Figura 7 – Decomposição celular exata. (a) decomposição do ambiente em células; (b) grafo de conectividade construído a partir da divisão do ambiente.



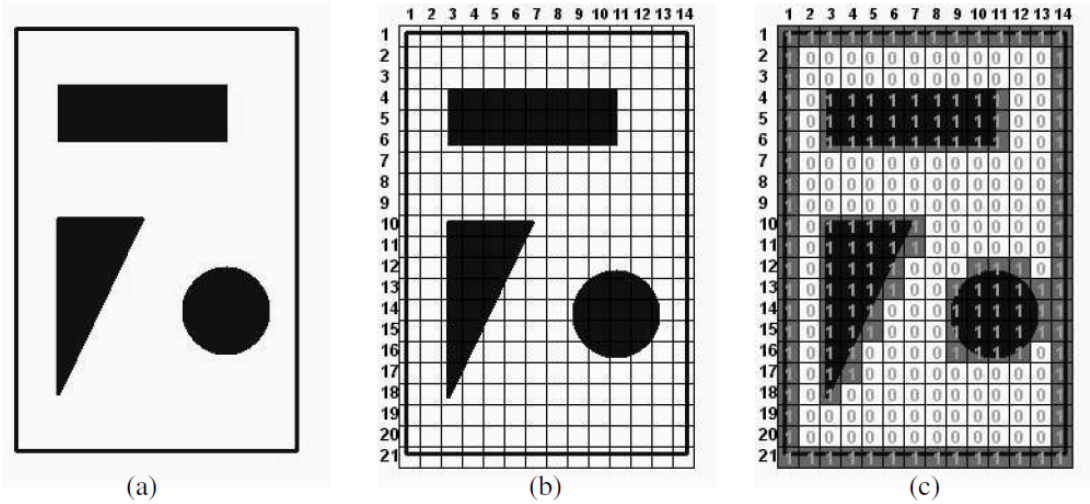
Fonte: Siegwart e Nourbakhsh (2004).

3.5.1.2 Decomposição celular aproximada

A decomposição por células aproximadas consiste em dividir o espaço livre do robô em células com a mesma forma geométrica, por exemplo, um quadrado ou cubo (quando o mapa é tridimensional). Esse método ao contrário da decomposição exata, geralmente perder informação, dependendo do tamanho das células e sua relação com a forma do ambiente. Isso acontece devido ao processo de discretização do espaço real em células com a mesma forma geométrica. A Figura 8 mostra as etapas para se realizar a decomposição celular aproximada. A Figura 8(a) mostra o mapa de um ambiente com três obstáculos. Esse mapa é discretizado em uma matriz de células de mesmo tamanho, todas com uma posição fixa na matriz, como mostrado na Figura 8(b).

Quando uma célula fica localizada numa região pertencente a um obstáculo, ela é considerada como ocupada, mesmo que não esteja totalmente preenchida. Na Figura 8(c), as células onde não haviam obstáculos foram pintadas com a cor branca, as células que estavam totalmente preenchidas com obstáculos foram pintadas de preto e as células que estavam apenas parcialmente preenchidas pelos obstáculos foram pintadas de cinza. Pela Figura 8(c) percebe-se que as células de cor cinza representam a imprecisão causada pela definição do tamanho das células. Uma forma de amenizar esse problema seria definir valores variáveis para o tamanho das células.

Figura 8 – Decomposição celular aproximada. (a) configuração do ambiente; (b) decomposição do ambiente em células; (c) aproximações realizadas pela discretização.



Fonte: Ottoni (2000)

A decomposição em células aproximadas será importante para a navegação baseada em comportamentos adaptativos, pois o sistema de controle do robô necessita saber sua pose no ambiente, principalmente em situações onde ele terá que sair de sua rota para desviar de algum obstáculo dinâmico e posteriormente ter que refazê-la.

3.5.2 Campo Potencial

Segundo Siegwart e Nourbakhsh (2004), o método de planejamento de caminhos denominado de campo potencial, considera o robô como um ponto sobre a influência de um campo potencial artificial. Essa abordagem considera a discretização do mapa do ambiente em uma grade de configurações, onde cada posição desta grade é associada a um valor de uma função que faz analogia a um campo potencial (OTTONI, 2000).

Essa abordagem considera que a posição da grade onde está a configuração final do robô, ou seja, o objetivo do robô possui um campo potencial atrativo e que as posições onde existe a presença de obstáculos possuem um campo potencial repulsivo. O robô busca atingir seu objetivo seguindo a direção dos campos potenciais.

Uma das principais características de se utilizar campos potenciais como mecanismo de planejamento de caminhos é a possibilidade de planejamento de trajetórias em espaços de configurações dinâmicos (SOUZA, 2008). Isso acontece porque, não é necessária a criação prévia de nenhuma estrutura de dados, na qual o robô vai utilizar para se guiar. Todo o trajeto é unicamente guiado pelos valores dos campos potenciais. A Figura 9 mostra como fica a direção do campo potencial com relação aos obstáculos e a configuração do alvo.

Figura 9 – Campo potencial.

Fonte: Souza (2008)

Neste trabalho, foi utilizado um planejamento de caminhos que utiliza os princípios da técnica de decomposição em células aproximadas e da técnica de campo de potencial. Foi utilizado um mapa em grade de ocupação, onde os espaços vazios (ou livres de obstáculos) foram ponderados de acordo com a função de navegação de Manhattan. Com isso, o robô deve seguir um caminho que o leve de um ponto inicial a um ponto objetivo, este com o menor potencial dentro do mapa.

3.6 ARQUITETURA DE ROBÔS MÓVEIS

Os diferentes tipos de arquiteturas para robôs móveis surgiram com base em algum tipo de paradigma robótico. As próximas seções irão definir quais os tipos de paradigmas existentes e qual foi utilizado nesse trabalho.

3.6.1 Paradigmas Robóticos

Um paradigma é uma filosofia ou conjunto de suposições ou técnicas que caracterizam uma abordagem sobre um problema (MURPHY, 2000). Os paradigmas surgiram como uma forma de concorrência, onde diferentes abordagens são utilizadas para se resolver um mesmo problema. Portanto, entender os paradigmas que regem a "inteligência" robótica é essencial para o desenvolvimento de aplicações "inteligentes".

Atualmente, a "inteligência" de um robô pode ser organizada através de três paradigmas: hierárquico ou também chamado de paradigma deliberativo, reativo e híbrido (MURPHY, 2000). Esses três paradigmas trabalham diretamente sobre as três primitivas básicas que regem a "inteligência" de um sistema robótico: sentir, planejar e atuar. A forma como essas primitivas foram organizadas possibilitou o surgimento dos paradigmas deliberativo, reativo e híbrido.

De forma geral, as três primitivas podem ser definidas como:

- Sentir

  Capacidade de sentir o ambiente a sua volta. Normalmente é feita através das leituras dos sensores.

- Planejar

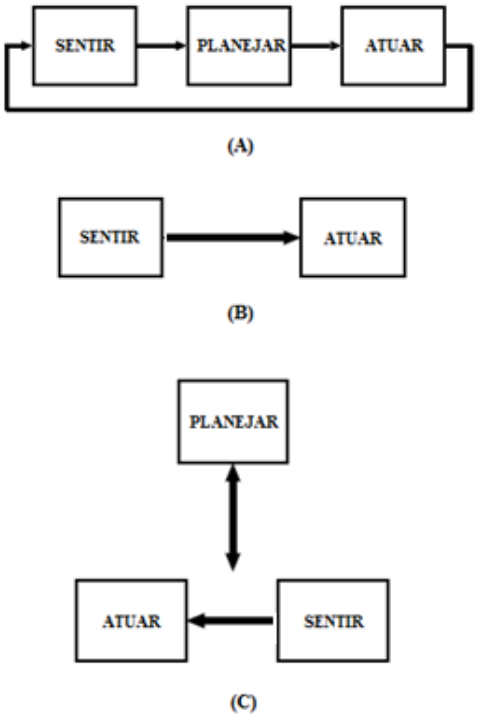
  Capacidade que o robô possui de processar dados obtidos pelos sensores ou de outras fontes de conhecimento para tomar decisões ou executar tarefas.

- Atuar

  Capacidade do robô interagir com o ambiente, enviando comandos diretos para os atuadores.

A Figura 10 representa os três paradigmas robóticos em termos das primitivas sentir, planejar e atuar.

Figura 10 – Paradigmas robóticos. (a) Paradigma hierárquico. (b) Paradigma reativo. (c) Paradigma Híbrido (hierárquico/reativo).



Fonte: Adaptado de Murphy (2000).

3.6.1.1 Paradigma hierárquico

O paradigma hierárquico surgiu por volta de 1967, sendo considerado o mais antigo entre os três. Como pode ser visto na Figura 10(a), o paradigma hierárquico utiliza uma abordagem *top-down* com forte tendência no planejamento. Isso acontece porque todo o processamento acontece sequencialmente e toda a parte decisória fica sobre a primitiva de planejamento. O paradigma hierárquico foi criado com base na forma como as pessoas pensam, normalmente, planejam uma ação e só depois a executam.

3.6.1.2 Paradigma reativo

O paradigma reativo, representado pela Figura 10(b), tem sido utilizado na robótica desde 1988 (MURPHY, 2000). Ao contrário do paradigma hierárquico que utiliza uma estrutura *top-down*, onde toda ação é resultado da entrada de dados captada pelos sensores, seguida do planejamento e finalmente de sua execução. No paradigma reativo assume-se que toda entrada sensorial está relacionada a uma determinada ação a ser executada pelo robô. Esse mapeamento sensor-atuador é implementado por

meio de processos concorrentes, chamados de comportamentos (MURPHY, 2000). Cada comportamento pode ser divido entre vários outros, isso vai depender da complexidade e do tipo de cada um deles.

3.6.1.3 Paradigma híbrido

O paradigma híbrido, representado pela Figura 10(c), surgiu por volta de 1990 da combinação do paradigma hierárquico com o paradigma reativo. Seu funcionamento consiste inicialmente em planejar (*off-line*) todos os comportamentos necessários para que o robô consiga completar sua missão e em seguida mapear cada comportamento a determinados padrões de dados obtidos pelos sensores. Além disso, todos os comportamentos são executados através do paradigma reativo, de forma que, todas as informações sensoriais obtidas pelo robô sejam compartilhadas entre todos os comportamentos. Dessa forma, cada comportamento pode reagir independentemente.

O paradigma híbrido foi utilizado neste trabalho por oferecer o melhor do paradigma reativo e deliberativo, como reagir rapidamente a estímulos externos (reativo) e planejar suas ações (deliberativo). Essas características possibilitam que mesmo aplicações que executam tarefas dispendiosas computacionalmente, podem reagir rapidamente a situações inesperadas, devido a capacidade de o paradigma reativo ligar diretamente a saída dos sensores a todos os comportamentos.

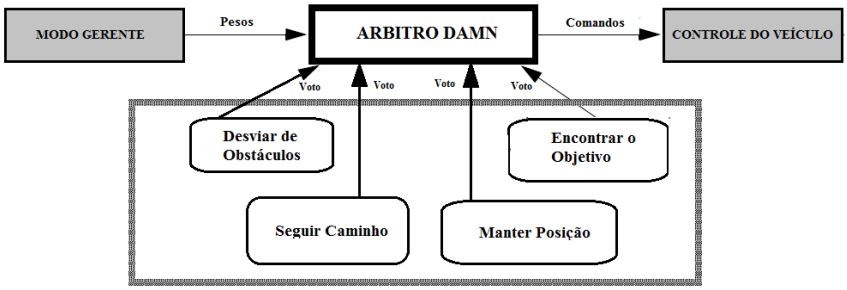
3.6.2 Arquitetura Híbrida DAMN

A *Distributed Architecture for Mobile Navegation* (DAMN) foi criada por Rosenblatt (1997) ao desenvolver sua tese de Pós-Doutorado na Universidade de Carnegie Mellon, Pittsburgh. Adotando o paradigma híbrido, essa arquitetura se diferencia das demais por possuir uma unidade controladora chamada de árbitro, responsável por gerenciar a prioridade na ordem de execução dos comportamentos.

A organização da DAMN consiste inicialmente na definição dos comportamentos necessários para o robô realizar determinadas tarefas como andar, desviar de obstáculos e definir rotas. A DAMN permite que todos os comportamentos tenham acesso a todos os níveis de controle do robô, através de uma votação. Essa votação consiste em atribuir pesos aos comportamentos de forma que reflitam a prioridade relativa em controlar o veículo (ROSENBLATT, 1997). Esses pesos devem mudar com o passar do tempo senão o sistema torna-se tendencioso. Uma maneira para que isso não ocorra é variar o valor

dos pesos durante o curso da missão de acordo com a relevância de cada comportamento em uma dada situação (ROSENBLATT, 1997). Essa variação dos pesos é tarefa de um módulo gerenciador.

Figura 11 – DAMN Árbitro e Comportamentos.



Fonte: Adaptado de Rosenblatt (1997)

Como pode ser visto na Figura 11 todos os comportamentos são construídos dentro de um bloco geral separado do árbitro, do gerente e do módulo de controle. Essa independência permite que novos comportamentos possam ser inseridos ou retirados sem interferir na estrutura do sistema robótico.

A DAMN opera assincronamente, permitindo que comportamentos possam ser executados paralelamente enviando suas saídas para o árbitro, que se encarrega de priorizar sua execução.

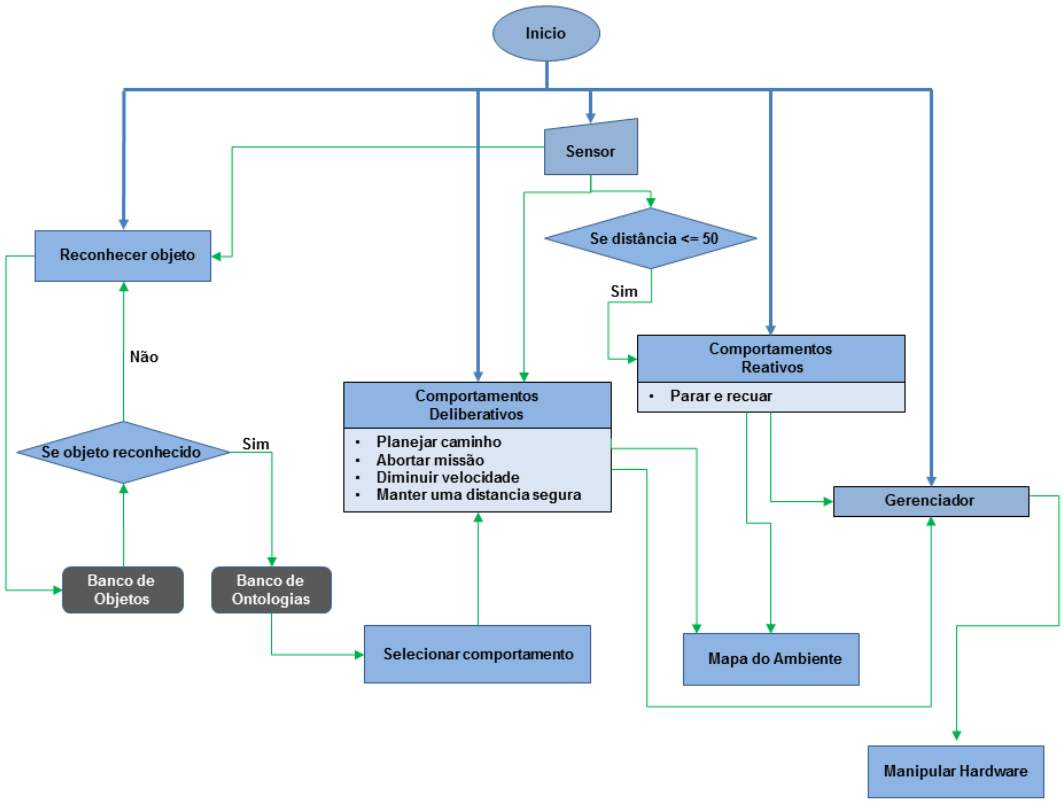
Nesse trabalho resolveu-se utilizar o modelo dessa arquitetura como base para o desenvolvimento do sistema de navegação. A sua escolha foi motivada principalmente pela robustez do paradigma híbrido e sua capacidade de trabalhar com comportamentos de forma paralela. Outro ponto que favoreceu a adoção desse modelo de arquitetura foi a utilização de uma figura centralizada de gerenciamento, permitindo sua aplicação não somente a hardware paralelo mas também a arquiteturas sequenciais.

4 SISTEMA DE NAVEGAÇÃO BASEADO EM COMPORTAMENTOS

O sistema desenvolvido nesse trabalho utiliza os conceitos e técnicas abordadas no referencial teórico, apresentado no capítulo anterior. Nesse capítulo, o sistema de navegação será descrito de forma modular, buscando detalhar o funcionamento de cada parte que constitui sua arquitetura, assim como o seu funcionamento de forma global.

4.1 PRINCÍPIO DE FUNCIONAMENTO

Figura 12 – Fluxo de execução do sistema



Fonte: Próprio autor.

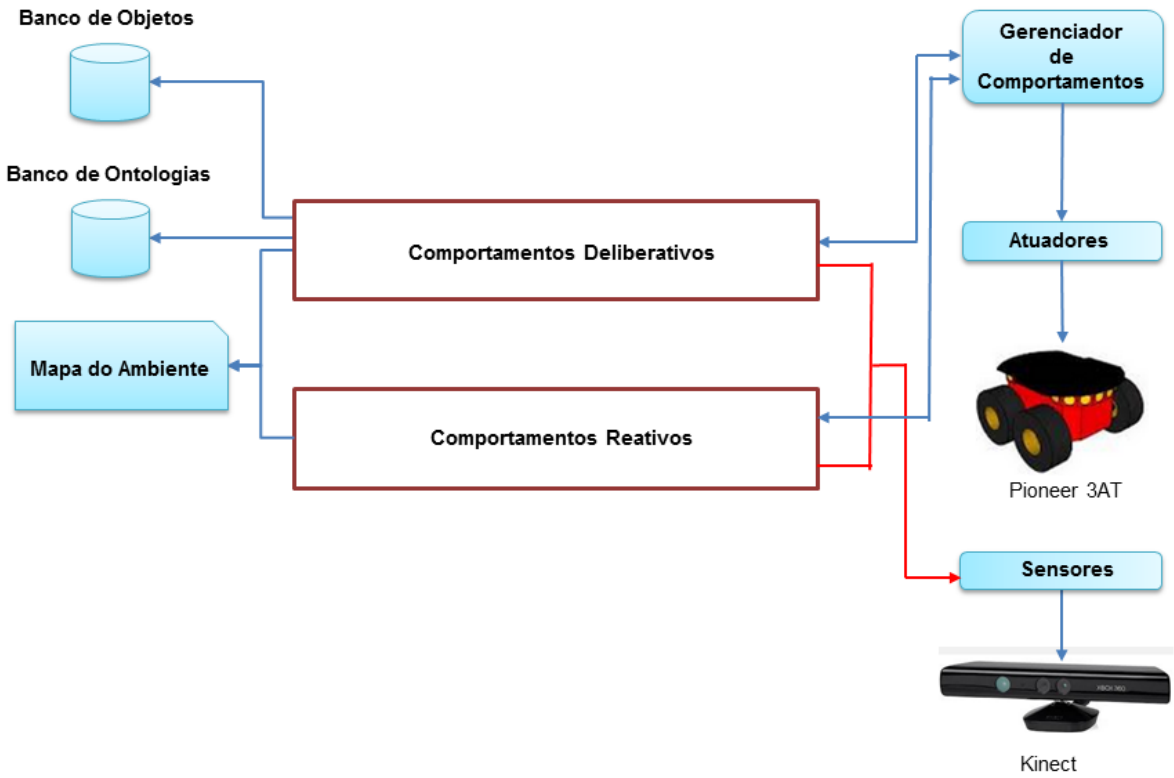
Como pode ser observado na Figura 12, o sistema desenvolvido inicia-se abrindo cinco fluxos de execução paralela, representados pelas linhas mais esparsas de cor azul. O primeiro fluxo, (da esquerda pra direita) é responsável pelo processo de reconhecimento dos objetos; o segundo fluxo ativa os comportamentos deliberativos; o terceiro é

responsável por ficar recebendo informações sensoriais do Kinect; o quarto ativa os comportamentos reativos e o quinto inicializa o gerenciador de comportamentos.

O terceiro fluxo sempre fica enviando a leitura dos sensores para os blocos de reconhecimento de objetos, comportamentos reativos e comportamentos deliberativos de forma paralela (simulado, pois está sendo utilizado um computador com apenas um processador). Quando o bloco "Reconhecer objeto" recebe uma imagem, o sistema em seguida acessa o "Banco de Objetos" procurando padrões similares a imagem de entrada. Caso os padrões coincidam, o objeto é reconhecido e sua respectiva ontologia é buscada no "Banco de Ontologias". Com base nas informações da ontologia, o sistema seleciona o comportamento deliberativo adequado e solicita acesso ao "Gerenciador" para manipular o hardware do robô. Quando o terceiro fluxo envia dados para o bloco de "Comportamentos Reativos" e "Comportamentos Deliberativos", uma condição verifica se algum ponto da cena possui um valor de distância menor ou igual a 50cm. Caso positivo, um comportamento reativo (para esse trabalho, apenas o comportamento "Parar e recuar") é ativado imediatamente e uma solicitação de acesso é requisitada ao "Gerenciador". O segundo fluxo abre comunicação para o bloco de "Comportamentos Deliberativos" ficar recebendo dados do sensor e também acessar dados de mapeamento do bloco "Mapa do Ambiente".

A arquitetura do sistema está representada na Figura 13.

Figura 13 – Arquitetura do sistema



Fonte: Próprio autor.

4.2 BLOCO BANCO DE ONTOLOGIAS

A função desse bloco é guardar a descrição formal das características e comportamentos que regem o conceito de cada objeto que o robô é capaz de reconhecer. A forma escolhida para realizar essa descrição foi através das ontologias. Dessa forma, quando o robô reconhece algum objeto, ele busca automaticamente nessa base de dados a sua ontologia correspondente. Com a ontologia do objeto, o sistema pode escolher qual comportamento adotar com base nas descrições e relacionamentos que a ontologia oferece e, assim, compor a semântica do objeto.

Para o desenvolvimento desse trabalho foi criado uma ontologia para uma cadeira de escritório e outra para um robô. A estrutura taxonômica de ambas as ontologias podem ser encontradas no apêndice 7. Com o desenvolvimento das ontologias foi possível criar uma base de dados e em seguida adicionar informações sobre o robô X80 e da cadeira de escritório.

A implementação da base de ontologias consistiu em instanciar cada classe com os atributos relacionados ao tipo de objeto que o sistema é capaz de reconhecer. A Tabela 3 mostra os valores instanciados para o objeto "Cadeira", utilizado nesse trabalho.

Tabela 3 – Atributos para uma instância de Cadeira

Forma Geométrica	Poliedro	Irregular	
Matéria	Estado Físico	Sólido	
	Mistura	Ferro, plastico, borracha	
	Propriedades da Matéria	Peso	58.8N
		Massa	6kg
Medida	Aceleração	0 m/s	
	Velocidade	0 m/s	
	Volume	0.50 m3	
	Área	1.38 m2	
	Densidade	12 kg/m3	
Ser não vivo	Cadeira	Apoio de braço	0
		Base	1
		Encosto	1
		Pé	Rodízio

Fonte: Próprio autor.

4.3 BLOCO BANCO DE OBJETOS

A função desse bloco é guardar informações que representem cada objeto que o sistema é capaz de reconhecer. As informações utilizadas nesse trabalho para realizar o reconhecimento de objetos foram os descritores de pontos. Esses descritores são estruturas de dados que conseguem, com base em um determinado ponto de uma imagem gerar uma descrição dos *pixels* de sua vizinhança, que possibilite seu reconhecimento em outras imagens. Os descritores de pontos foram obtidos através do algoritmo *Speeded-Up Robust Feature* (SURF)(BAY et al., 2008), implementado pela biblioteca *Open Source Computer Vision Library* (OPENCV). Para detecção dos pontos foi utilizado a classe *SurfFeatureDetector* através do método *void detect(const Mat image, vector<KeyPoint> keypoints, const Mat mask=Mat()) const* que recebe como parâmetro, uma imagem *const Mat image* e retorna os pontos detectados *vector<KeyPoint> keypoints*. Após a detecção, os pontos são passados como parâmetro para o método *void compute(const Mat image, vector<KeyPoint> keypoints, Mat descriptors) const* que realiza a descrição da vizinhança de cada *keypoint* e os armazena em *Mat descriptors*.

Para o reconhecimento dos objetos, foi preciso inicialmente armazenar os descritores de pontos de quatro imagens tiradas de ângulos diferentes, de cada objeto. Em seguida, todos os descritores armazenados, juntamente com os descritores obtidos da imagem do Kinect foram analisados, procurando identificar correspondências boas. Essa análise foi obtida por meio da *Fast Library for Approximate Nearest Neighbors* (FLANN) (MUJA; LOWE, 2009), uma biblioteca formada por uma coleção de algorítmos, otimizados para realizar correspondência de pontos entre vizinhos mais próximos. Utilizando a interface que a biblioteca OPENCV oferece para se trabalhar com a FLANN, foi possível realizar a correspondência entre descritores por meio do método *match(const Mat queryDescriptors, const Mat trainDescriptors, vector<DMatch> matches, const Mat mask=Mat()) const* da classe *FlannBasedMatcher*. Esse método recebe como parâmetro, os descritores das duas imagens: *const Mat queryDescriptors, const Mat trainDescriptors*, e devolve apenas os descritores que possuem um determinado grau de similaridade, representado por *<DMatch> matches*.

4.4 BLOCO MAPA DO AMBIENTE

Esse bloco é responsável por armazenar o mapa do ambiente aonde o robô realizará seu trajeto, sendo acessado tanto pelo bloco de comportamentos reativos (para mapear a posição dos objetos desconhecidos, que surgem durante o processo

de navegação), como pelo bloco de comportamentos deliberativos, nas atividade de planejamento de caminhos e reconhecimento de objetos (para mapear a posição dos objetos reconhecidos). O mapa adotado nesse trabalho utilizou uma abordagem métrica do tipo grade de ocupação determinística (sem tratamento estatístico), o que facilita o planejamento de caminhos com a técnica de decomposição por células aproximadas. No entanto, o sistema desenvolvido permite que outros tipos de mapas possam ser facilmente adaptados para serem utilizados pelo robô.

4.5 BLOCO SENSORES

Esse bloco é responsável por disponibilizar aos blocos de comportamentos reativos e deliberativos, informações capturadas pelos sensores do robô. Todos os comportamentos podem acessar simultaneamente os dados sensoriais. Para esse trabalho foi utilizado como sensor o dispositivo da Microsoft o Kinect (MICROSOFT, 2014). Esse dispositivo captura do ambiente uma imagem RGB-D que além de informações de cores, também disponibiliza dados de profundidade. Estes são diretamente utilizados para o acionamento dos comportamentos reativos e de forma indireta pelos comportamentos deliberativos, no replanejamento de caminhos.

4.6 BLOCO ATUADORES

Esse bloco é responsável por controlar todos os atuadores do robô, ou seja seus motores. Dessa forma, quando um comportamento necessita executar uma determinada tarefa, que precise manipular os atuadores do robô, esse bloco simplesmente abstrai todas as funcionalidades de baixo nível de manipulação do hardware. Os atuadores são controlados e acionados a partir de um controle cinemático de posição apresentado por Siegwart e Nourbakhsh (2004). Para esse trabalho foi utilizado o robô Pioneer 3-AT, desenvolvido pela ActivMedia Robotcs.

4.7 BLOCO COMPORTAMENTOS REATIVOS

Os comportamentos reativos foram projetados para reagirem diretamente a estímulos recebidos do ambiente por meio do sensor, sem necessitar de nenhum tipo de planejamento ou "inteligência". Por ter essas características, os comportamentos reativos são utilizados principalmente para o robô evitar colisões com objetos imprevistos.

Nesse trabalho foi desenvolvido somente um comportamento reativo;

- Parar e retroceder

  Esse comportamento é responsável somente por parar o robô e em seguida retroceder uma determinada distância, caso algum objeto se aproxime a menos de 10 centímetros dele. Levando em consideração que a distância mínima necessária para que o Kinect consiga obter os valores de profundidade de uma imagem é de aproximadamente 50 cm. Optou-se por colocar o kinect na parte traseira do robô, para que a distância entre a parte frontal do robô e o objeto a sua frente, fosse no máximo 10 cm, respeitando a distância mínima que o Kinect necessita.

O conjunto de comportamentos reativos pode ser expandido de acordo com as funcionalidades do robô utilizado. Assim, sons de alerta, luzes de atenção, entre outros, podem ser implementados como comportamentos reativos.

4.8 BLOCO COMPORTAMENTOS DELIBERATIVOS

Esse bloco foi criado para armazenar todos os comportamentos relacionados a raciocínio e "inteligência" do robô. Os comportamentos deliberativos implementados nesse trabalho foram: planejar caminho, abortar missão, diminuir velocidade e por fim manter uma distância segura do objeto.

- Planejar caminho

  Com base nas informações do mapa, esse comportamento consegue planejar qual o caminho que o robô deve seguir partindo de um determinado ponto inicial até um ponto final.

  Para desenvolver o planejamento de caminho foi necessário utilizar a técnica de decomposição de células aproximadas. No entanto, para tornar mais simples o desenvolvimento dessa fase, não foi construído o grafo de conectividade, padrão dessa técnica e sim utilizado a função de Manhattan com um algoritmo de busca sobre o mapa decomposto.

A função de navegação de Manhattan utiliza o conceito da Métrica retangular, também conhecida como Distância de Manhattan, Métrica Metropolitana ou Distância do Táxi. Essa técnica utiliza como base o conceito de campo potencial, para preencher e calcular um caminho (caso ele exista) entre uma posição inicial e final de um mapa em grade de ocupação.

O cálculo da distância entre dois pontos utilizando a Distância de Manhattan pode ser obtido através da expressão 4.1. Onde, $Dist_M$ representa a distância entre dois pontos A e B em um plano bidimensional.

$$Dist_M = [|X_A - X_B| + |Y_A - Y_B|] \tag{4.1}$$

A Figura 14 mostra como é realizado o cálculo de um novo caminho utilizando a navegação de Manhattan.

Figura 14 – Função de navegação de Manhattan.

2	1	2	3	4	5		
1	**fim**	1	2	3	4		
2	1	2	3	4	5		
3	2			5	6		
4	3			6	7	8	**Inicio**
5	4			7	8	9	10
6	5	6	7	8	9	10	11
7	6	7	8	9	10	11	12

Fonte: Próprio autor.

Para calcular a trajetória do robô, foi utilizado a grade de ocupação do ambiente mapeado, onde as células de cor preta representam obstáculos e as células de cor branca indicam lugares por onde o robô pode navegar. Posteriormente, foi atribuído o valor zero a célula correspondente a posição final do robô e o valor de cada célula vizinha a posição final foi incrementado em uma unidade, menos as células ocupadas por obstáculos. Em seguida foi necessário procurar o próximo menor valor e incrementar em uma unidade o valor de suas células vizinhas. Esse procedimento foi realizado até que todas as células do mapa fossem preenchidas.

Com o mapa preenchido, o próximo passo foi utilizar à célula correspondente a posição inicial do robô para achar a próxima posição do caminho. Para isso foi utilizado uma busca nos valores das oito células vizinhas. A célula com menor valor é escolhida para compor o caminho e assim, se repete até chegar na célula correspondente a posição final. O caminho gerado pela Função de Navegação de Manhattan está representado na Figura 14, pelas células de cor cinza.

- Abortar missão

  Esse comportamento é responsável por cancelar a missão imposta ao robô devido a algum motivo que o impeça de continuar sua trajetória. O motivo escolhido foi quando ele não conseguir planejar um novo caminho, geralmente quando um obstáculo está bloqueando sua passagem.

- Diminuir velocidade

  Esse comportamento altera a velocidade do robô de acordo com o tipo de objeto reconhecido. Para os objetos reconhecidos pelo sistema, foram estipuladas as seguintes velocidades em metros por segundo.

  - Cadeira – (Não altera a velocidade)
  - Robô (X80) – 0.1m/s

- Manter uma distância segura do objeto

  Esse comportamento consegue limitar a distância de aproximação do robô, de acordo com o tipo de objeto reconhecido. Os seguintes valores foram estipulados em centímetros.

  - Cadeira – 0 cm
  - Robô (X80) – 60 cm

4.9 BLOCO GERENCIADOR DE COMPORTAMENTOS

A definição desse bloco é bem similar a de um escalonador de processos no contexto de sistemas operacionais. Primeiramente, os comportamentos solicitam ao gerenciador, acesso ao hardware do robô. Dependendo do tipo de comportamento (reativo ou deliberativo) o gerenciador pode liberar ou recusar seu acesso. Todo comportamento reativo possui uma prioridade de execução maior que os comportamentos deliberativos. Caso um comportamento deliberativo esteja sendo executado e, de repente, um comportamento reativo pede acesso devido, por exemplo, a um risco de colisão, o gerenciador irá aceitar a solicitação do reativo e abortar imediatamente a execução do deliberativo.

A prioridade entre comportamentos do mesmo bloco é igual. Assim, se um comportamento deliberativo estiver em execução e outro deliberativo também solicitar acesso, o ultimo que solicitou terá que aguardar o fim do comportamento em execução. No entanto, no caso dos comportamentos reativos (caso exista mais de um), todos eles podem interromper um ao outro.

5 EXPERIMENTOS

Para validar esse trabalho foram realizados alguns experimentos práticos, com o intuito de demonstrar o sistema de navegação desenvolvido, utilizando a abordagem de reconhecimento de objetos, aplicado ao processo de navegação robótico.

Os experimentos foram feitos com um robô Pioneer 3-AT munido de um sensor RGB-D Microsoft Kinect, apresentado pela Figura 15

Figura 15 – Plataforma robótica utilizada.

Fonte: Próprio autor.

5.1 AMBIENTE DE TESTES

O ambiente utilizado para se realizar os testes foi uma sala de laboratório, representado pela Figura 16, com 6,0 metros de comprimento e 3,90 metros de largura. A sala é composta por bancadas, cadeiras, armários e outros objetos de pequeno porte.

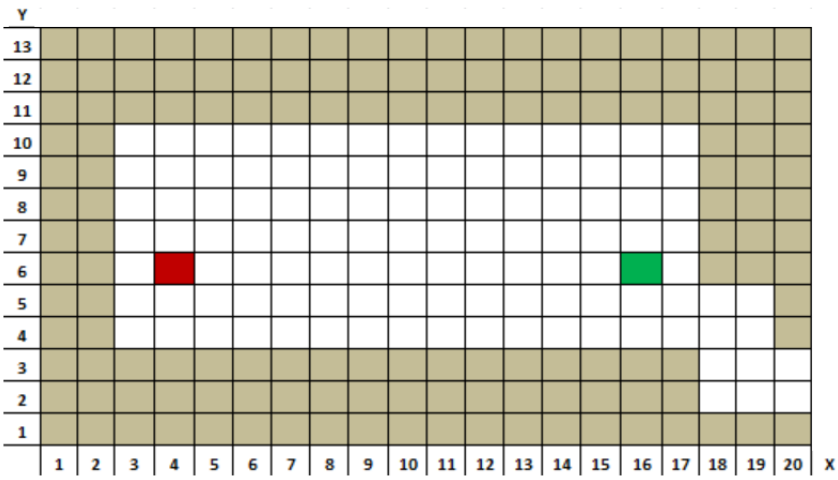
Esse espaço foi escolhido, por possuir dimensões mais amplas, favorecendo uma maior manobrabilidade para o robô.

Figura 16 – Ambiente de testes

Fonte: Próprio autor.

O ambiente escolhido foi mapeado previamente de forma manual (com fita métrica) e suas dimensões foram utilizadas para criar um mapa em grade de ocupação de duas dimensões, utilizando células quadradas com dimensões de $0,3m$ (dimensões das lajotas que compõe o piso utilizado no ambiente de testes). O mapa criado está representado pela Figura 17, onde as células de cor branca indicam espaços vazios por onde o robô pode se locomover e as células de cor marrom os espaços ocupados. Em todos os experimentos o robô tem a missão de sair do ponto $x = 1,05m$ e $y = 1,65m$ (célula de cor vermelha) e chegar no ponto $x = 4,65m$ e $y = 1,65m$ (célula de cor verde).

Figura 17 – Mapa em grade de ocupação



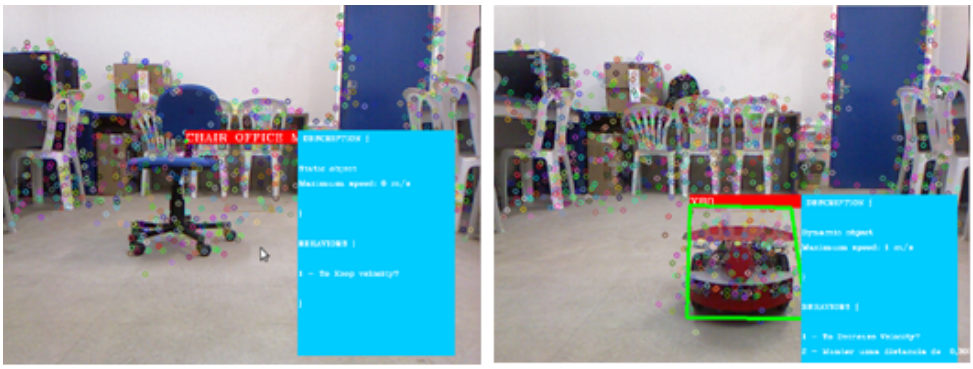
Fonte: Próprio autor.

5.2 EXPERIMENTOS DO RECONHECIMENTO DE OBJETOS

Para a etapa do reconhecimento foram utilizados dois objetos: uma cadeira e um robô modelo X80. Para cada objeto foram tiradas quatro imagens de ângulos diferentes, permitindo que o objeto fosse reconhecido mais facilmente. Em média a resolução das oito imagens que compõe o reconhecimento dos dois objetos ficou de 153 por 170 *pixels*.

A Figura 18 mostra a interface gráfica do sistema ao reconhecer os objetos.

Figura 18 – Interface gráfica do reconhecimento de objetos. (a) Objeto cadeira, (b) Objeto robô X80



Fonte: Próprio autor.

Quando um objeto é reconhecido um quadro na cor azul mostra as características do objeto e os comportamentos que ele deve adotar. Essas informações sobre o objeto são obtidas por meio de sua ontologia. A Tabela 4 mostra as descrições e comportamentos (presente no retângulo azul da Figura 18) adotados para cada objeto.

Tabela 4 – Descrições e comportamentos dos objetos reconhecidos

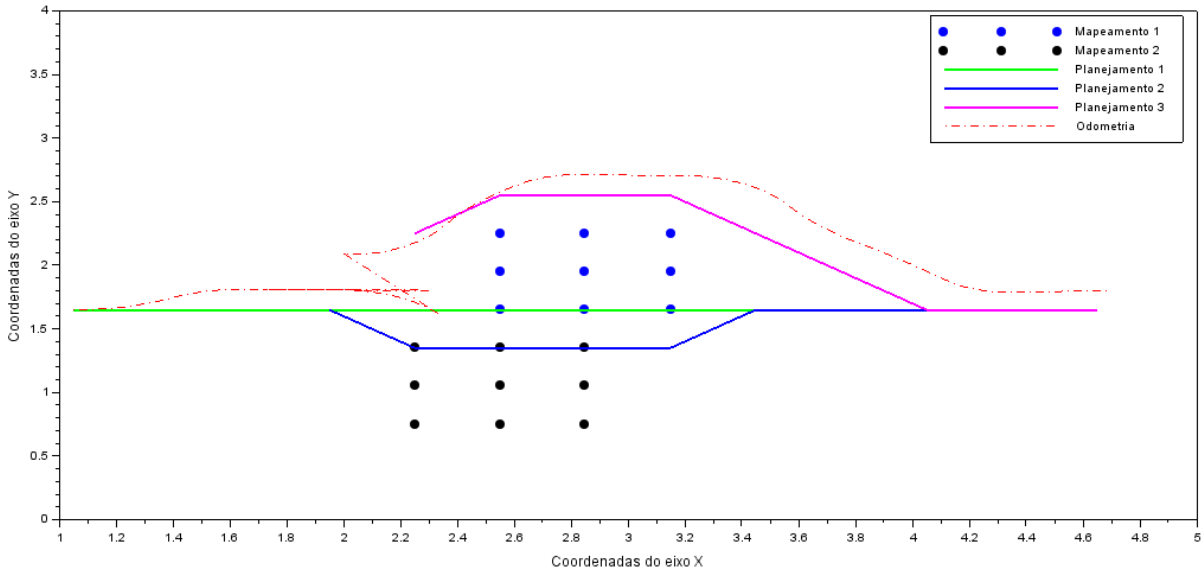
Robô X80	Descrições	Objeto dinâmico
		Velocidade máxima: 1m/s
	Comportamentos	Diminuir velocidade
		Manter uma distância de 60 cm
Cadeira	Descrições	Objeto estático
		Velocidade máxima: 0m/s
	Comportamentos	Manter velocidade

Fonte: Próprio autor.

5.3 EXPERIMENTOS DOS COMPORTAMENTOS REATIVOS

Para verificar o funcionamento dos comportamentos reativos foi criado um cenário, onde o robô planeja sua rota utilizando o mapa inicial e em seguida, um objeto dinâmico representado por uma pessoa, bloqueia seu caminho, impedindo-o e obrigando-o a desviar e refazer seu percurso novamente. A partir do momento que um obstáculo é detectado e ele não está no mapa inicial, então o sistema atualiza o mapa e replaneja o caminho. O resultado do teste está representado pelo gráfico da Figura 19.

Figura 19 – Comportamentos reativos - Odometria, Mapeamento e Planejamento



Fonte: Próprio autor.

No gráfico da Figura 19 o robô começa sua missão realizando seu primeiro planejamento (linha de cor verde), que consiste em sair da posição *(x=1,05m e y=1,65m)* e chegar na posição *(x=4,65m e y=1,65m)* do mapa. A linha tracejada em vermelho representa o trajeto real que o robô realizou, segundo a sua odometria. Seguindo esse planejamento, o robô consegue chegar até a posição *(x=2,27m e y=1,81m)* onde um objeto inesperado é detectado a uma distância de 28cm do robô e sua posição mapeada (pontos de cor azul), atualizando o mapa do ambiente e obrigando o robô a parar e recuar para a posição *(x=1,72m e y=1,81m)*. Logo em seguida, o robô realiza seu segundo planejamento (linha de cor azul), que consiste em sair da posição *(x=1,72m e y=1,81m)* e chegar na posição *(x=4,65m e y=1,65m)*. Seguindo o segundo planejamento, o robô consegue chegar até a posição (x=2,30m e y=1,61m), onde novamente um objeto é detectado a uma distância de 27cm do robô e sua posição mapeada, atualizando o mapa e obrigando-o a parar e recuar para a posição *(x=1,98m e y=2,20m)*. Agora, o robô realiza seu terceiro planejamento (linha de cor roxa), sair da posição *(x=1,98m e*

y=2,20m) e chegar na posição *(x=4,65m e y=1,65m)*. Seguindo o ultimo planejamento, o robô consegue completar sua missão, chegando na posição *(x=4,65m e y=1,65m)*.

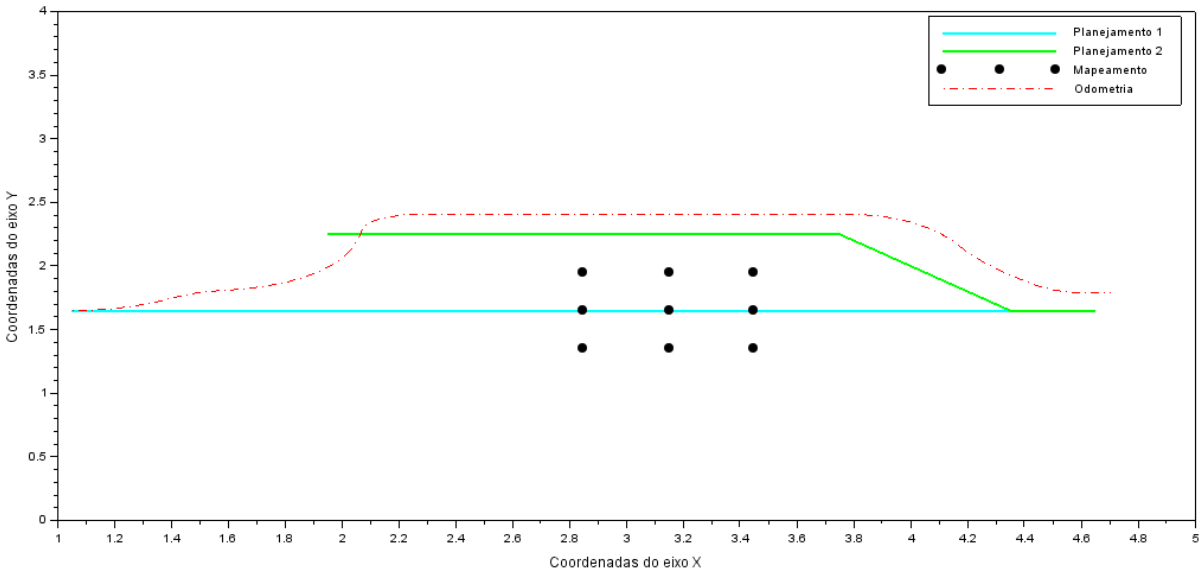
5.4 EXPERIMENTOS DOS COMPORTAMENTOS DELIBERATIVOS

Para testar os comportamentos deliberativos foram montados dois cenários. O primeiro cenário busca analisar o comportamento do robô quando ele reconhece um obstáculo estático. Já o segundo cenário foi utilizado para verificar seu comportamento ao reconhecer um obstáculo dinâmico.

5.4.1 Objetos estáticos

No primeiro cenário, o robô tem a missão de sair da posição *(x=1,05m e y=1,65m)* e chegar na posição *(x=4,65m e y=1,65m)* do mapa, adotando um comportamento que esteja relacionado com o tipo de objeto que ele consegue reconhecer. Para esse teste, foi utilizado uma cadeira de escritório.

Figura 20 – Comportamentos deliberativos - Odometria, Mapeamento e Planejamento (Cadeira).



Fonte: Próprio autor.

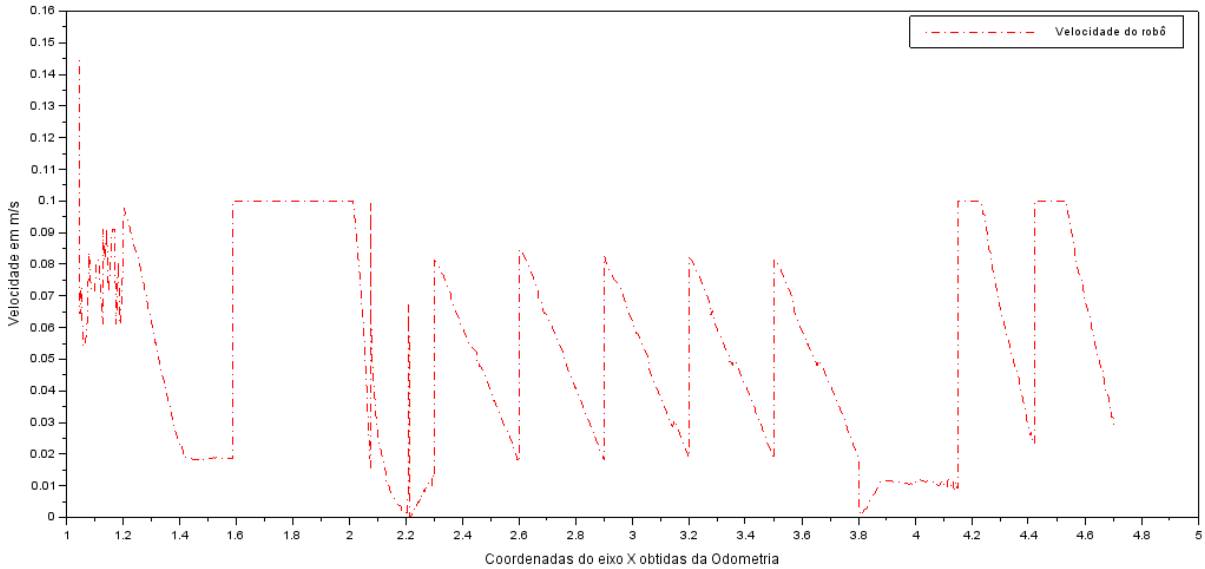
Nesse cenário, o robô começa executando seu primeiro planejamento (linha de cor ciano), que consiste em sair da posição *(x=1,05m e y=1,65m)* e chegar na posição

(x=4,65m e y=1,65m). Seguindo o primeiro planejamento, o robô consegue chegar até a posição *(x=1,91m e y=2,29m)* onde acontece o reconhecimento do objeto ”cadeira” (pontos de cor preta). O objeto reconhecido é mapeado e um novo planejamento acontece, levando em consideração os comportamentos para esse tipo de objeto. Como a cadeira se trata de um objeto estático sem velocidade de locomoção própria, o comportamento adotado pelo robô foi manter sua velocidade. Esse novo planejamento (linha de cor verde) consiste em sair da posição *(x=1,91m e y=2,29m)* e chegar na posição *(x=4,65 e y=1,65)*. Seguindo esse segundo planejamento o robô consegue completar sua missão e chegar na posição *(x=4,65m e y=1,65m)*.

Novamente, a linha pontilhada em vermelho mostra o caminho real executado pelo robô, de acordo com os cálculos do sistema de localização baseado em odometria.

Para verificar a mudança de comportamento do robô quando ele reconhece um determinado objeto, sua velocidade de locomoção foi monitorada e representada pelo gráfico da Figura 21 para demonstrar que sua velocidade máxima não foi alterada quando o objeto "cadeira"foi reconhecido.

Figura 21 – Comportamentos deliberativos - Variação da velocidade (Cadeira).



Fonte: Próprio autor.

Pelo gráfico da Figura 21 percebe-se que a velocidade do robô manteve-se oscilante na maior parte do tempo. Isso acontece porque o controlador de posição cinemático desenvolvido para o robô, calcula sua velocidade máxima de célula para célula, ocasionando as ondas dente de serra no gráfico da Figura 21. Em alguns locais como na posição *(x=2,2m e x=3,8m)*, a velocidade do robô chega a zero, devido a ocorrência de replanejamentos de caminhos e movimentos de rotações. Toda vez que necessita-se de um novo planejamento, o robô deve parar, replanejar e em seguida continuar seu trajeto.

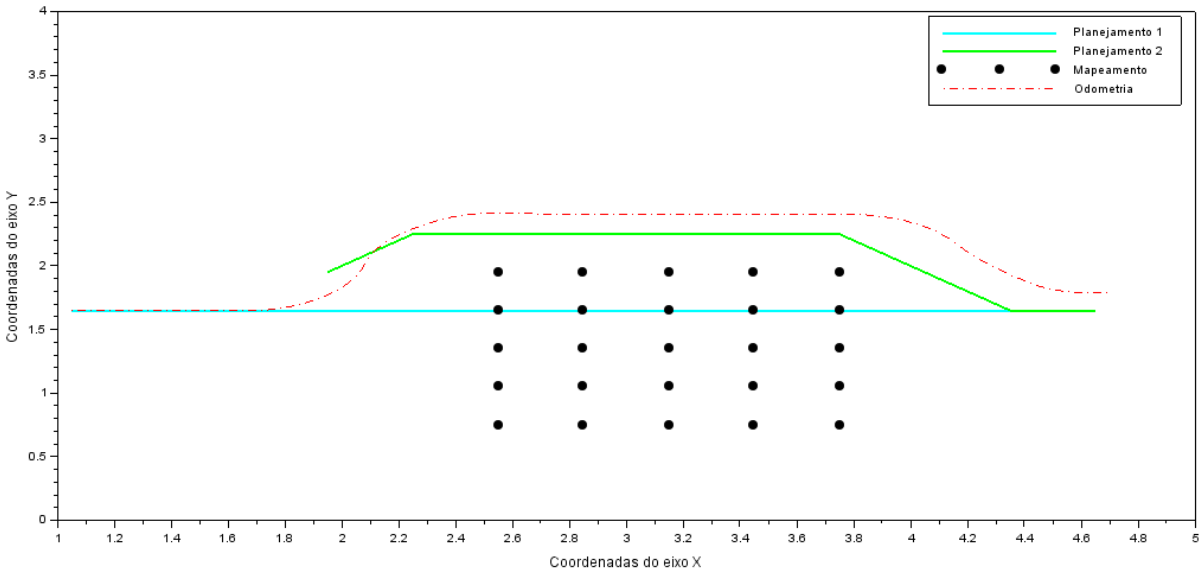
Uma forma de solucionar isso, seria otimizar o planejamento de caminhos, de forma a descartar células intermediárias que estão no mesmo eixo x ou y do trajeto, deixando apenas as células inicial e final. Isso permitiria, que o robô atingisse níveis de velocidade mais superiores e constante durante seu trajeto.

No geral, a velocidade do robô manteve-se entre o esperado para o tipo de objeto reconhecido, abaixo de 0.1 m/s.

5.4.2 Objetos dinâmicos

O segundo cenário representado pelo gráfico da Figura 22, aborda o comportamento do robô ao reconhecer objetos dinâmicos. Para esse experimento foi utilizado outro robô chamado X80, como objeto dinâmico inserido nesse contexto. Desse forma, o sistema deve reconhecer o robô X80 e alterar sua navegação de acordo com as descrições do objeto, encontradas na sua ontologia.

Figura 22 – Comportamentos deliberativos - Odometria, Mapeamento e Planejamento (Robô X80).



Fonte: Próprio autor.

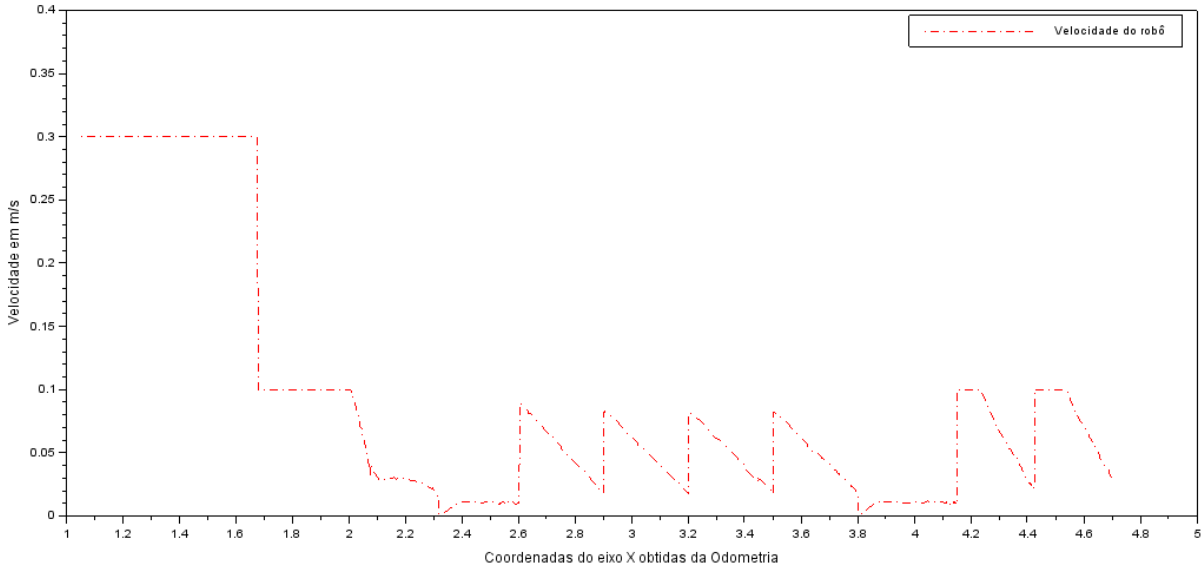
Nesse cenário, o robô inicia sua missão na posição *(x=1,05m e y=1,65m)*, realizando seu primeiro planejamento (linha de cor ciano), que consiste em sair da posição *(x=1,05m e y=1,65m)* e chegar na posição *(x=4,65m e y=1,65m)*. Seguindo esse planejamento o robô consegue chegar até a posição *(x=1,68m e y=1,98m)* onde acontece o reconhecimento do objeto dinâmico "robô X80". Quando o objeto é reconhecido, o sistema mapeia sua posição no mapa e realiza seu planejamento de caminhos, levando em consideração os comportamentos adotados para esse tipo de objeto. Como o "robô X80" é um objeto

dinâmico, foi estabelecido que o robô (utilizando o sistema de navegação desenvolvido) diminuisse sua velocidade para valores inferiores a 0.1m/s e mantivesse uma distância de 60cm do objeto.

Após o reconhecimento do objeto dinâmico, o robô realiza seu segundo planejamento de caminhos que o conduz da posição *(x=1,68m e y=1,98m)* até chegar na posição *(x=4,65m e y=1,65m)*. Seguindo esse planejamento o robô consegue completar sua missão, adotando dois comportamentos específicos para o tipo de objeto reconhecido.

De forma geral, percebe-se que adotando o comportamento de manter distância, o robô alterou sua trajetória de forma esperada, contornando todas as células preenchidas (representando o objeto dinâmico) e consequentemente mantendo uma distância maior que 60 centímetros.

Figura 23 – Comportamentos deliberativos - Velocidade (Robô X80).



Fonte: Próprio autor.

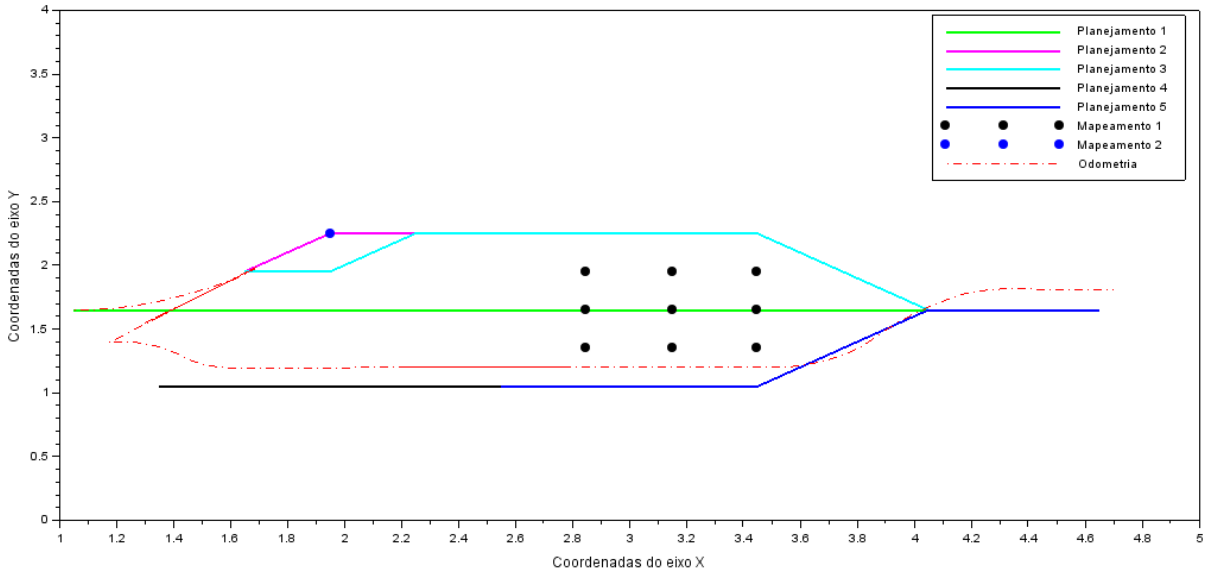
Observando o gráfico da Figura 23, fica evidente a diminuição da velocidade do robô ao reconhecer o objeto "robô X80" em relação ao experimento com a cadeira (objeto estático). No inicio da execução do trajeto, o robô começa com uma velocidade de 0,30 m/s. Logo em seguida, na coordenada *(x=1,68m e y=1,98m)* o robô reconhece o objeto e adota o comportamento de alterar sua velocidade máxima para 0,1m/s. A mudança de velocidade no intervalo *(x=2m e x=2,6m)* ocorreu quando o robô realizou uma rotação para contornar o objeto, acontecendo o mesmo no intervalo *(x=3,8m e 4,2m)*. As ondas dente de serra, representam a velocidade linear do robô, mantida abaixo de 0,1m/s à partir do momento que o objeto foi reconhecido.

5.5 EXPERIMENTO UTILIZANDO COMPORTAMENTOS REATIVOS E DELIBERATIVOS

Desde o inicio, o objetivo principal desse trabalho foi desenvolver um sistema de navegação "inteligente" utilizando uma arquitetura híbrida, ou seja, que o robô pudesse reagir espontaneamente a situações inesperadas e também decidir qual comportamento tomar baseando-se no tipo de objeto reconhecido. Assim, para concluir esse capítulo de testes foi realizado um experimento geral, para verificar como o sistema se comporta juntando o reconhecimento de objetos, os comportamentos reativos e os comportamentos deliberativos.

O cenário para esse teste consistiu em colocar o robô para seguir o mesmo trajeto que os experimentos anteriores, mas dessa vez, ele teve que desviar de um obstáculo imprevisto inserido no ambiente e também reconhecer o objeto "Cadeira" colocado no meio de sua trajetória inicial. O gráfico da Figura 24 mostra os resultados desse teste.

Figura 24 – Comportamentos deliberativos e reativos - Odometria, Mapeamento e Planejamento.



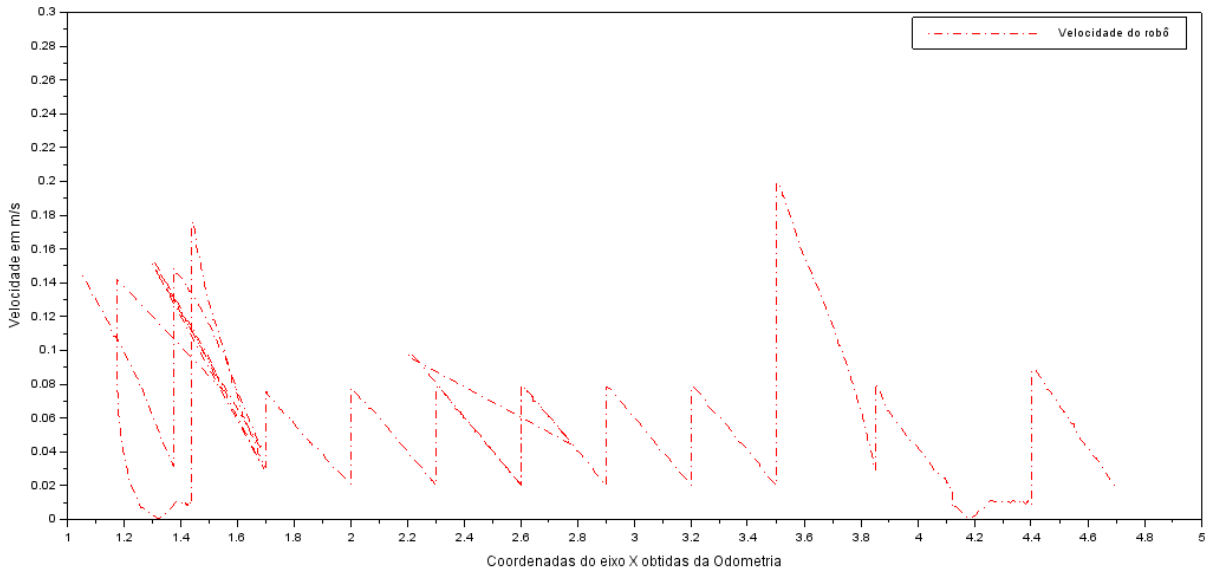
Fonte: Próprio autor.

Nesse experimento, o robô inicia sua missão na posição *(x=1,05m e y=1,65m)*, realizando seu primeiro planejamento de caminhos(linha de cor verde), que consiste em sair da posição *(x=1,05m e y=1,65m)* e chegar na posição *(x=4,65m e y=1,65m)*. Seguindo esse caminho, o robô consegue chegar até a posição *(x=1,20m e y=1,67m)* onde o objeto "Cadeira" é reconhecido. Após o reconhecimento, o sistema mapeia a posição do objeto (pontos de cor preta) no mapa e realiza o segundo planejamento (linha de cor roxa), que consiste em sair da posição *(x=1,20m e y=1,67m)* e chegar na posição *(x=4,65m e y=1,65m)*. O robô consegue seguir esse planejamento até a posição *(x=1,67m*

e y=1,98m) onde ele encontra um objeto inesperado (ponto de cor azul) que o obriga a parar, mapeá-lo e recuar para a posição *(x=1,36m e y=1,60m)*. Em seguida um terceiro planejamento (linha de cor azul) acontece, saindo da posição *(x=1,36m e y=1,60m)* até a posição *(x=4,65m e y=1,65m)*. No entanto, assim que o robô começa a executar o terceiro planejamento o objeto se move na sua direção, interrompendo novamente sua trajetória na posição *(x=1,66m e y=1,99m)*. Com a interrupção do trajeto o robô é obrigado a parar na posição *(x=1,66m e y=1,99m)* e recuar para *(x=1,13m e y=1,20m)*, onde acontece o quarto planejamento de caminhos (linha de cor preta). Esse planejamento consiste em sair da posição *(x=1,13m e y=1,20m)* e chegar na posição *(x=4,65m e y=1,65m)*. Seguindo esse planejamento, o robô deveria completar sua missão chegando na posição *(x=4,65m e y=1,65m)* sem nenhum problema.

No entanto, por erros de odometria, gerados pelos movimentos iniciais para desviar dos objetos, inflenciaram no sentido de localização do robô, chegando ao ponto que na posição *(x=2,82m e y=1,20m)* ele passa muito próximo do objeto mapeado (pontos de cor preta), ativando um comportamento reativo que força o robô a recuar para a posição *(x=2,20m e y=1,20m)*. Dessa forma, o quinto planejamento acontece saindo da posição *(x=2,20m e y=1,20m)* até a posição *(x=4,65m e y=1,65m)*. Para que o mesmo fato não se repetisse, já que o quinto planejamento foi igual ao quarto, foi necessário adaptar o ambiente real as informações de odometria do robô. Assim, o objeto "Cadeira" foi deslocado um pouco da rota do robô para não ativar novos comportamentos reativos. Dessa forma, seguindo o quinto planejamento o robô conseguiu completar sua missão chegando na posição *(x=4,65m e y=1,65m)*.

Figura 25 – Comportamentos deliberativos e reativos - Velocidade.



Fonte: Próprio autor.

O gráfico da Figura 25 demonstra que em média a variação da velocidade se manteve com valores inferiores a 0,1m/s, representadas através das ondas dente de

serra. Algumas exceções aconteceram entre as posições *(x=1,0m e x=1,70m)*, resultado da ativação dos comportamentos reativos do início do teste. Também ocorreu um aumento expressivo da velocidade na posição *(x=2,20m e y=1,20m)*, devido a ocorrência inesperada de um comportamento reativo, gerado quando o robô passou muito perto do objeto "Cadeira". Outra variação de velocidade ocorreu na posição *(x=3,35m e y=1,20m)*, quando o robô realiza um giro no sentido anti-horário. De forma geral, o sistema se comportou como o esperado, conseguindo evitar colisões com objetos dinâmicos que surgiram inesperadamente no seu caminho e adotar um comportamento de navegação específico para o tipo de objeto reconhecido, que nesse teste foi manter sua velocidade ao reconhecer o objeto "Cadeira".

6 CONCLUSÃO E TRABALHOS FUTUROS

O trabalho desenvolvido buscou contribuir de forma positiva, para tornar o processo de navegação de robôs móveis ainda mais inteligente, utilizando para isso, um sistema de navegação baseado na mudança de comportamento do robô ao reconhecer um determinado objeto.

O sistema de navegação foi desenvolvido utilizando os princípios do paradigma híbrido e baseado no modelo da arquitetura DAMN, permitindo que o robô possuisse tanto comportamentos de baixo tempo de reação, quanto comportamentos de "inteligência" que necessitam de um maior tempo para serem processados. Utilizando esse tipo de paradígma foi possível criar comportamentos deliberativos que conseguissem alterar a forma de navegação do robô baseando-se nas características dos objetos reconhecidos. Essas características, foram modeladas por meio de ontologias, com o objetivo de fornecer informações sobre o tipo de objeto armazenado.

Para o processo de reconhecimento de objetos, todas as imagens obtidas em tempo real pelo Kinect foram processadas pelo algorítmo SURF, objetivando extrair pontos de interesse para serem comparados com os descritores dos padrões armazenados previamente. Essa comparação foi realizada por correspondência de pontos utilizando o algorítmo FLANN, baseado na busca dos vizinhos mais próximos.

Unindo o reconhecimento de objetos com as descrições de cada um deles, foi possível criar comportamentos para o robô diminuir a velocidade quando reconhecesse o objeto "robô X80" ou "manter a velocidade" quando o objeto reconhecido fosse uma cadeira. Também foi possível estimar uma distância segura para o robô manter dos objetos. Para o objeto "robô X80" foi estabelecido uma distância segura de 60cm por se tratar de um objeto dinâmico, enquanto para o objeto "Cadeira" a distância não precisaria ser alterada já que se trata de um objeto estático que não se move por conta própria.

Analisando os resultados obtidos nos testes, percebeu-se que de forma geral o sistema de navegação desenvolvido obteve os resultados esperados, ou seja, o robô conseguiu reagir a obstáculos desconhecidos que apareciam em seu trajeto, e também conseguiu tomar diferentes comportamentos com base nos objetos reconhecidos.

Para o desenvolvimento desse trabalho foi necessário a utilização de várias áreas da robótica que são extremamente complexas quando trabalhadas de forma aprofundada. Por isso, e pela questão do tempo envolvido, a parte de odometria não foi tratada por nenhum método probabilístico, que pudesse diminuir a quantidade de erros envolvidos na navegação. Como esperado, esse fato prejudicou um pouco o sentido de localização do robô ao realizar muitas curvas e replanejamentos. No entanto, pelos

gráficos obtidos nos testes percebe-se que, em seu mapa o robô segue a trajetória correta, encontrada com base no reconhecimento dos objetos. Quanto aos comportamentos que alteram a velocidade do robô, os dados obtidos e representados nos gráficos demonstram que realmente a velocidade de locomoção do robô foi alterada à medida que os objetos foram reconhecidos. Para o objeto "robô X80" que começou com uma velocidade de 0,3m/s, diminuiu para 0,1m/s, enquanto o objeto "Cadeira" manteve sua velocidade média com valores inferiores a 0,1m/s durante todo o trajeto.

Para trabalhos futuros, deseja-se utilizar a odometria visual, por ser menos propensa a erros do que a odometria baseada em *encoders*. Quanto ao reconhecimento de objetos, pretende-se melhorá-lo tanto na forma qualitativa como quantitativa, verificando a questão de escalabilidade dos objetos a serem reconhecidos, já que se trata de um sistema de navegação robótico e em tempo real. Deseja-se também aprofundar-se um pouco mais na parte das ontologias, analisando o relacionamento entre conceitos e seleção de comportamentos utilizando Inteligência Artificial, objetivando criar comportamentos ainda mais "inteligentes".

REFERÊNCIAS

BAY, H. et al. Speeded-up robust features (surf). In: . [S.l.: s.n.], 2008. p. 346–359. Citado na página 42.

BISNETO, C. R. de B. *Reconhecimento de Objetos Utilizando Redes Neurais Artificiais e Geometria Fractal*. Dissertação (Mestrado) — Senai Cimatec, Curso de Modelagem Computacional e Tecnologia Industrial, Salvador, 2011. Citado na página 24.

BURGES, C. J. A tutorial on support vector machines for pattern recognition. In: . [S.l.]: IEEE, 1998. p. 121–167. Citado na página 20.

COZMAN, V. F. P. P. E. S. F. G. Reconhecimento de objetos baseado em contexto utilizando a lógica de descrição probabilística cralc. In: . Curitiba: [s.n.], 2012. p. 1–11. Citado na página 24.

GRACIANO, A. B. V. *Modelagem e reconhecimento de objetos estruturados: Uma abordagem estatístico-estrutural*. Tese (Doutorado) — Universidade de São Paulo, Curso de Ciências, São Paulo, 2012. Citado 2 vezes nas páginas 23 e 24.

LEI, S.; KODAGODA, S.; GAMINI, D. Environment classification and semantic grid map building based on laser range finder data. In: . [S.l.: s.n.], 2010. Citado na página 19.

LI, G. et al. Robot semantic mapping through wearable sensor-based human activity recognition. In: . [S.l.]: IEEE, 2012. p. 5228–5233. Citado 2 vezes nas páginas 21 e 22.

LIU, Z.; CHEN, D.; WICHERT, G. V. Online semantic exploration of indoor maps. In: . [S.l.]: IEEE, 2012. p. 4361–4366. Citado 2 vezes nas páginas 20 e 21.

MAIA, J. G. R. *Detecção e Reconhecimento de objetos usando Descritores Locais*. Tese (Doutorado) — Universidade Federal do Ceará, Curso de Ciência da Computação, Fortaleza, 2010. Citado na página 24.

MICROSOFT. *Kinect*. 2014. Citado na página 43.

MUJA, M.; LOWE, D. G. Fast approximate nearest neighbors with automatic algorithm configuration. In: *International Conference on Computer Vision Theory and Application VISSAPP'09)*. [S.l.]: INSTICC Press, 2009. p. 331–340. Citado na página 42.

MURPHY, R. *Introduction to AI robotics*. [S.l.]: MIT press, 2000. Citado 4 vezes nas páginas 14, 35, 36 e 37.

OTTONI, G. de L. *Planejamento de Trajetórias para Robôs Móveis*. 2000. Citado 4 vezes nas páginas 31, 32, 33 e 34.

PANGERCIC, D. et al. Semantic object maps for robotic housework - representation, acquisition and use. In: . [S.l.]: IEEE, 2012. p. 4644–4651. Citado na página 22.

RABINER, L. R. A tutorial on hidden markov models and selected applications in speech recognition. In: . [S.l.]: IEEE, 1989. p. 257–286. Citado na página 20.

R.N, K.; A., A.-A.; A., A.-J. Feature subset selection using differential evolution. In: . [S.l.: s.n.], 2008. Citado na página 19.

ROSENBLATT, J. K. *DAMN: A Distributed Architecture for Mobile Navegation*. Tese (Doutorado) — Carnegie Mellon University, Robotics Institute, Pittsburgh, 1997. Citado 2 vezes nas páginas 37 e 38.

SANTANA, A. M. *Localização e Mapeamento Simultâneos de Ambientes Planos Usando Visão Monocular e Representação Híbrida do Ambiente*. Tese (Doutorado) — Universidade Federal do Rio Grande do Norte, Programa de Pós-Graduação em Engenharia Elétrica e Computação, Natal, 2011. Citado 3 vezes nas páginas 14, 26 e 27.

SIEGWART, R.; NOURBAKHSH, I. R. *Introduction to autonomous mobile robots*. Massachusetts: The MIT Press, 2004. Citado 5 vezes nas páginas 30, 31, 32, 34 e 43.

SOUZA, A. et al. Principles of chiral perturbation theory. In: . [S.l.]: IEEE, 2013. p. 1–6. Citado na página 26.

SOUZA, A. A. de S. *Mapeamento com Sonar Usando Grade de Ocupação Baseado em Modelagem Probabilística*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, Programa de Pós-Graduação em Engenharia Elétrica e Computação, Natal, 2008. Citado 4 vezes nas páginas 19, 28, 29 e 30.

SOUZA, A. A. de S. *Mapeamento Robótico 2,5-D com Representação em Grade de Ocupação-Elevação*. Tese (Doutorado) — Universidade Federal do Rio Grande do Norte, Programa de Pós-Graduação em Engenharia Elétrica e Computação, Natal, 2012. Citado 3 vezes nas páginas 26, 27 e 30.

SOUZA, S. C. B. de. *Planejamento de Trajetória para um Robô Móvel com duas rodas Utilizando um Algoritmo A-Estrela Modeificado*. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, Curso de Engenharia Elétrica, Rio de Janeiro, 2008. Citado na página 34.

STAAB, S.; STUDER, R. *Handbook on Ontologies*. 2. ed. [S.l.]: Springer, 2010. Citado na página 25.

THEODORIDIS, S.; KOUTROUMBAS, K. *Pattern Recognition*. 4. ed. London: Elsevier Science, 2009. Citado na página 23.

THRUN, S. *Robotic mapping: a survey*. San Francisco: Morgan Kaufmann Publishers, 2003. Citado 2 vezes nas páginas 26 e 27.

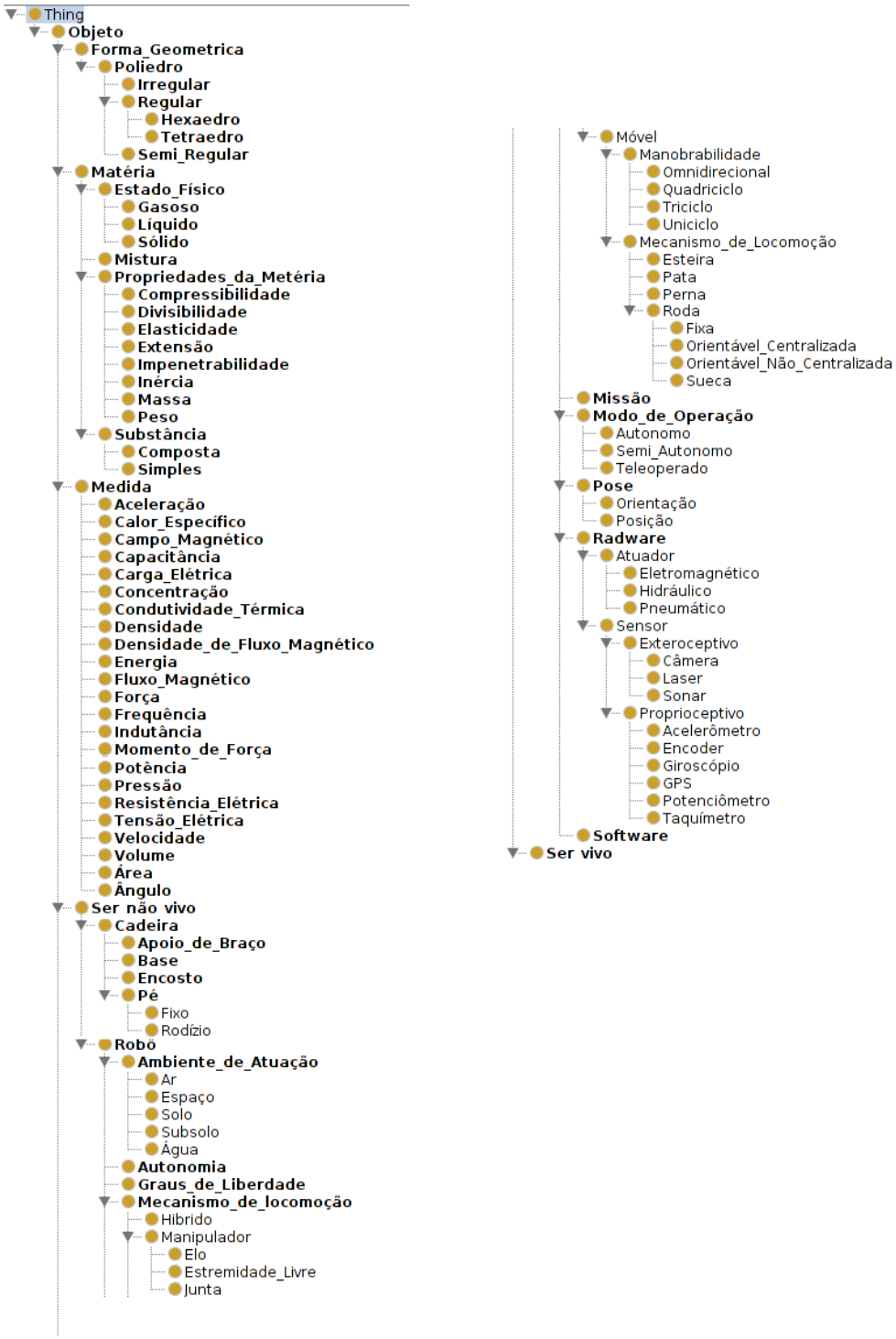
THRUN, S.; BURGARD, W.; FOX, D. *Probabilistic Robotic*. Cambridge: The MIT Press, 2005. Citado na página 29.

USCHOLD, M.; GRUNINGER, M. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, v. 11, p. 93–136, 1996. Citado na página 25.

WOLF, D. F.; SUKHATME, G. S. Semantic mapping using mobile robots. In: . [S.l.]: IEEE, 2008. p. 245–258. Citado na página 20.

7 ONTOLOGIA DO SISTEMA

Figura 26 – Ontologia geral do sistema



Fonte: Próprio autor.