DANIEL CHIN MIN WEI

Método de desvio de obstáculos aplicado em veículo autônomo

São Paulo
2015

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO
DEPARTAMENTO DE ENGENHARIA DE TRANSPORTES

DANIEL CHIN MIN WEI

Método de desvio de obstáculos aplicado em veículo autônomo

São Paulo
2015

DANIEL CHIN MIN WEI

Método de desvio de obstáculos aplicado em veículo autônomo

Dissertação apresentada à
Escola Politécnica da Universidade de São Paulo
para a obtenção do título de Mestre em Ciências

Área de Concentração:
Engenharia de Transportes
Ênfase: Informações Espaciais

Orientador: Prof. Dr.
Edvaldo Simões da Fonseca Júnior

São Paulo
2015

DANIEL CHIN MIN WEI

Método de desvio de obstáculos aplicado em veículo autônomo

Dissertação apresentada à
Escola Politécnica da Universidade de São Paulo
para a obtenção do título de Mestre em Ciências

Área de Concentração:
Engenharia de Transportes
Ênfase: Informações Espaciais

Orientador: Prof. Dr.
Edvaldo Simões da Fonseca Júnior

São Paulo
2015

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 20 de Julho de 2015

Daniel Chin Min Wei

Prof. Dr. Edvaldo Simões da Fonseca Júnior

FICHA CATALOGRÁFICA

Wei, Daniel Chin Min

Método de desvio de obstáculos aplicado em veículo autônomo / D.C.M. Wei. -- São Paulo, 2015.

100 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Transportes.

**1.Veículos guiados remotamente 2.Veículos especiais
3.Sensores ultrassonicos I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Transportes II.t.**

DEDICATÓRIA

À minha esposa Elisa e à minha filha Isabela cujo apoio foi essencial para eu chegar aqui. Sem vocês, eu nunca iria conseguir nada.

AGRADECIMENTOS

A Deus, que me deu saúde e disposição para viver e enfrentar todas as dificuldades.

À minha família, cujo apoio eu sempre tive e cujo amor e incentivo me fortalece em todos os momentos.

Aos meus professores, que tiveram a paciência e a generosidade de ensinar e repetir até eu entender. Um agradecimento especial ao meu orientador, Prof. Edvaldo Simões, cujo apoio foi sempre presente, mesmo quando era para repreender. Sem vocês, também não teria conseguido chegar aqui.

Aos meus colegas, tanto de mestrado quanto de profissão, pela compreensão nos meus momentos de dificuldade e apoio nos momentos de desânimo. Um agradecimento especial, ao Rodrigo Pissardini, que me ajudou em vários momentos de crise nesta caminhada.

Ao pessoal da USP, em especial à Patrícia, ao Edson, ao Arildo, ao Adalberto, à Elisa e ao João, pela ajuda nos momentos de necessidade.

À Continental, pelo apoio neste projeto de vida. Em especial às pessoas de Marcelo Fernandes, Ricardo Wetzel, Márcio Kopke e Marcello Pereira, que, em várias ocasiões, me deram ajuda crucial para este objetivo.

Ao Luis Felipe e ao Anderson Mori, predecessores que abriram caminho para esta jornada.

E a todos os amigos que, durante estes 3 anos, me incentivaram com palavras e atos.

Muito obrigado e que Deus os abençoe.

Um minuto na vida é um milagre que jamais se repete
(autoria desconhecida)

RESUMO

A operação de veículos autônomos necessita de meios para evitar colisões quando obstáculos não conhecidos previamente são interpostos em sua trajetória. Algoritmos para executar o desvio e sensores apropriados para a detecção destes obstáculos são essenciais para a operação destes veículos.

Esta dissertação apresenta estudos sobre quatro algoritmos de desvio de obstáculos e tecnologia de três tipos de sensores aplicáveis à operação de veículos autônomos. Após os estudos teóricos, um dos algoritmos foi testado para a comprovação da aplicabilidade ao veículo de teste.

A etapa experimental foi a realização de um programa, escrito em linguagem de programação Java, que aplicou o algoritmo Inseto 2 para o desvio de obstáculos em uma plataforma robótica (Robodeck) com o uso de sensores ultrassônicos embarcados na referida plataforma.

Os experimentos foram conduzidos em ambiente fechado (*indoor*), bidimensional e horizontal (plano), fazendo o Robodeck executar uma trajetória. Para os testes, obstáculos foram colocados para simular situações variadas e avaliar a eficácia do algoritmo nestas configurações de caminho.

O algoritmo executou o desvio dos obstáculos com sucesso e, quando havia solução para a trajetória, ela foi encontrada. Quando não havia solução, o algoritmo detectou esta situação e parou o veículo.

Palavras Chave: Veículo Autônomo, Desvio de Obstáculos, sensor ultrassônico.

ABSTRACT

The operation of autonomous vehicles need means to avoid collisions when unforeseen obstacles are posed in its trajectory. Algorithms to perform the deviation and suitable sensors for detecting these obstacles are essential for the operation of such vehicles.

This dissertation presents studies on four obstacle avoidance algorithms and technology of three types of sensors applicable to the operation of autonomous vehicles. After the theoretical studies, one of the algorithms has been tested for evidence of applicability to the test vehicle.

The experimental phase was the implementation of a program written in Java programming language, which applied the Insect 2 algorithm for obstacle avoidance in a robotic platform (Robodeck) using ultrasonic sensors embedded in the platform.

The experiments were conducted in a closed environment (indoor), two-dimensional and flat, making Robodeck perform a trajectory. For testing, obstacles were placed to simulate various situations and evaluating the algorithm efficacy for these path configurations.

The algorithm successfully performed the deviation of obstacles and, when there was a solution to the trajectory, it was found. When there was no solution, the algorithm has detected this situation and stopped the vehicle.

Keywords: Driverless Vehicle, Obstacle Avoidance, ultrasonic sensors.

LISTA DE ILUSTRAÇÕES

Figura 1 - Algoritmo do Inseto 1 com destino alcançável	9
Figura 2 - Algoritmo do Inseto 1 com destino inalcançável	10
Figura 3 - Algoritmo Inseto 2 com destino alcançável	12
Figura 4 - Algoritmo Inseto 2 com destino inalcançável	13
Figura 5 - Planejamento com Inseto Tangencial	14
Figura 6 - Movimento de Inseto Tangencial	15
Figura 7 - Inseto Tangencial com sensor de alcance nulo	16
Figura 8 - Inseto Tangencial com sensor de alcance finito.....	16
Figura 9 - Inseto Tangencial com sensor de alcance infinito.....	17
Figura 10 – Campo de Força Virtual - atração e repulsão.....	18
Figura 11 - Pontos de mínimo local.....	19
Figura 12 - Diretividade de sensor	20
Figura 13 - Interferência cruzada de sensores ultrassônicos	25
Figura 14 - Incerteza de um sensor ultrassônico.....	26
Figura 15 - Lidar Velodyne HDL-64E	29
Figura 16 - Diagrama de detecção do radar.....	32
Figura 17 - Dispositivos externos do Robodeck	35
Figura 18 - Movimentação das rodas do Robodeck	37
Figura 19 - Planejamento	41
Figura 20 - Medição de Odometria.....	46
Figura 21 - Diferença de distância Sensor Dianteiro e frente do Robodeck.....	53
Figura 22 - Medição de ângulo de abrangência	55
Figura 23 - Abrangência e Pontos Cegos do Sensor Ultrassônico.....	56
Figura 24 - Estados de desvio.....	58
Figura 25 - Dependência dos estados.....	59
Figura 26 - Fluxograma 1 (Sobre a trajetória original).....	61
Figura 27 - Fluxograma 2 (Contorno à esquerda)	62
Figura 28 - Fluxograma 3 (Contorno à direita)	63
Figura 29 - Desvio de Obstaculo Simples – trajetória esperada.....	65
Figura 30 - Desvio de Obstáculo Côncavo	66
Figura 31 - Objetivo Inalcancável	67
Figura 32 - Programa Lego Mindstorms.....	74

LISTA DE TABELAS

Tabela 1 - Velocidade do Som no ar	23
Tabela 2 - Características do Velodyne HDL-64E	29
Tabela 3 - Características do Radar ARS-30X.....	31
Tabela 4 - Medição de acurácia da odometria	47
Tabela 5 – Leituras da bússola do Robodeck	49
Tabela 6 – Ângulos dos pontos cardeais.....	50
Tabela 7 - Avaliação de erro angular	52
Tabela 8 - Medidas de distância com sensor ultrassônico	54
Tabela 9 - Acuracidade angular do Lego (valores em graus).....	76
Tabela 10 - Diferença de angulo com variação da velocidade (em graus).....	76
Tabela 11 - Afastamento no destino (em metros)	77
Tabela 12 - Formato da mensagem do MCR	80
Tabela 13 - Mensagem de Confirmação do MCR	81
Tabela 14 - Mensagem de Negação do MCR	82

LISTA DE ABREVIATURAS E SIGLAS

ARM	<i>Advanced RISC Machine</i>
ASCII	<i>American Standard Code for Information Interchange</i>
bps	Bits por segundo
DARPA	<i>Defense Advanced Research Projects Agency</i>
FOV	<i>Field Of View</i>
GPS	<i>Global Positioning System</i>
IDE	<i>Integrated Development Environment</i>
LASER	<i>Light Amplification by Stimulated Emission of Radiation</i>
LIDAR	<i>Light Detection and Ranging</i>
MAP	Módulo de Alta Performance
MCR	Módulo de Controle Robótico
NMEA	<i>National Marine Electronics Association</i>
RADAR	<i>Radio Detection and Ranging</i>
RISC	<i>Reduced Instruction Set Computer</i>
SDK	<i>Software Development Kit</i>
SLAM	<i>Simultaneous localization and Mapping</i>
TTL	<i>Transistor-Transistor-Logic</i>
USACE	<i>United States Army Corps of Engineering</i>
USB	<i>Universal Serial Bus</i>
VFH	<i>Vector Field Histogram</i>

SUMÁRIO

1	INTRODUÇÃO.....	1
1.1	OBJETIVOS.....	2
1.2	ESTRUTURA DO TRABALHO	3
2	REVISÃO DA LITERATURA.....	5
2.1	MÉTODOS DE PLANEJAMENTO E DESVIO	5
2.1.1	<i>Algoritmo do Inseto (Bug 1 and Bug 2 Algorithm).....</i>	<i>8</i>
2.1.2	<i>Algoritmo do Inseto Tangencial (Tangent Bug).....</i>	<i>13</i>
2.1.3	<i>Algoritmo de campo de força virtual (Virtual Force Field)</i>	<i>17</i>
2.2	SENSORES	19
2.2.1	<i>Sensores ultrassônicos.....</i>	<i>21</i>
2.2.2	<i>LIDAR (Light Detection and Ranging)</i>	<i>27</i>
2.2.3	<i>Radar (RADio Detection and Ranging)</i>	<i>30</i>
3	MATERIAIS E MÉTODOS	33
3.1	A PLATAFORMA PARA VALIDAÇÃO	33
3.1.1	<i>Alimentação e autonomia.....</i>	<i>34</i>
3.1.2	<i>Sensores disponíveis.....</i>	<i>35</i>
3.1.3	<i>Movimentação</i>	<i>36</i>
3.1.4	<i>Processamento embarcado.....</i>	<i>37</i>
3.2	PLANEJAMENTO DOS EXPERIMENTOS	38
3.2.1	<i>Sensores ultrassônicos.....</i>	<i>39</i>
3.2.2	<i>Algoritmo de desvio.....</i>	<i>40</i>
3.2.3	<i>Ambiente de desenvolvimento e programação.....</i>	<i>41</i>
3.2.4	<i>Posicionamento horizontal e orientação angular.....</i>	<i>42</i>
3.3	EXECUÇÃO DOS EXPERIMENTOS.....	44
3.3.1	<i>Movimentação em linha reta.....</i>	<i>45</i>
3.3.2	<i>Giro sobre o eixo.....</i>	<i>47</i>
3.3.3	<i>Sensor ultrassônico.....</i>	<i>53</i>
3.3.4	<i>Programa de teste.....</i>	<i>56</i>
3.4	EXPERIMENTOS COM O ALGORITMO	64
3.4.1	<i>Desvio de obstáculo Simples</i>	<i>65</i>
3.4.2	<i>Desvio de Obstáculo Côncavo</i>	<i>66</i>
3.4.3	<i>Objetivo inalcançável.....</i>	<i>67</i>
4	RESULTADOS OBTIDOS.....	68
5	CONSIDERAÇÕES FINAIS.....	70

5.1	MELHORIA NA EXECUÇÃO DOS DESVIOS	70
5.2	MELHORIAS NO ROBODECK.....	71
5.2.1	<i>Protocolo com recuperação de erros.</i>	71
5.2.2	<i>Cão de guarda e Indicação de presença</i>	72
5.2.3	<i>Interface de comunicação mais imune a ruídos.</i>	72
5.2.4	<i>Parada de Emergência</i>	73
APÊNDICE A - EXPERIMENTOS COM LEGO MINDSTORM EV3		74
A.1	ACURÁCIA ANGULAR	76
A.2	PERCURSO EM LINHA RETA.....	77
APÊNDICE B - FÓRMULAS MATEMÁTICAS E GEOMÉTRICAS		78
B.1	DISTÂNCIA ENTRE DOIS PONTOS	78
B.2	EQUAÇÃO DA RETA QUE UNE DOIS PONTOS EM ESPAÇO PLANO	78
B.3	DISTANCIA DE PONTO A RETA.....	79
APÊNDICE C - COMUNICAÇÃO COM O MCR		80
C.1	FORMATO DAS MENSAGENS	80
C.2	DIVERGÊNCIAS DE DOCUMENTAÇÃO	82
C.2.1	<i>Identificação dos sensores ultrassônicos</i>	82
C.2.2	<i>Unidade de medida da bússola</i>	82
C.2.3	<i>Identificação dos encoders do motor de tração</i>	83
APÊNDICE D - LISTA DE DOCUMENTOS EM DISCO.....		84
REFERÊNCIAS BIBLIOGRÁFICAS		85

1 INTRODUÇÃO

Os veículos terrestres tem sido objeto de esforços cada vez mais intensos para automação. Um dos grandes motivadores desta linha de pesquisa é a eliminação ou redução da participação do fator de maior influência nos acidentes: o ser humano. Estatísticas feitas, tais como em (DEPARTAMENTO AUTÔNOMO DE ESTRADAS DE RODAGEM - RS, 2010), indicam percentuais na ordem de 80% ou maiores para o fator humano como causa de acidentes.

Com isso, pode-se afirmar que um objetivo atual é o aumento na segurança dos veículos com a transferência de parte ou do todo das decisões para sistemas automatizados tais como os veículos autônomos.

Uma das principais missões de um veículo autônomo é executar um percurso de um ponto inicial até um ponto final com segurança e exatidão. Isso significa planejar e percorrer um caminho sem colidir com obstáculos que existam ou que venham a se interpor. Para este fim, diversos requisitos são essenciais (SIMMONS, 1996):

- A condução precisa ser tolerante ao ruído (erro) nos sensores. Não se trata de falha do sensor, mas de desvios e imprecisões inerentes à tecnologia.
- O algoritmo utilizado precisa ser eficiente ao ser executado no processador adotado. Um processo é inútil se só se completa depois de um tempo que invalida a missão do veículo.
- As características dinâmicas do veículo e da via (velocidade, capacidade de direção, frenagem) devem ser levadas em consideração. Planejar a passagem de um veículo por um local impraticável para ele é um contraexemplo para este caso.

Para cumprir uma missão, a navegação se divide em duas categorias:

- (i) global ou planejamento de rota, que procura um caminho ou direção a ser seguido, baseado em um conhecimento prévio (em geral, um mapa) do ambiente que será percorrido; e
- (ii) local ou desvio de obstáculos, cujo objetivo é levar o veículo de um

ponto até o seguinte, evitando e ultrapassando obstáculos no caminho.

Muitas vezes as duas funções se misturam, pois um algoritmo de planejamento de rota precisa evitar e desviar de obstáculos conhecidos.

Em um ambiente conhecido, o uso de mapas resolve de maneira satisfatória o problema de planejamento de rota. Entretanto, é muito comum o surgimento de obstáculos não mapeados previamente que só podem ser detectados quando se interpuserem no caminho do veículo. Nesta situação, o uso de sensores é essencial para se obter informações atualizadas do entorno.

Esta pesquisa se concentrou em métodos para:

- i) ler e interpretar dados de sensores; e
- ii) executar uma trajetória, evitando os obstáculos no caminho.

A parte prática aplicou estes métodos em uma plataforma de desenvolvimento robótico, e realizar um percurso com o veículo, desviando de obstáculos para chegar ao seu destino.

1.1 OBJETIVOS

Os objetivos desta dissertação são:

- Estudar algoritmos de desvio de obstáculos e sua aplicação para a tarefa de contornar obstáculos não planejados, relacionando suas vantagens, desvantagens, condições de aplicação e restrições;
- Estudar as características dos sensores que possam ser aplicados para este objetivo, identificando suas limitações e como elas afetam o método de planejamento;
- Escolher e aplicar um método e um sensor na plataforma Robodeck para permitir que se locomova sem colisões, evitando os obstáculos existentes em um espaço não infraestruturado e não conhecido previamente;
- Testar e validar os conceitos com uma realização aplicada na

plataforma Robodeck.

1.2 ESTRUTURA DO TRABALHO

Este primeiro capítulo faz um resumo do que é este documento de pesquisa, mostrando sua estrutura de apresentação para facilitar o acompanhamento.

O capítulo 2 é uma revisão da literatura e faz um levantamento de conceitos e algoritmos encontrados para o planejamento de rota e o desvio de obstáculos. Também faz um levantamento de tecnologias de sensores atualmente disponíveis, avaliando suas vantagens e limitações.

O capítulo 3 apresenta a aplicação prática do método e do sensor escolhidos: decisões tomadas, condições de contorno, recursos utilizados, experimentos executados e limitações encontradas. As tabelas com os resultados estão incluídas neste capítulo para facilitar a associação e a compreensão dos dados.

O capítulo 4 faz um fechamento sobre os resultados obtidos, os conhecimentos adquiridos e as limitações encontradas durante a execução dos testes.

O capítulo 5 apresenta propostas de continuidade do trabalho e considerações finais.

O apêndice A traz os dados obtidos durante a fase de testes usando o Lego Mindstorm. Embora estes dados não tenham sido usados para se chegar às conclusões do trabalho, foram importantes para preparar os testes.

O apêndice B traz algumas fórmulas matemáticas usadas na realização do algoritmo de desvio de obstáculos, principalmente para a avaliação das distâncias entre o veículo e o ponto de destino.

O apêndice C mostra o formato das mensagens trocadas entre o Módulo de Controle Robótico (MCR) do Robodeck e o Módulo de Alto Desempenho (MAP). Este protocolo foi o utilizado para fazer o Robodeck executar o percurso planejado e desviar dos obstáculos.

O apêndice D traz uma lista de documentos que não estão incluídos no corpo desta dissertação, mas que estão no CD/DVD que a acompanha.

2 REVISÃO DA LITERATURA

2.1 MÉTODOS DE PLANEJAMENTO E DESVIO

Segundo (CHOSSET, et al., 2005), a movimentação de um veículo autônomo pode ser dividida em 4 tarefas básicas: navegação (ou condução), cobertura, localização e mapeamento.

- A tarefa de condução consiste em encontrar um movimento factível para o veículo, com ausência de colisões, partindo de uma posição de origem e indo até uma posição de destino.
- A tarefa de cobertura é garantir a varredura dos sensores por todos os pontos do espaço planejado.
- A tarefa de localização é a capacidade de usar dados de sensores para determinar a posição do veículo e inseri-lo em um mapa ou um planejamento conhecido previamente.
- O mapeamento é a tarefa de usar informações dos sensores em um ambiente não conhecido para construir um conjunto de informações (mapa) que seja útil para as 3 tarefas anteriores.

De maneira análoga, em (BEKEY, 2005), no início do capítulo 14, existe um conjunto de perguntas que precisam ser respondidas para um movimento correto de um robô:

- “Como chego lá?” envolve a determinação da existência de obstáculos e caminhos irregulares que o veículo precisa vencer para atingir o seu destino da melhor maneira possível. O termo “melhor” implica em uma avaliação de “custo” do trajeto, podendo este custo ser em tempo, em distância, em energia ou qualquer outra métrica que seja fundamental para uma dada missão.
- “Onde estou?” se relaciona com o problema de localização. Em (BEKEY, 2005) é feita uma analogia divertida com o “problema de um robô raptado”. Um robô é raptado, vendado e deixado em uma nova posição. Como ele procederá para saber a sua atual posição em

relação a coordenadas conhecidas? Que sensores e que informações ele precisa para isso?

- “Onde eu estive?” é a terceira pergunta e envolve mapear locais desconhecidos em que um mapa prévio ainda não existe. Juntamente com a pergunta anterior, cria o acrônimo, SLAM (*Simultaneous Localization and Mapping*), que define o uso dos sensores durante o percurso para se localizar e criar um mapa de onde passou.

Para a movimentação de um veículo, duas ações similares são necessárias (FEITEN, et al., 1994 (vol 3)). A primeira, denominada planejamento de rota, leva em conta a posição atual e a posição alvo, gerando uma sequência de pontos de destino intermediários que devem ser percorridos para se chegar ao destino desejado. Já a segunda, denominada de manobra local, visa conduzir o veículo de um ponto intermediário até o seguinte, evitando os obstáculos que impeçam a trajetória desejada.

Em um plano horizontal sem obstáculos (ambiente ideal), o percurso de um veículo autônomo envolveria apenas a posição de origem e a posição de destino. Com estas duas informações, o responsável por planejar a rota traçaria uma linha reta e bastaria ao veículo percorrer este caminho para atingir o destino.

Em um ambiente real, diversos fatores mudam estas condições:

- Podem existir obstáculos previstos no caminho, que exijam uma rota de contorno. Estes obstáculos podem ser previamente conhecidos pelo planejador da rota com o uso de um mapa do ambiente.
- Obstáculos imprevistos podem se interpor no caminho do veículo durante o percurso, exigindo que uma ação seja executada para evitar e contornar o obstáculo.
- O destino pode ser inatingível por causa de um obstáculo intransponível ou pela simples inexistência de rota viável dentro das condições estáticas e dinâmicas do veículo.

A descoberta dos obstáculos, a sua identificação quanto ao grau de impedimento para a atual trajetória e a determinação de novas direções são o objetivo dos métodos de desvio de obstáculos.

Três conceitos utilizados com frequência, descritos em (CHOSSET, et al., 2005) e (BEKEY, 2005) são:

- Caminho (*path*) – conjunto de posições no espaço que será percorrido pelo veículo para cumprir a sua missão.
- Trajetória (*trajectory*) – conjunto de pares caminho/velocidade utilizado pelo veículo para ir de um ponto a outro dentro do caminho planejado.
- Planejamento – processo de construir uma trajetória livre de colisões entre um ponto de partida (estado inicial) e um ponto de chegada (destino ou estado final).

As propriedades de um veículo limitam as suas possibilidades de operação. As principais são os seus graus de liberdade (ângulo máximo de esterção, direções de movimentação possíveis), as suas características estáticas (formato, tamanho, peso) e suas características dinâmicas (limites de aceleração, velocidade e frenagem).

Alguns algoritmos levam em consideração um subconjunto destes fatores para o cálculo de trajetória. Outros consideram o veículo como um objeto pontual, para facilitar o manuseio das expressões matemáticas. Este ponto pode ser o seu centro de massa ou o centro geométrico. (SIMMONS, 1996) (FOX, et al., 1997).

Uma estratégia simples de expandir o algoritmo para levar em conta as dimensões do veículo é o de considerar a sua forma como circular, de diâmetro igual à maior dimensão do veículo acrescido da tolerância inerente ao sensor utilizado.

Um conceito referente aos graus de liberdade de movimento do veículo é a holonomicidade (tradução livre do termo em inglês *holonomicity*) ou holonomia. O objeto é dito holonômico (ou holônomo) se, dentro do ambiente em que ele se propõe a operar, não há restrição para a sua movimentação em nenhum dos graus de liberdade possíveis. Em outras palavras, a quantidade de operações de movimentação é igual ou maior do que o número de graus de liberdade aplicáveis ao objeto.

Para entender melhor este conceito, considere-se uma posição inicial P_0 em um instante T_n e o conjunto de pontos adjacentes a ele que o veículo deveria ser capaz de alcançar no instante T_{n+1} . Se, a partir de P_0 , o veículo é capaz de atingir qualquer um destes pontos adjacentes no instante T_{n+1} , ele é dito holonômico.

No caso do veículo terrestre, o ambiente de operação é considerado como um espaço 2-D e o robô é classificado como holonômico se ele puder alterar seu movimento em qualquer direção (liberdade de 360 graus). Quando isso não é factível, o veículo é dito não-holonômico. Um automóvel, por exemplo, não é

holonômico pois um movimento impossível para ele é andar lateralmente.

O número de variáveis envolvidas em um planejamento de trajetória faz com que a sua conclusão (termo livremente traduzido do inglês *completeness*) possa ser possível, totalmente impossível ou impossível dentro de um limite de recursos definidos (ex: tempo ou quantidade de energia).

Um planejador pode ser *offline*, se a trajetória é feita *a priori*, usando um modelo do ambiente, ou *online*, se for feita durante a execução da trajetória, usando dados obtidos de sensores (CHOSET, et al., 2005). Na prática, os veículos unem um planejamento *offline* (mapas) com um planejamento *online* (desvio de obstáculos).

A seguir serão apresentadas algumas estratégias e algoritmos para evitar obstáculos. O foco foi dado à aplicação dos algoritmos e nos seus pontos fortes e fracos.

2.1.1 Algoritmo do Inseto (*Bug 1 and Bug 2 Algorithm*)

É uma realização do instinto natural, apresentado por insetos como as formigas, de ir em direção ao objetivo até encontrar um obstáculo, contorná-lo e retomar o curso. Para fazer isso, o movimento do veículo é dividido em duas fases: movimento em direção ao alvo e movimento de contorno de borda (*boundary following*). O pressuposto é de que o veículo seja dotado de sensores que possam detectar o limite de um obstáculo e tenha algum meio para determinar a distância entre a sua posição corrente e o destino pretendido. É um algoritmo que garante o sucesso em chegar ao destino, desde que ele seja alcançável (CHOSET, et al., 2005).

A descrição do algoritmo é a seguinte:

1. Mover-se em direção ao destino até alcançá-lo ou encontrar um obstáculo antes.
2. Se um obstáculo for encontrado na trajetória planejada, inicia-se um movimento de contorno. Durante o contorno do obstáculo, duas informações são constantemente obtidas:
 - a. Distância entre o ponto no contorno e o ponto de destino;
 - b. Se existe caminho desobstruído até o destino (o obstáculo não se

interpõe entre a posição corrente e o destino).

3. Quando tiver feito o contorno completo, o veículo estará de volta ao ponto de início onde o obstáculo foi detectado. Como as distâncias de cada ponto do caminho de contorno até o destino foram calculadas durante o movimento de contorno, haverá duas possibilidades:
 - a. Existem pontos a partir do qual é possível seguir em direção ao destino sem ser obstruído pelo obstáculo. Neste caso, o algoritmo identifica, dentro deste subconjunto de pontos desobstruídos, aquele que tenha a menor distância até o destino e retoma o movimento em direção ao destino a partir daquele ponto
 - b. Não há ponto desobstruído que permita um caminho para o destino. Neste caso, o destino é inalcançável e o algoritmo termina sem sucesso.

As figuras abaixo mostram o movimento usando este algoritmo.

Figura 1 - Algoritmo do Inseto 1 com destino alcançável

Fonte: (CHOSET, et al., 2005)

A Figura 1 (CHOSSET, et al., 2005) apresenta uma situação com dois obstáculos e com o destino alcançável. Nesta configuração o veículo chega ao obstáculo no ponto $qH1$ e inicia o contorno. Após completar o contorno e retornar a $qH1$, o algoritmo identifica $qL1$ como o ponto desobstruído mais próximo do destino. Assim, ele se move até $qL1$ e retoma o caminho até encontrar um novo obstáculo em $qH2$.

Novamente o veículo faz o contorno e, após retornar a $qH2$, identifica $qL2$ como o ponto desobstruído mais próximo do destino. Ele se move até $qL2$ e, a partir de lá, retoma o caminho em direção ao destino.

A Figura 2 (CHOSSET, et al., 2005) mostra uma situação em que o destino não é alcançável. O veículo se move em direção ao destino até encontrar o obstáculo em $qH1$. Então ele inicia o contorno até retornar a $qH1$. Neste percurso, o algoritmo identificou $qL1$ como o ponto mais próximo do destino mas ele é inalcançável a partir de lá. Como não encontrou nenhum ponto durante o percurso que fosse utilizável para se atingir o destino, o algoritmo termina sem sucesso possível.

Note-se que este algoritmo faz com que o veículo percorra o caminho inteiro pelo menos uma vez antes de tomar uma decisão.

Figura 2 - Algoritmo do Inseto 1 com destino inalcançável

Fonte: (CHOSSET, et al., 2005)

Uma variante do algoritmo anterior, chamada de Inseto 2, foi criada para otimizar o processo. A diferença é o ponto de decisão a partir do qual o veículo retoma o caminho em direção ao destino (ponto de relargada). No algoritmo do Inseto 1, o ponto de relargada de um obstáculo é aquele que está desobstruído e que oferece a menor distância até o destino. Para determinar isso, o veículo tem que contornar o obstáculo pelo menos uma vez para mapear todas as pontos viáveis para uma relargada.

A proposta do Inseto 2 é determinar um caminho (linha reta) entre o ponto de origem e o ponto de destino. Quando um obstáculo é encontrado, o mesmo movimento de contorno é iniciado. Mas, assim que encontrar um ponto no caminho de contorno que seja desobstruído e que pertença à linha de movimentação inicial, o veículo abandona o contorno e retoma o caminho inicial.

As figuras abaixo mostram este movimento nas mesmas situações do algoritmo anterior.

A Figura 3 (CHOSSET, et al., 2005) mostra o comportamento do algoritmo do Inseto 2 em um caso de destino alcançável. O veículo se move sobre a linha que liga o ponto $q_{inicial}$ ao ponto $q_{destino}$. Quando atinge o obstáculo em $qH1$, ele inicia o contorno até encontrar novamente a linha original em $qL1$. Então ele retoma o movimento sobre a linha planejada inicialmente até encontrar o novo obstáculo em $qH2$. Novamente ele faz o contorno até atingir o ponto $qL2$ onde retoma o curso original e atinge o destino.

Figura 3 - Algoritmo Inseto 2 com destino alcançável

Fonte: (CHOSSET, et al., 2005)

A Figura 4 (CHOSSET, et al., 2005) apresenta o comportamento com o destino inalcançável. O veículo anda até encontrar o obstáculo em q_{H1} . Então ele faz o contorno até retornar ao mesmo ponto, sem ter encontrado um caminho viável para o destino. A conclusão é que o destino é inalcançável e o algoritmo se encerra.

Figura 4 - Algoritmo Inseto 2 com destino inalcançável

Fonte: (CHOSET, et al., 2005)

2.1.2 Algoritmo do Inseto Tangencial (*Tangent Bug*)

É mais uma evolução do algoritmo do inseto e conta com uma melhoria do tipo de sensor. Nos anteriores, um sensor de contato ou de curtíssimo alcance bastava para fazer o algoritmo funcionar. Neste aqui, o alcance do sensor passa a ser utilizado para melhorar a eficiência do algoritmo.

O pressuposto para este algoritmo é um sensor com um alcance limitado R e uma orientação de 360° com resolução alta (idealmente infinita). Os obstáculos são detectados quando entram no alcance do sensor e seu contorno é mapeado para conhecer seus limites.

O objetivo do algoritmo é manter a direção de movimento apontado para o ponto mais extremo do contorno conhecido do obstáculo e que apresente, ao mesmo tempo, a menor distância para o destino. Em teoria, o veículo conseguiria passar ao lado do obstáculo (tangenciá-lo) sem ter que contorná-lo, obtendo assim um caminho ótimo.

Figura 5 - Planejamento com Inseto Tangencial

Fonte: (CHOSSET, et al., 2005)

A Figura 5 (CHOSSET, et al., 2005) mostra duas situações: Na primeira (Figura 5-a), o veículo detecta o obstáculo WO_1 mas ainda não "vê" WO_2 . Os seus sensores detectam os pontos O_1 e O_2 como sendo os limites visíveis do obstáculo.

Como o ponto O_2 é mais próximo do destino do que o ponto O_1 , o algoritmo planeja o caminho $x \rightarrow O_2 \rightarrow q_{destino}$ para tangenciar o primeiro obstáculo e alcançar o objetivo.

Nota-se que este caminho faria o veículo passar por dentro de WO_2 mas deverá ser replanejada assim que WO_2 for detectado pelos sensores.

A segunda figura (Figura 5-b) assume que o sensor do veículo possui um alcance maior e "enxerga" WO_2 . Assim, ele determina que o ponto O_4 é mais eficiente e planeja a rota $x \rightarrow O_4 \rightarrow q_{destino}$ sem fazer o contorno de WO_1 .

Uma observação interessante na segunda figura: O caminho entre o ponto O_4 e o destino passa dentro do obstáculo WO_2 porque, no momento em que detecta o ponto limite O_4 , o veículo não tem a informação sobre o que existe após ele. À medida que se aproximar do obstáculo, o sensor deverá indicar ao veículo que o ponto a ser tangenciado é ligeiramente diferente.

A Figura 6 (CHOSSET, et al., 2005) mostra um veículo com um sensor com alcance finito (simbolizado pelo círculo pontilhado) se aproximando de um obstáculo. Enquanto o obstáculo não está ao alcance do sensor (momento $t=1$), o curso

seguido é uma linha reta para o destino.

No momento $t=2$ o sensor já detectou o obstáculo e determinou 2 pontos de tangência: um à esquerda e outro à direita. Na avaliação de distâncias, o ponto à direita está mais próximo do destino do que o ponto à esquerda. Por isso, o curso é modificado para à direita.

Este movimento continua em $t=3$ e $t=4$, à medida que o sensor determina o contorno do obstáculo. Em cada momento, um novo ponto mais próximo do destino é definido e o curso é mudado para este novo ponto.

Figura 6 - Movimento de Inseto Tangencial

Fonte: (CHOSSET, et al., 2005)

O alcance e o ângulo de definição do sensor tem grande influência neste algoritmo. Olhando a Figura 6 no instante $t=2$, nota-se que o veículo começa a manobrar pois detectou um obstáculo e identificou dois pontos com distâncias diferentes para o destino. Se o sensor tiver um ângulo de definição muito grande (baixa diretividade), o robô pode não ser capaz de diferenciar estes 2 pontos.

A Figura 7 (CHOSSET, et al., 2005) mostra o comportamento de um veículo com este algoritmo e dotado de um sensor de alcance zero ou muito pequeno. Nota-se a semelhança do comportamento com o dos algoritmos de Inseto 1 e 2.

Figura 7 - Inseto Tangencial com sensor de alcance nulo

Fonte: (CHOSSET, et al., 2005)

A Figura 8 (CHOSSET, et al., 2005) apresenta os mesmos obstáculos da Figura 7 mas com o veículo dotado de um sensor com alcance maior, embora ainda limitado. Note-se a suavização das mudanças de rota e a melhoria obtida.

Figura 8 - Inseto Tangencial com sensor de alcance finito

Fonte: (CHOSSET, et al., 2005)

A Figura 9 (CHOSSET, et al., 2005) idealiza um veículo com sensor de alcance muito maior. No limite hipotético de um sensor de alcance infinito, a trajetória se aproxima ao que se consegue baseado em mapas, em que os obstáculos são previamente conhecidos.

Figura 9 - Inseto Tangencial com sensor de alcance infinito

Fonte: (CHOSSET, et al., 2005)

2.1.3 Algoritmo de campo de força virtual (*Virtual Force Field*)

Neste algoritmo, apresentado por (BORENSTEIN & KOREN, 1989) o destino do veículo é representado por uma força de atração e os obstáculos são representados por forças de repulsão. O seu princípio de funcionamento junta dois métodos: função de potencial (*potential function*) para planejamento de trajetória e grade de certeza (*certainty grid*) para avaliação das informações dos sensores.

O método de função potencial foi criado inicialmente para o planejamento de rotas. Para isso, define potenciais positivos para o veículo e para os obstáculos enquanto que o destino recebe um potencial negativo.

Os potenciais de mesmo sinal se repelem e os de sinais opostos se atraem. A resultante das forças faz com que o veículo navegue sendo atraído pelo destino e repelido pelos obstáculos.

O método de grade de certeza é um método para mapear o ambiente baseado em informações de sensores de distância. Para cada célula onde possa estar um obstáculo, é atribuído um valor que corresponde ao grau de certeza de existência deste obstáculo. Este grau de certeza reflete a confiabilidade da informação do sensor.

O algoritmo de *Virtual Force Field*, ilustrado pela Figura 10 (BORENSTEIN & KOREN, 1989), junta estes dois métodos atribuindo a cada célula um valor de

repulsão:

- proporcional à certeza da informação de presença do obstáculo;
- inversamente proporcional ao quadrado da distância do obstáculo ao veículo.

A dimensão da grade é função da taxa de amostragem dos sensores e da velocidade máxima do veículo. O tamanho de cada célula deve ser tal que seu lado seja menor do que a distância percorrida pelo veículo entre duas amostragens.

Matematicamente falando:

$$\Delta s > T * V_{max} \text{ onde}$$

Δs é o lado da célula
 T é o período entre 2 amostragens
 V_{max} é a velocidade máxima do veículo

Figura 10 – Campo de Força Virtual - atração e repulsão

Fonte: (BORENSTEIN & KOREN, 1989)

Os algoritmos de potencial são vulneráveis a uma situação de aprisionamento (*trap state*) em que o veículo atinge um ponto de potencial mínimo local sem ter atingido o destino, que tem um potencial mínimo global. Neste ponto de

mínimo local, há um equilíbrio entre as forças de repulsão e atração que impede o algoritmo de resolver o impasse.

A Figura 11 (CHOSSET, et al., 2005) mostra duas situações que exemplificam este caso, em que o veículo percorre um caminho até um ponto em que existe o equilíbrio.

Figura 11 - Pontos de mínimo local

Fonte: (CHOSSET, et al., 2005)

Para sair desta situação, a função de condução e detecção de obstáculos precisa prever um comportamento alternativo. Em (BORENSTEIN & KOREN, 1989) sugere-se que, quando se atinge uma situação destas, se alterne para um algoritmo de “seguimento de parede” (*Wall-following*) que é o mesmo que o algoritmo do inseto. Com isso, é possível contornar o ponto de mínima e prosseguir com o percurso pretendido.

2.2 SENSORES

Para a detecção de obstáculos, sensores são essenciais para a execução destas tarefas. O veículo Boss, da Universidade Carnegie Mellon, por exemplo, era equipado com um conjunto de 17 sensores entre Lidar, Radar e GPS (URMSON, et al., 2009) que fornecia um campo de visão de 360° em torno do veículo.

Em várias situações as regiões de ação dos diferentes tipos de sensores se

sobrepunham, fornecendo informações redundantes ou complementares. O principal sensor do Boss era um Lidar da Velodyne, que é mostrado na seção 2.2.2, mas dispunha também de três radares da Continental.

Os algoritmos descritos contam com informações que precisam ser extraídas de sensores cujo comportamento esperado é o ideal. Por exemplo, o “inseto tangencial” espera contar com um sensor de longa distância com alta diretividade e um erro de distância baixo para poder operar com sucesso.

A diretividade do sensor é um fator importante para a localização acurada de obstáculos. O sinal emitido por um sensor se abre em um lóbulo e a detecção de um objeto se dará dentro dos ângulos de abertura deste lóbulo. Desta maneira, um objeto será detectado se estiver em qualquer posição dentro da região do espaço abrangida pelo lóbulo de irradiação do sensor.

Quanto menor o ângulo de abertura, maior é a diretividade de um sensor e menor é a incerteza da localização de um obstáculo. Um sensor a laser, devido ao baixo espalhamento da luz laser, tem um ângulo de abertura de $0,5^\circ$ enquanto um sensor ultrassônico tem um ângulo de abertura de 30° . Portanto, dizemos que o sensor a laser tem uma diretividade maior do que o sensor ultrassônico.

Nos algoritmos de inseto, quanto maior for a diretividade angular, melhor a definição dos pontos limites dos obstáculos. A importância de identificar estes limites pode ser demonstrada pela Figura 12 que mostra a mesma situação de obstáculos com sensores de diretividade diferentes.

O sensor de baixa diretividade vai relatar um obstáculo único, como se fosse uma parede sólida, e o caminho estará obstruído. Já o sensor de alta diretividade vai relatar um vão e o veículo poderá atravessar.

Figura 12 - Diretividade de sensor

No mundo real, os sensores estão sujeitos a diversos tipos de erros tais como não-linearidades, ruídos elétricos ou térmicos, espalhamento de sinal, reflexões e outros. Tudo isso precisa ser considerado na avaliação da confiabilidade da informação recebida dos sensores.

O compromisso entre custo e desempenho também é um fator decisivo para a utilização dos sensores. Na plataforma Robodeck, não há sensores de longa distância, considerando-se longas distâncias a partir de 10 m. O principal sensor de média distância disponível é um sensor ultrassônico cujo ângulo de abertura é de $22,5^\circ$ (45° de abertura de cone) e serve basicamente para a detecção de existência de obstáculo. Sua utilização para a identificação de obstáculos com o fim de mapeamento é muito precária devido ao seu alcance limitado a poucos metros e a sua baixa diretividade.

Por outro lado, o uso de sensores com maior alcance pode produzir um caminho planejado com maior eficiência, como foi visto na seção 2.1.2.

Neste capítulo alguns sensores são apresentados e suas características técnicas são analisadas quanto à sua utilização para identificar e evitar obstáculos. O foco será dado aos sensores de alcance médio uma vez que o objetivo deste trabalho é o desvio de obstáculos.

2.2.1 Sensores ultrassônicos

Sensores ultrassônicos são comercialmente disponíveis e de uso mais generalizado atualmente. Seu funcionamento se baseia no conhecimento da velocidade média de uma onda sonora e na recepção da reflexão de uma onda transmitida pelo sensor, sendo usados em produtos de mercado tais como detectores de proximidade e sensores de estacionamento. Uma referência sobre a teoria de base dos sensores ultrassônicos pode ser encontrada em (KINSLER, et al., 2000).

O sensor emite um feixe de ondas sonoras e aguarda o retorno do eco desta onda. O tempo de retorno indica a distância do objeto detectado. Uma condição simplificadora no caso de veículos terrestres é a ausência de mudança de meio, com

a onda sonora transitando apenas pelo ar, desprezando-se mudanças de temperatura e de pressão no meio de propagação da onda sonora.

A velocidade do som pode ser calculada pela expressão (KINSLER, et al., 2000)

$$v = \sqrt{\gamma * P \div D}$$

onde v = velocidade do som

γ = razão entre calores específicos do meio,
sob pressão constante e sob volume
constante

P = Pressão do meio (no caso, atmosférica)

D = densidade do meio (kg/m^3)

Assumindo o comportamento de um gás perfeito, a velocidade do som, a uma temperatura de 0°C e sob uma pressão atmosférica ao nível do mar, é obtida com os seguintes valores (KINSLER, et al., 2000):

$$\gamma = 1,402$$

$$P = 1,01325 \times 10^5 \text{ Pa}$$

$$D = 1,293 \text{ kg/m}^3$$

$$v_0 = \sqrt{\frac{1,402 * 1,01325 \times 10^5}{1,293}} = 331,5 \text{ m/s}$$

A razão P/D do ar livre, sob condição de temperatura constante, é pouco afetada pela variação de pressão, pois uma variação em um sentido da pressão é acompanhada por uma variação da densidade no mesmo sentido. Portanto, a velocidade do som é uma função majoritariamente dependente da temperatura, podendo ser expressa (KINSLER, et al., 2000) por:

$$v(T_k) = v_0 \sqrt{\frac{T_k}{273}}$$

onde T_k é a temperatura do ar, em $^\circ\text{K}$.

A Tabela 1 mostra a velocidade do som, no ar, calculado para algumas temperaturas,

Tabela 1 - Velocidade do Som no ar

Temperatura em °C	Velocidade do som em m/s
+35	352,1
+30	349.2
+25	346.3
+20	343.4
+15	340.4
+10	337.5
+5	334.5
0	331.5
-5	328.4
-10	325.3
-15	322,3
-20	319,1
-25	315,9

As principais características para a análise dos sensores ultrassônicos são (EVERETT, 1989):

- Alcance mínimo e alcance máximo: a velocidade máxima que pode ser imposta ao veículo e o tipo de obstáculos que pode superar (ex: corredores estreitos) dependem deste fator.

Por exemplo, com um sensor dotado de um alcance maior, o robô pode rodar com uma velocidade maior. Ao detectar um obstáculo a uma distância maior, ele tem mais tempo para reduzir a velocidade ou manobrar para desviar e evitar uma colisão.

- Faixa de frequência: A frequência e o tipo de onda usada pelo sensor influenciam no ambiente, no tipo e no tamanho de objeto que pode ser detectado.

Sensores ultrassônicos que usam frequências mais altas podem detectar objetos menores. Frequências mais baixas são mais atenuadas, e menos difratadas, pelo meio de propagação.

- Acurácia e resolução angular: A exatidão de detecção e o tamanho dos obstáculos dependem deste fator.

Os pontos mais sensíveis deste sensor são a sua dependência do tipo da superfície do alvo (rugosidade, grau de absorção e ângulo de inclinação), do nível de ruído ambiente, de sua suscetibilidade à interferência cruzada (*cross talk*) e do seu

ângulo de abertura.

Para sensores ultrassônicos, irregularidades (ou até mesmo objetos) cujas dimensões sejam da ordem de grandeza do comprimento da onda podem afetar sensivelmente o nível de reflexão da onda emitida pelo sensor. A porosidade do material do alvo, partículas suspensas no ar e umidade são fatores que afetam a sua eficiência. Os principais efeitos são a atenuação e o espalhamento, ambos reduzindo o desempenho do sensor.

Outro fator que pode interferir na medida da resposta da onda refletida é a inclinação da área frontal em relação ao eixo incidente. Estas diluições se apresentam como um sinal refletido de menor intensidade, o que pode tornar um obstáculo invisível. Uma análise sobre os efeitos de incidência da onda sonora em diversas condições pode ser encontrada em (KINSLER, et al., 2000).

Para compensar esta perda de sensibilidade, uma alternativa é aumentar a amplificação no receptor, mas traz, como consequência, um aumento na inserção de ruídos, causando falsas detecções ou erros de medidas de distâncias.

O efeito de interferência cruzada é outro fator negativo. Pode ser causado por

- i) emissão de sensores similares de outros veículos; ou
- ii) ser resultado de reflexão do sinal do próprio veículo.

A primeira causa pode ser compensada através de algoritmos tais como alternar o período de transmissão (BORENSTEIN & KOREN, 1995) para permitir diferenciar sinais originados por fontes distintas. Para isso, é necessário ter controle total sobre o processo de geração dos pulsos no emissor.

A segunda causa é mais complicada de resolver. A Figura 13 (BORENSTEIN & KOREN, 1995) mostra duas situações de interferência cruzada. Nela, os sensores marcados com “X” são os emissores e os sensores marcados com “Y” são os receptores que vão receber a sinal de interferência cruzada.

Na Figura 13-a, o sinal emitido por X chega até a parede e retorna para X mas chega também nos dois Ys. Se não houver um meio de diferenciar o sinal emitido por X do sinal emitido por Y, este pode interpretar o sinal recebido como um obstáculo na visada angular de Y.

Já na Figura 13-b, o sensor Y recebe o sinal emitido por X após três reflexões, fazendo parecer que existe um obstáculo em uma posição e orientação diferente da real.

Figura 13 - Interferência cruzada de sensores ultrassônicos

Fonte: (BORENSTEIN & KOREN, 1995)

O grau de abertura do sinal emitido faz com que a acurácia na identificação da posição se deteriore. O sinal ultrassônico se abre em um cone e é impossível dizer com exatidão a posição do alvo detectado, no caso de uma detecção positiva. O que se pode afirmar é que algum objeto existe e foi detectado dentro de uma região de incerteza.

A Figura 14 (BORENSTEIN & KOREN, 1991) mostra um objeto detectado que tem uma pequena parte dentro do segmento de coroa circular A, delimitado pelo cone de abertura do sensor, pela distância medida pelo eco e pela resolução da medida. A detecção deste objeto pelo sensor traz a informação de que existe

alguma obstrução dentro da área A, mas ela pode estar em qualquer parte da área sombreada.

Figura 14 - Incerteza de um sensor ultrassônico

Fonte: (BORENSTEIN & KOREN, 1991)

Alguns dos métodos estudados procuram compensar a imprecisão dos sensores ultrassônicos. Por exemplo, o *Vector Field Histogram* (BORENSTEIN & KOREN, 1991), faz leituras repetidas nos diversos sensores ultrassônicos e considera um obstáculo detectado apenas na direção do eixo de cada sensor.

A suposição, neste caso, é de que repetidas medições com o veículo em movimento irão permitir a eliminação dos falsos positivos, que irão aparecer em um sensor, mas não aparecerão no sensor vizinho. As leituras repetidas são usadas para construir um histograma de confiabilidade sobre a existência de um obstáculo.

Já em (BORENSTEIN & KOREN, 1995) propõe a geração de padrões de sinal diferentes para cada um dos sensores, de maneira a poder diferenciar se o eco

recebido é de um ou de outro sensor. Também propõe leituras repetidas e filtragem dos valores para eliminar ruídos não repetitivos e de média nula.

O disparo de um conjunto de pulsos por um emissor de ultrassom gera uma vibração que não desaparece imediatamente após o fim da excitação (MÁGORI & WALKER, 1987). Existe uma janela de tempo em que não é possível se determinar se o eco detectado é um resultado da vibração residual ou se é um retorno de um objeto muito próximo. Por esta razão existe uma distância mínima para a detecção de obstáculos por um sensor de ultrassom. Esta distância depende da frequência de operação do sensor e de detalhes de construção e amortecimento.

O tratamento dos dados recebidos do sensor ultrassônico é um dos obstáculos que precisará ser vencido neste trabalho. O uso de filtros para a atenuação dos erros introduzidos e o tratamento probabilístico são as principais ferramentas para este fim.

Os sensores ultrassônicos podem ser arranjados em redes para compor efeitos tais como geração de lóbulos de irradiação de ângulos diferentes ou a rotação da direção do lóbulo através de disparos sequenciados no tempo (KINSLER, et al., 2000).

2.2.2 LIDAR (Light Detection and Ranging)

O LIDAR é um sensor baseado na emissão de um ou mais feixes laser, pela recepção de sua reflexão e no conhecimento da velocidade de propagação da luz. O tempo de ida e volta é a grandeza física que é usada para medir a distância entre o sensor e o alvo. O alcance está na ordem de algumas dezenas de metros e é dependente da frequência do laser emitido.

O sensor dispara o feixe circularmente, recolhendo os ecos recebidos. Devido à alta frequência e à alta velocidade da luz, a janela de guarda entre um pulso e o seguinte, que no sensor de ultrassom é da ordem de milisegundos, no LIDAR é muito reduzida, ficando na ordem de algumas dezenas de picosegundos.

Além disso, por causa da altíssima diretividade do feixe laser, é possível haver o disparo simultâneo de vários feixes em ângulos diferentes, aumentando a velocidade de amostragem do sensor.

O LIDAR mostrado nesta seção é o Velodyne HDL-64E (VELODYNE Inc., s.d.), que foi utilizado no veículo Boss, da Universidade Carnegie Mellon, vencedor do desafio DARPA 2007 de navegação urbana. Este dispositivo tem uma resolução angular de $0,5^\circ$ na vertical e $0,09^\circ$ na horizontal, gerando uma nuvem de informação de mais de 1 milhão de pontos por segundo. O seu mecanismo gira a velocidades configuráveis de 5 a 20 voltas por segundo. São 64 lasers, distribuído em 2 blocos de 32 emissores e faz disparos simultâneos de um laser de cada bloco, em cada instante.

A informação fornecida pelo LIDAR é um conjunto de pontos (chamado de “nuvem de pontos”) contendo distância, azimute e intensidade de retorno do sinal. O laser do HDL-64 opera na faixa do infravermelho, em uma região do espectro que não é prejudicial ao olho humano, com comprimento de onda na ordem de 900 nm, e uma medida com precisão de alguns milímetros.

O ponto forte do LIDAR é o baixo espalhamento da onda de luz laser, que oferece uma alta precisão na determinação de posição do ponto, tanto no ângulo horizontal quanto no ângulo vertical. Isso permite identificar os pontos dos obstáculos e uma condução mais exata.

O principal ponto fraco do LIDAR, no momento, é o seu alto custo, quando comparado a outros sensores disponíveis no mercado, mas que deve cair à medida em que o seu uso for sendo mais disseminado.

A Figura 15 (VELODYNE Inc., s.d.) mostra o LIDAR HDL-64E, da empresa Velodyne, e a Tabela 2 apresenta as suas principais características.

Figura 15 - Lidar Velodyne HDL-64E

Fonte: (VELODYNE Inc., s.d.)

Tabela 2 - Características do Velodyne HDL-64E

Especificação do Lidar Velodyne HDL-64E	
Laser:	<ul style="list-style-type: none"> • Classe 1 – não prejudicial à vista humana • Comprimento de onda de 905 nm (faixa do infravermelho) • Medida de distância pelo tempo de percurso (<i>Time of flight</i>) • Alcance de medição: de 1m até 70 m
Sensor:	<ul style="list-style-type: none"> • 64 pares de emissores / detectores de laser • Campo de visada vertical (<i>vertical field of view</i>) de +2° até -29.5° • Campo de visada horizontal de 360° • Taxa de quadros (<i>frame rate</i>) entre 5 e 20 Hz • Temperatura de operação entre -10°C a +50°C • Precisão: <2 cm • Resolução angular vertical ~0.5° • Resolução angular horizontal ~0.09°
Mecânica:	<ul style="list-style-type: none"> • Consumo: 4 A @ 12V • Peso: 13 kg • Nível de proteção ambiental: IP67
Saída:	<ul style="list-style-type: none"> • Aproximadamente 1.333.000 pontos/s • Interface Ethernet 100 Mbps

Copyright ©2011 Velodyne Lidar, Inc..

2.2.2.1 Padronização de interface

Uma dificuldade no uso do Lidar, que aparece imediatamente, é a quantidade de dados por unidade de tempo que é fornecida pelo sistema. Para cada ponto que o Lidar identifica, pelo menos três coordenadas precisam ser enviadas. O formato destes dados depende do fabricante e, por uma questão de eficiência, é melhor se for transmitida em codificação binária, que é mais compacta, do que em codificação texto (p.ex. ASCII).

Entretanto outro problema está sendo discutido pelos fabricantes e usuários (entre eles, o Corpo de Engenheiros do Exército dos Estados Unidos - USACE): a padronização da interface de dados dos lidares. Cada fabricante pode enviar os dados no formato que considerar mais eficiente, dificultando a integração de dispositivos diferentes.

Este assunto está sendo discutido pela ASPRS (American Society for Photogrammetry and Remote Sensing – www.asprs.org) e poderá definir um padrão de formato de arquivo .LAS (de LASer). No momento a última proposta disponível é a versão 1.4 de Novembro/2011, disponível em

http://info.asprs.org/society/committees/standards/LAS_1_4_r13.pdf .

2.2.3 Radar (RAdio Detection and Ranging)

O Radar é comercialmente utilizado em veículos para funções tais como detecção de obstáculos à frente (detecção de colisão) e laterais (gerenciamento de mudança de faixa). Sua operação é baseada no envio de pulsos de ondas eletromagnéticas (rádio) e na avaliação do tempo de resposta do eco refletido.

Veículos autônomos como o Stanley, da Universidade de Stanford, vencedor do premio DARPA 2005 (THRUN, et al., 2006) estavam equipados com um ou mais sistemas de radar para a identificação de obstáculos a distâncias de algumas centenas de metros (200 metros, no caso do Stanley). Com ângulos de visão (FOV – *Field Of View*) tão pequenos quanto 1º, fornecem uma resolução suficiente para discriminar convenientemente os obstáculos a longa distância.

A tabela 3 mostra as principais características do sistema ARS 300 (CONTINENTAL Corporation, 2012), da A.D.C. GmbH, empresa do grupo Continental Corporation, usado no veículo autônomo Stanley. Este sistema de radar, além de detectar a presença dos obstáculos, é capaz de rastrear e acompanhar o movimento de um número limitado deles, fornecendo azimuth e distância a cada instante.

Uma funcionalidade no sistema permite que o radar acompanhe grupos de objetos e gere uma mensagem de aviso caso algum deles esteja em rota de colisão e esta colisão possa ocorrer dentro de um limiar de tempo configurado.

Tabela 3 - Características do Radar ARS-30X

ARS 30X – Radar-Sensor 77 GHz

Característica	Valor para alvos não-refletores
Distância de detecção	0,25m a 200 m
Resolução de medição de distância	0,25m ou 1,5% para distâncias acima de 1 m
Ângulo de visada horizontal	-8,5° a + 8,5°
Limites de velocidade	-88 km/h (aproximando) a +265 km/h (afastando)
Resolução de medição de velocidade	2,76 km/h
Ciclo de medição	66 ms
Número de antenas	17

Fonte: (CONTINENTAL Corporation, 2012)

A principal vantagem do radar é a grande distância alcançada, aliada a um custo relativamente baixo por ser uma tecnologia já conhecida. Além disso, por operar com um tipo de onda diferente e em uma frequência mais alta do que o ultrassom, é mais imune a interferências geradas por partículas suspensas no ar tais como gotículas de água em suspensão em condições de neblina .

Figura 16 - Diagrama de detecção do radar

Fonte: (CONTINENTAL Corporation, 2012)

3 MATERIAIS E MÉTODOS

A fase experimental desta pesquisa consistiu em escolher um método de desvio de obstáculo, realizá-lo em uma plataforma robótica e verificar sua eficiência e limitações, identificando se foram oriundas do algoritmo utilizado ou causadas pelas características dos sensores e atuadores.

O algoritmo selecionado foi o do Inseto-2 e a plataforma robótica em que foi aplicada foi o Robodeck, que já vem dotada de sensores e atuadores controláveis por placas de controle programáveis.

O procedimento planejado foi:

- Avaliar os sensores quanto à acurácia das medidas
- Avaliar os atuadores quanto ao seu funcionamento e eficácia
- Realizar alguns percursos, com diferentes configurações de obstáculos, observando a capacidade de alcançar o objetivo com contorno e recuperação da trajetória original após a detecção do obstáculo.

O ambiente onde o veículo trafegou foi um espaço plano 2-D (bidimensional) e todos os obstáculos foram considerados dentro deste espaço.

3.1 A PLATAFORMA PARA VALIDAÇÃO

A plataforma selecionada foi o Robodeck, produzido pela empresa XBOT, que oferece algumas facilidades já embarcadas no veículo para a realização proposta nesta dissertação. Nesta seção são apresentadas as principais características da plataforma que influenciam esta pesquisa. Estas informações foram retiradas de (XBOT, 2011).

O Robodeck é uma plataforma robótica móvel, de acordo com a própria definição de seu fabricante. Na definição geral, é um veículo terrestre de 4 rodas com movimentação controlada por atuadores que são acionados por placas de processamento embarcadas.

Possui diversos sensores já integrados a duas placas de processamento, motores e atuadores que permitem sua movimentação e direção, comunicação sem fio e um ambiente de *software* baseado em sistema aberto (Linux) com biblioteca de funções e SDK (*Software Development Kit*).

No início da fase experimental o Robodeck teve que passar por uma manutenção. Enquanto se esperava pelo seu retorno, alguns conceitos foram aplicados em uma plataforma Lego Mindstorms EV3. Entretanto, esta plataforma se mostrou muito limitada e foi utilizada apenas para os testes iniciais. O relatório de testes e resultados obtidos com o Lego Mindstorms estão no Apêndice A - Experimentos com Lego Mindstorm EV3.

3.1.1 Alimentação e autonomia

O Robodeck é alimentado com baterias de Íons de Lítio (Li-Io) recarregáveis que energizam motores de tração, motores de direção, sensores e duas placas de processamento embarcadas.

Existem 6 posições de encaixe (*slots*) para as baterias mas o fabricante recomenda operar com 2 ou 4 baterias. O número de baterias afeta a autonomia e a velocidade máxima que o veículo pode atingir.

De acordo com a documentação do Robodeck, com 4 baterias o Robodeck poderia operar por 4 horas. Durante os experimentos, o Robodeck operou sempre com 2 baterias. Nesta condição, o tempo médio de operação foi de menos de 1 hora.

Uma característica que foi observada durante os experimentos foi que a carga da bateria afeta a exatidão da movimentação. Não foram feitos testes para avaliar a extensão desta influência. A medida adotada foi iniciar cada experimento com um conjunto de baterias totalmente carregadas.

3.1.2 Sensores disponíveis

O Robodeck possui os seguintes recursos de sensoriamento já instalados e mostrados na Figura 17:

- Web Cam (1);
- Sensores infravermelhos (7), com alcance entre 4 cm e 30 cm e um ângulo de abertura de 5°, posicionados da seguinte maneira:
 - Dois sensores apontados para a frente, localizados na frente direita e na frente esquerda;
 - Dois sensores direcionados para baixo, localizados na frente e na traseira;
- 4 Sensores ultrassônicos (2 e 5) localizados na frente, na traseira e nas duas laterais, com alcance entre 3 cm a 6 m, com um ângulo cônico de abertura de feixe de 45° (22,5° para cada lado do eixo);

Figura 17 - Dispositivos externos do Robodeck

Fonte: (XBOT, 2011)

Além destes sensores mostrados na figura, o Robodeck possui internamente os seguintes recursos:

- Bússola Eletrônica que fornece a orientação do Robodeck através do ângulo formado entre o eixo de movimentação orientado no sentido

da parte dianteira do veículo e o norte magnético.

- Receptor GPS com antena embutida e interface NMEA¹.
- Acelerômetro de 3 eixos sendo o eixo X-positivo apontado para a traseira do veículo, o eixo Y-positivo apontado para a direita e o eixo Z-positivo apontado para baixo.
- Sensor de Temperatura e Umidade: da marca Sensirion® SHT11, capaz de coletar dados sobre a temperatura e umidade ambiente. As leituras podem ser feitas na faixa de -40°C a +123°C) e a umidade entre 0 e 100%.

3.1.3 Movimentação

O Robodeck pode ser movimentado através do controle exercido pelos seus motores de tração e direção. Esta movimentação pode ser resumida nas características abaixo:

- Tração independente nas duas rodas dianteiras com um motor para cada roda. Cada motor de tração possui informação de conta giros independentes e que pode ser lida pelo controlador para oferecer uma função de odometria.
- Controle de direção independente nas 4 rodas, permitindo movimentos múltiplos tais como esterçamento, giro sobre o próprio eixo e movimento em diagonal. A Figura 18 mostra algumas posições possíveis.
- O giro máximo do eixo de direção das rodas é de 47°, o que impede um grau de liberdade total. Portanto o robô é um veículo não holonômico.
- Velocidade máxima varia de acordo com a quantidade de baterias que forem instaladas. Na configuração mínima (2 baterias), pode atingir 5 km/h e na configuração máxima (4 baterias) pode atingir 3

¹ O formato NMEA prevê chaves e campos pré-definidos para identificar as informações fornecidas por um receptor GPS. É um padrão de interface para o mercado de localização por satélite

km/h.

- Autonomia estimada pelo fabricante: 4 h com as 4 baterias instaladas.

Figura 18 - Movimentação das rodas do Robodeck

3.1.4 Processamento embarcado

O Robodeck opera com 2 placas de processamento embarcados, distribuindo as funções. Os nomes atribuídos a estas placas são de autoria do fabricante.

- MCR (Módulo de Controle Robótico), que atua diretamente sobre o *hardware* do veículo. Este módulo é composto por um processador de

arquitetura ARM, *drivers* que acionam os motores e interfaces para o recebimento das informações dos sensores.

Ao MCR estão conectados os seguintes recursos do Robodeck:

- Atuadores de direção das rodas
- Motores de tração (rodas dianteiras)
- Bússola
- Sensores ultrassônicos
- Sensores infravermelhos
- Acelerômetro
- MAP (Módulo de Alta *Performance*) que se destina ao processamento de mais alta complexidade e a interfaces de alto nível tais como dispositivos USB.

O MAP é um PC Mini-ITX embarcado no Robodeck. Este módulo possui um sistema operacional Linux e permite que o Robodeck possa interagir com outros dispositivos através de comunicação via Zigbee ou Wifi.

Ao MAP estão conectados os seguintes recursos

- Câmera
- Comunicação WiFi
- Comunicação Zigbee
- Porta USB
- Porta de vídeo VGA

O MCR e o MAP se comunicam entre si através de uma interface serial operando em um formato de 8 bits a uma velocidade de 19200 bps (bits por segundo). Programas embarcados no MAP podem controlar o Robodeck emitindo comandos para o MCR e recebendo informações do MCR através desta interface.

O formato das mensagens e o protocolo utilizado está no Apêndice C.1.

3.2 PLANEJAMENTO DOS EXPERIMENTOS

O objetivo dos experimentos foi verificar a viabilidade e eficiência de um

método de desvio de obstáculos baseado em sensores, sem um conhecimento prévio dos obstáculos contidos no trajeto.

Para isso, os seguintes passos foram planejados:

1. Selecionar um algoritmo e um conjunto de sensores
2. Escolher um ambiente de desenvolvimento e uma linguagem de programação para realizar o experimento
3. Testar a operação do Robodeck para verificar a eficiência dos sensores e dos atuadores
4. Escrever um programa que realizasse um percurso utilizando o algoritmo e os sensores escolhidos
5. Testar o funcionamento do programa em diversas situações.

3.2.1 Sensores ultrassônicos

O Robodeck vem equipado com quatro sensores ultrassônicos que medem distâncias entre 0,03m e 6m, apontados para a frente, para trás, para a direita e para a esquerda, em relação à frente do robô. As características destes sensores se encontram no manual do Robodeck.

Os sensores do Robodeck são componentes que já fornecem a informação de distância calculada. Uma desvantagem de usar este tipo de sensor que já fornecem a distância pronta é a impossibilidade de se aplicar algoritmos pulsantes diferenciados para redução de erros, tal como o descrito em (BORENSTEIN & KOREN, 1995) uma vez que quem gerencia a emissão e o tratamento do sinal ultrassônico é o próprio sensor.

A baixa diretividade, o número de sensores e a sua distribuição no Robodeck foi um fator limitante para os experimentos. Por exemplo, em (BORENSTEIN & KOREN, 1990) foram usados 24 sensores ultrassônicos Polaroid 6500 Sonar Ranging, operando a 50 kHz e com ângulo de abertura de 15°, distribuídos uniformemente à volta do robô, para ter uma abrangência plena de 360°, sem pontos cegos².

² Um ponto cego é uma região não coberta por um sensor, de onde não se tem informação

Durante a pesquisa, foi notado que o ângulo de abertura do sensor afeta diretamente a exatidão dos movimentos do veículo. Um veículo com apenas quatro sensores pode ter o conhecimento apenas de quatro direções e limitado ao ângulo de abrangência do sensor. As posições não cobertas pelos sensores não devem ser utilizadas pelo veículo por falta de conhecimento do conteúdo de tais posições.

A opção de melhor eficiência seria criar uma rede de sensores dispostos circularmente, de maneira similar ao que foi feito em (BORENSTEIN & KOREN, 1990). Cada sensor cobrirá um ângulo de abertura e a área dentro daquele ângulo será avaliada de acordo com a leitura do sensor.

3.2.2 Algoritmo de desvio

Para o planejamento de trajetória do Robodeck, foi selecionada a variante Inseto 2 por ser apropriada para uso com sensores de baixo alcance em distância e exigir uma capacidade de processamento de informações dentro das limitações do processador do MAP. Por exemplo, o algoritmo do inseto tangencial necessitava de sensores de maior alcance e diretividade do que os disponíveis no Robodeck.

Além disso, por exigir menos movimentação do que o Inseto 1 para determinar o ponto de relargada, deveria acumular menos erros causados pelas incertezas de odometria e giro angular.

De posse das coordenadas dos pontos origem e destino, o programa traça um segmento de reta entre eles e inicia o movimento sobre esta reta e com sentido apontado para o destino.

A Figura 19 mostra o percurso pretendido pelo Robodeck e o comportamento esperado em face de um obstáculo. Nela, os pontos numerados indicam situações de tomadas de decisão, descritos a seguir:

- (1) Obstáculo detectado na trajetória planejada: Inicia manobra de contorno
- (2) Limite de obstáculo atingido. Mudar curso para acompanhar o contorno
- (3) Limite de obstáculo atingido. Mudar curso para acompanhar o contorno
- (4) Retorno à trajetória original com retomada do curso em direção ao alvo

Figura 19 - Planejamento

Durante o percurso, o programa mantém uma distância de segurança de qualquer obstáculo. Esta distância mínima é necessária para assegurar que o Robodeck possa executar a manobra de desvio sem colidir com o obstáculo detectado.

Durante os experimentos, esta distância mínima foi definida arbitrariamente com o equivalente ao comprimento do Robodeck.

3.2.3 Ambiente de desenvolvimento e programação

A linguagem escolhida foi o Java por já ser familiar e por ser portátil entre os diferentes ambientes de execução (Windows e Linux).

Para o desenvolvimento dos programas, foi escolhido o IDE (*Integrated Development Environment*) NetBeans, obtível gratuitamente em www.netbeans.org. O ambiente já traz um compilador Java embutido e pode também ser executado em ambos os sistemas.

Durante a execução dos experimentos, o IDE foi mantido estável, sem buscar novas atualizações que poderiam instabilizar ou mudar algum resultado anterior. O ambiente utilizado foi o seguinte:

- NetBeans IDE 8.0.2 (Build 201411181905)
- Java: 1.7.0_25; Java HotSpot(TM) 64-Bit Server VM 23.25-b01
- Runtime: Java(TM) SE Runtime Environment 1.7.0_25-b17

Por uma questão de praticidade, o desenvolvimento foi feito inteiramente em um notebook dotado de sistema operacional Windows 7 Enterprise. Este notebook substituiu o MAP durante a execução dos experimentos, com influência desprezível tanto nos resultados quanto na arquitetura original do Robodeck.

Os experimentos foram desenvolvidos e executados no notebook que se comunicava com o MCR do Robodeck através de uma interface serial. Esta interface serial era a mesma que conectava originalmente o MAP ao MCR e consistia em um adaptador USB-Serial com saída em TTL-3.3V. Não foi possível identificar com certeza o fabricante mas, em pesquisa na internet, o produto que se assemelhava o instalado no Robodeck era o TTL-232R-3V3, da FTDI (Future Technologies Devices), disponível em www.ftdichip.com.

Assim, as instruções enviadas pelo programa no notebook eram executadas pelo MCR e as informações dos sensores e atuadores eram lidas de volta pelo programa.

3.2.4 Posicionamento horizontal e orientação angular

Para o funcionamento do algoritmo de desvio, além da medição de distâncias, o conhecimento de posição do veículo é necessário. Para isso, duas informações precisam ser obtidas:

- Posicionamento horizontal, que identifica a localização onde o veículo está em relação ao seu destino. Para os fins deste experimento, a posição horizontal do Robodeck foi adotado como sendo dentro de um espaço de coordenadas cartesianas bidimensional.
- Orientação angular, que identifica para onde o veículo está apontado e sendo movimentado. Para os fins deste experimento, a orientação foi determinada como o desvio formado pelo eixo do Robodeck (frente

do veículo) e o eixo X positivo de coordenadas cartesianas.

Para o posicionamento horizontal, os sensores de posição das rodas (*encoders*) foram utilizados. Estes sensores informam, através de uma contagem, o movimento angular das rodas. Com esta contagem é possível saber quanto cada roda girou.

Dois pressupostos foram assumidos para o cálculo de posicionamento:

- a) Conhecimento da posição horizontal e da orientação angular ao início do experimento.
- b) Movimentações feitas em linha reta, sem mudança da orientação angular durante cada movimentação

Ao final de cada movimentação, as novas coordenadas são calculadas com base nas coordenadas anteriores acrescidas do deslocamento obtido durante a movimentação.

Denominando-se:

- (X_0, Y_0) como as coordenadas iniciais, antes do movimento
- (X_f, Y_f) como sendo as coordenadas finais do movimento
- D como sendo a distância em linha reta percorrida no movimento
- α como sendo a orientação angular durante o movimento

calculam-se as novas coordenadas de acordo com as expressões:

- $X_f = X_0 + D \cdot \cos(\alpha)$ (i)
- $Y_f = Y_0 + D \cdot \sin(\alpha)$ (ii)

Devido ao fato de que o Robodeck dispunha apenas de quatro sensores ultrassônicos, permitindo o conhecimento de obstáculos apenas nas quatro direções principais (frente, trás, direita e esquerda), as manobras do Robodeck foram feitas com giros de 90° . Com, isso, as expressões (i) e (ii) ficam muito simplificadas pois os movimentos passaram a ser sempre paralelos aos eixos coordenados, com os ângulos α assumindo sempre os valores de 0° , 90° , 180° e 270° .

A determinação da orientação angular apresentou mais problemas. Inicialmente foi planejado o uso da bússola para a determinação desta orientação. Dessa maneira, o eixo coordenado X seria apontado na direção leste-oeste, com o eixo X-positivo apontado para o leste.

Entretanto, durante o experimento, a bússola se mostrou inviável conforme mostraram os resultados dos testes apresentados mais adiante. Por este motivo, a determinação de orientação angular foi feita usando contagem de número de voltas das rodas durante giro do Robodeck.

3.3 EXECUÇÃO DOS EXPERIMENTOS

O MCR oferece rotinas prontas para fazer o Robodeck executar diversas ações. Estas rotinas são executadas de acordo com o comando enviado pela porta serial e estão listadas no manual de software do Robodeck.

As rotinas estão divididas em comandos de baixo nível e comandos de alto nível. Para os experimentos, não foi levado em conta se o comando seria de baixo nível ou de alto nível. Apenas o efeito que seria obtido foi levado em conta.

Para o controle de direção, foi usado apenas o sensor de contagem de pulsos para informação de odometria e o controle dos servo-motores de direção para esterçar o veículo. O percurso foi executado baseado nas coordenadas cartesianas do ponto de origem e do ponto de destino.

Os experimentos foram divididos em 2 partes:

- Calibração e conhecimento dos comandos
 - Movimentação em reta
 - Giro angular
 - Medição de distâncias
- Execução do programa de desvio de obstáculos
 - Desvio simples
 - Desvios complexos
 - Objetivo inalcançável

3.3.1 Movimentação em linha reta

O Robodeck oferece um comando que o movimenta em linha reta, com uma velocidade (intensidade) definida como parâmetro do comando. Por ser uma operação em malha aberta, sem realimentação de qualquer tipo, era esperado que a suposta trajetória em reta apresentasse um desvio não desprezível em relação à trajetória prevista.

Os objetivos destes experimentos foram:

- 1) Calibrar as informações de rotação fornecidas pelos motores, relacionando-as com distâncias em metros.
- 2) Avaliar a acurácia das informações fornecidas pelo sistema Robodeck, relacionados com as distâncias percorridas.
- 3) Avaliar o desvio que o robô apresenta em uma reta

Para a leitura dos sensores de giro das rodas, os dois valores (roda esquerda e roda direita) foram lidos e o valor médio (média aritmética simples) foi usado como informação de odometria.

Inicialmente foi programado um percurso em linha reta com um número arbitrário de pulsos de contagem. Este teste indicou uma relação entre a contagem e a distância percorrida em metros. O valor arbitrário escolhido foi 2000 pulsos e a distância encontrada foi de 1,83m.

Com este valor, foi determinada uma relação aproximada de 1093 pulsos por metro percorrido. Este valor foi reintroduzido no programa e foram feitos 5 testes medindo a distância real percorrida e o desvio da reta, de acordo com a Figura 20.

Figura 20 - Medição de Odometria

A Tabela 4 contém os resultados obtidos para esta movimentação de 1093 pulsos, com uma intensidade de movimentação (velocidade) de 4000 (valor interno do Robodeck), escolhido arbitrariamente.

# Tentativa	Distância real (m)	Desvio (m)	Contagem Direita	Contagem Esquerda
1	1,03	0,20	1033	1155
2	1,04	0,14	1046	1175
3	1,04	0,10	1063	1139
4	1,07	0,12	1063	1144
5	1,05	0,10	1072	1141
Média	1,04	0,17		

Tabela 4 - Medição de acurácia da odometria

O valor de rotação é obtido pela leitura dos contadores de giros nas rodas (*encoders*) que é fornecido pelo MCR. A leitura de cada roda (direita ou esquerda) é independente.

Mesmo em linha reta, os valores de contagem de cada roda são diferentes. A tabela anterior mostra os valores de contagem das duas rodas independentes. As hipóteses levantadas para estas diferenças foram:

- diferenças de tamanho das rodas,
- deslizamentos durante os movimentos
- ângulo das rodas
- diferenças dos motores de tração

3.3.2 Giro sobre o eixo

O Robodeck oferece um comando que o faz girar sobre o seu próprio eixo. Para isso, o MCR vira o ângulo das rodas do eixo dianteiro em um sentido e o das rodas do eixo traseiro no sentido oposto. Dessa maneira, quando os motores de tração são acionados, o robô gira sobre o seu próprio eixo vertical.

Este giro foi usado para fazer as manobras de desvio: quando o Robodeck manobrava, ele parava o movimento retilíneo, girava 90° para um dos lados e depois prosseguia. A determinação da exatidão destas manobras foi o objetivo deste experimento. Para realimentar a tarefa de controle, era necessária uma informação

de sensoriamento angular. Para isso, foram testadas duas abordagens

3.3.2.1 Com uso da Bússola

O objetivo deste teste foi avaliar a acurácia e a precisão das informações fornecidas pela bússola do Robodeck. A exatidão desta informação viabilizaria o seu uso como realimentação de malha relacionado com a manobra de giro. Em todas as tabelas desta seção, onde não houver menção da unidade, o valor estará expresso em décimos de grau, que é a unidade fornecida pela bússola do Robodeck.

Inicialmente o Robodeck foi colocado com a sua frente apontada para o norte magnético. Este posicionamento inicial foi feito com o auxílio de uma bússola magnética de agulha.

Um ajuste fino foi feito, usando a própria leitura da bússola do Robodeck, posicionando-o manualmente para dar uma leitura o mais próxima possível de 0° , que é o valor de azimuth fornecido para a direção Norte. Após isso, marcas foram feitas para estabelecer as orientações ortogonais (0° , 90° , 180° e 270°)

A rotina de leitura de bússola era feita com os valores retornados sendo inseridos em uma média móvel de 4 amostras. A amostra mais recente era incluída na média e a mais antiga era retirada.

Para cada orientação, foram feitas 3 leituras. Os resultados das leituras da bússola estão expressos na Tabela 5. A interpretação para os valores listados são os seguintes:

- Ângulo posicionado: Posição que o Robodeck estava sendo presumivelmente colocado. Estas posições (0° , 90° , 180° e 270°) foram marcadas no chão, previamente
- Leitura bruta: É o valor fornecido pela bússola do Robodeck, após a colocação na posição pressuposta. Embora o ângulo inicial de 0° tivesse sido marcado com o auxílio de uma bússola externa, a leitura da bússola do Robodeck fornecia um valor diferente. Por isso, o valor lido na posição presumida de 0° foi usado como correção de deslocamento inicial (*offset*) para os demais valores lidos.

- Leitura compensada: Valor do ângulo após a retirada do deslocamento inicial
- Desvios: diferença entre o ângulo fornecido pela bússola e o ângulo que era esperado.

	Ângulo Posicionado (em °)	Leitura bruta	Leitura compensada	Desvio em ° (s/compensação)	Desvio em ° (c/compensação)
Teste #1	0	270	0	27	0
	90	1113	843	21,3	-5,7
	180	1812	1542	1,2	-25,8
	270	2614	2344	-8,6	-35,6
	360	195	-75	19,5	-7,5
Teste #2	0	201	0	20,1	0
	90	1153	952	25,3	5,2
	180	1855	1654	5,5	-14,6
	270	2612	2411	-8,8	-28,9
	360	203	2	20,3	0,2
Teste #3	0	215	0	21,5	0
	90	1186	971	28,6	7,1
	180	1856	1641	5,6	-15,9
	270	2622	2407	-7,8	-29,3
	360	121	-94	12,1	-9,4

Tabela 5 – Leituras da bússola do Robodeck

Como um exemplo, tome-se a medida do teste 1, no ângulo de 90° (posições sombreadas na Tabela 5).

- Valor lido (em décimos de grau): 1113
- Subtraindo-se o offset inicial (270 décimos de grau, presente na posição de 0°), obtém-se um ângulo corrigido de 843 décimos de grau. Este é o ângulo que o Robodeck informa para a posição de 90°
- Como o ângulo esperado era de 90°, o erro calculado (com a compensação do *offset* de 270 décimos de grau) foi de 5,7° e o erro medido (sem a compensação) foi de 21,3°.

Observou-se um grande erro nas medidas angulares absolutas fornecidas pela bússola do Robodeck.

O teste seguinte foi uma verificação dos valores que a bússola informa quando o veículo está apontado para os pontos cardeais. Em outras palavras, quais valores correspondem às posições ortogonais de 0° (Norte), 90° (Leste), 180° (Sul) e 270° (Oeste).

A colocação foi feita manualmente, obedecendo a marcas no chão feitas previamente. Estas marcas foram feitas usando o Norte do Robodeck (ângulo zero da bússola) como referência.

	Esperado	Teste 1	Teste 2	Teste 3	Media	Desvio
Norte	0	0	-40	-50	-30	26,5
Leste	900	1023	1011	1001	1012	11
Sul	1800	1741	1764	1742	1749	13
Oeste	2700	2467	2458	2462	2462	4,5

Tabela 6 – Ângulos dos pontos cardeais

A Tabela 6 apresenta estes valores e mostra que:

- Os valores que o Robodeck informa para os pontos cardeais não correspondem aos ângulos esperados para os mesmos. Por exemplo, para a posição correspondente ao Leste, o valor esperado seria de 900 (décimos de grau) mas a leitura fornecida foi de 1023, no teste 1.
- As indicações no eixo Norte-Sul se mostraram mais precisas, com erros menores do que as do eixo Leste-Oeste.
- Para um determinado ângulo, as leituras fornecidas são muito próximas. Por exemplo, a indicação de Leste, nos 3 testes, teve uma variação máxima de 22 décimos de grau (entre 1023 e 1001). Esta variação apresentada pode ser atribuída ao erro de posicionamento, uma vez que foi feito manualmente.

Além deste erro na medição, a informação da bússola do Robodeck contém ruídos que provocam interpretações errôneas e que necessitam de tratamento antes de serem usadas.

Dois problemas foram sentidos durante os experimentos:

- A colocação de um notebook sobre a carenagem do Robodeck causa erros nas leituras. Dependendo da posição em que o notebook era colocado, as leituras variavam.

- Em várias ocasiões, houve um atraso na resposta da bússola. Embora o robô já estivesse em um determinado ângulo, a bússola reportava um ângulo anterior e, depois de algumas leituras erradas, fornecia o ângulo correto.

Para tentar reduzir o primeiro efeito, durante todos os testes com a bússola, o notebook foi colocado o mais longe possível do Robodeck. Devido à distância máxima do cabo USB que ligava o MCR ao notebook, esta distância foi de aproximadamente 0,4 m.

Mesmo assim, quando havia o atraso na leitura e este atraso coincidia com a aproximação do ângulo que o programa estava esperando, o giro feito pelo Robodeck terminava com um erro muito grande.

Por causa destas dificuldades, o sensor bússola foi abandonado como referência de orientação.

3.3.2.2 Com uso da Odometria

O controle de giro com realimentação pela odometria se baseou no pressuposto de que um número determinado de pulsos das rodas corresponderia a um determinado ângulo de giro, desde que o ângulo de esterção das rodas fosse sempre o mesmo.

Para obter este valor, foi feito um procedimento de tentativa e erro. O procedimento adotado foi o seguinte:

1. Fixar uma posição de referência inicial no Robodeck
2. Executar um giro com uma contagem inicial arbitrária e verificar se completou um giro de 360°
3. Realimentar o valor de contagem e repetir os passos (2) e (3) até conseguir um giro de 360° , anotando a contagem do odômetro (*encoder*) das rodas.
4. Dividir o valor obtido em (3) por 4 para obter a contagem correspondente a um giro de 90° , que seria usado pelo programa de desvio de obstáculos

Todas as contagens de pulsos foram feitas usando a média das contagens

individuais de cada uma das rodas de tração. Em outras palavras, a contagem usada foi a soma das contagens de pulsos das duas rodas de tração, dividida por 2. Isso foi feito para poder desconsiderar a diferença de contagens entre a roda interna (menor número de pulsos) e a roda externa (maior número de pulsos), durante uma manobra de giro.

Foram feitas várias tentativas até se chegar a um número que resultava em uma volta de 360 graus com um erro menor do que 5°. Com estes testes, notou-se que a movimentação de giro não era simétrica, resultando em valores diferentes para o sentido horário e para o sentido anti-horário.

Após alguns testes, os melhores valores obtidos para um giro de 360° foram:

- Giro no sentido anti-horário: 2130 pulsos dos contadores das rodas
- Giro no sentido horário: 2140 pulsos dos contadores das rodas.

Para uma avaliação do erro inserido pela manobra de giro foram feitos os testes descritos na tabela abaixo, já utilizando os valores obtidos nos testes empíricos.

Descrição do teste angular	Erro acumulado ao final (em °) ³		
	Teste 1	Teste 2	Teste 3
8 movimentos de 90 graus no sentido horário	10	10	10
8 movimentos de 90 graus no sentido anti-horário	5	5	10
8 movimentos de 90 graus em sentidos alternados (horário e anti-horário)	5	5	5

Tabela 7 - Avaliação de erro angular

Um fator que apareceu foi a influência da carga da bateria nos movimentos angulares. Quando a bateria está a plena carga, o erro angular se mostrava menor do que com a bateria mais descarregada.

³ Todos os ângulos foram medidos com o uso de um transferidor manual simples. Por isso a precisão dos valores foi aproximada para 5°

3.3.3 Sensor ultrassônico

O sensor utilizado para a medição de distâncias, necessário ao funcionamento do algoritmo do inseto, foi o ultrassônico por ser o que estava disponível no Robodeck e que tinha um alcance suficientemente grande.

Os objetivos deste teste foram:

- verificar a relação entre o valor fornecido pelo sensor ultrassônico e a distância real do obstáculo.
- Avaliar o ângulo de cobertura do sensor ultrassônico

3.3.3.1 Medidas de distância

O experimento foi feito posicionando-se o Robodeck a uma distância conhecida de uma parede sólida. A distância utilizada como a real foi medida a partir da ponta da carenagem do Robodeck mas o sensor se localiza um pouco para trás, na parte superior da carenagem, devido à inclinação dela. Isto acarretaria em um erro de 0,09m entre a medida lida e a distância real, conforme mostra a Figura 21.

Figura 21 - Diferença de distância Sensor Dianteiro e frente do Robodeck

Em um primeiro instante chegou-se a aventar a hipótese de se fazer a compensação desta diferença nas leituras obtidas. Mas os resultados mostraram que a diferença (falta de acurácia) das leituras era muito maior do que esta compensação, principalmente para distâncias maiores.

Os resultados dos testes são apresentados na Tabela 8.

	Distância da ponta da carenagem até o Obstáculo (valores em metros)			
Teste	Real = 0,63	Real = 1,04	Real = 1,45	Real = 2,67
1	0,74	1,16	1,25	1,65
2	0,74	1,15	1,24	1,45
3	0,75	1,16	1,27	1,18
4	0,74	1,16	1,26	1,74
5	0,74	1,16	1,25	1,56
Erro médio	0,11	0,12	-0,19	-1,15

Tabela 8 - Medidas de distância com sensor ultrassônico

Os resultados dos testes indicam uma melhor acurácia no valor informado para distâncias menores. No caso dos 2 primeiros testes (0,63m e 1,04m), se for compensada a distância entre o sensor e a ponta da carenagem, este erro cai para 0,02m. Entretanto, quando a distância do obstáculo aumenta, o erro aumenta de maneira desproporcional.

Durante os testes, apareceram valores espúrios muito fora de erros de medida aceitáveis. Estes valores foram expurgados dos resultados deste experimento mas indicaram que o sensor pode fornecer informações erradas com características de ruído.

Por este motivo, na realização do programa de teste, a leitura do sensor ultrassônico foi feito com a média de 4 valores fornecidos pelo sensor. Esta média procurou reduzir o efeito de uma leitura espúria.

3.3.3.2 Medidas de Ângulo de Abrangência

A determinação do ângulo de abrangência visou verificar as posições em

que o sensor conseguia detectar obstáculos. Este teste permitiu identificar pontos “cegos” para o Robodeck – locais onde um obstáculo não seria identificado, mesmo estando presente lá.

Para este teste, o Robodeck foi posicionado de maneira que outros obstáculos que pudessem interferir com a medida estivessem a uma distância maior do que 2,5 m. Esta distância foi um valor empírico, resultante da medição da distância do Robodeck até o obstáculo mais próximo, após o obstáculo usado para o teste.

Um obstáculo de teste, situado a 1,00 m da frente do Robodeck foi movido lentamente, a partir de um ponto situado à direita do Robodeck, em direção ao seu eixo (frente), perpendicularmente ao eixo, até ser detectado pelo sensor ultrassônico. Esta detecção foi caracterizada pela mudança da medida de distância apresentado pelo programa de testes.

O mesmo procedimento foi repetido com o obstáculo vindo de um ponto à esquerda do Robodeck. A Figura 22 mostra uma vista superior do que foi executado neste experimento.

Figura 22 - Medição de ângulo de abrangência

O ângulo de abertura pode ser calculado pela expressão

$$w = 2 * \arctg\left(\frac{a}{2d}\right)$$

O valor da distância “a” medida no experimento foi de 1,30m. Com isso, o

ângulo de abertura do sensor frontal medido nos testes com o objeto a 1m foi de 66° (33° para cada lado).

Com isso, temos uma visão dos pontos cegos do Robodeck na configuração atual, para obstáculos situados a 1 m do robô. A Figura 23 mostra as regiões cobertas pelos sensores (áreas sombreadas) e as regiões “cegas” (áreas claras).

É sabido que o sensor de ultrassom possui lóbulos de irradiação que apresentam ângulos equivalentes de abertura diferentes para distâncias diferentes. O teste feito com a escolha da distância de 1 m por ser da ordem do espaço de manobra previsto para o programa de desvio de obstáculo.

Figura 23 - Abrangência e Pontos Cegos do Sensor Ultrassônico

3.3.4 Programa de teste

Para a realização dos experimentos, foi escrito um programa em linguagem Java. Este programa continha funções ativáveis pelo teclado. Cada função executava uma atividade:

- Movimentar em linha reta por uma contagem de pulsos definida como

parâmetro.

- Girar em torno do seu eixo vertical por uma contagem de pulsos definida como parâmetro.
- Medir a distância informada pelo sensor ultrassônico e apresentar na console do equipamento Robodeck.
- Medir o ângulo informado pela bússola e apresentar na console do Robodeck
- Percorrer uma trajetória, de um ponto de origem até um ponto de destino, passado como parâmetro para o programa.

Durante o percurso, o Robodeck pode estar em um dos estados mostrados na Figura 24

- Estado 1: Sobre a linha inicial. O veículo está indo em direção ao objetivo
- Estado 2, 3 e 4: Desvio à esquerda. O veículo abandonou o curso anterior com uma manobra à esquerda. A manobra de contorno será feita até reencontrar a trajetória original.
- Estado 5, 6 e 7: Desvio à direita. O veículo abandonou o curso anterior com uma manobra à direita. A manobra de contorno será feita até reencontrar a trajetória original.

Figura 24 - Estados de desvio

Em cada um dos estados, a ação a ser tomada depende da informação recebida dos sensores. Além disso, o estado por si só não é suficiente para determinar a posição do veículo. O estado anterior deve ser guardado para ser retomado ao final de cada estado.

A Figura 25 mostra esta dependência e as decisões tomadas pelo programa, em uma manobra simples.

- Ponto A: O veículo detecta um obstáculo à frente. Como o lado esquerdo está ocupado (bloqueado), ele faz uma manobra à direita. A partir deste ponto ele está no estado de contorno à direita
- Ponto B e C: O veículo detecta que o lado esquerdo está livre. Como estava no estado de desvio à direita, ele vira à esquerda para continuar o contorno do obstáculo. Entretanto, como a distância até a

linha inicial é diferente de zero ($\Delta y < 0$), o estado continua como de contorno à direita.

- Ponto D: O veículo detecta que retornou à linha original e o seu lado direito está livre. Como estava no estado de contorno à direita, ele efetua um desvio à direita e encerra os estados de desvio.

Figura 25 - Dependência dos estados

3.3.4.1 Fluxogramas

Para facilitar a compreensão das decisões tomadas pelo programa, foram feitos alguns fluxogramas básicos que são mostrados nesta seção.

Quando estes fluxogramas foram transformados em código, alguns detalhes, descritos a seguir, tiveram que ser definidos para que o programa funcionasse e o Robodeck conseguisse contornar os obstáculos.

- Quando se detecta um obstáculo e se inicia o contorno do obstáculo, a decisão inicial (contornar pela esquerda ou pela direita) tem que ser

armazenada para poder retomar a trajetória quando retornar à reta original.

- Zerar (*reset*) o *encoder* das rodas a cada parada, para não ultrapassar a capacidade de contagem da variável do sistema.
- Quando retorna à reta original, verificar se o robô está se afastando do destino. Neste caso, o destino está dentro do obstáculo e deve ser considerado inatingível.
- Definir uma distância mínima para permitir manobra do Robodeck

Estes detalhes estão descritos aqui e no código fonte, mas não foram incluídos nos fluxogramas para não poluí-los em demasia.

Os fluxogramas a seguir mostram as decisões nos três estados do programa:

- Sobre a trajetória original é o estado inicial e final do robô, durante um percurso executado com sucesso.
- Contorno à esquerda é o estado que é assumido após o desvio de um obstáculo feito com um giro à esquerda. Durante este estado, o robô percorre mantendo o obstáculo à sua direita, contornando-o até atingir a trajetória original.
- Contorno à direita é o estado que é assumido após o desvio de um obstáculo feito com um giro à direita. Durante este estado, o robô percorre mantendo o obstáculo à sua esquerda, contornando-o até atingir a trajetória original.

Um ponto de interesse é a decisão que é feita no fluxograma 1 quando um obstáculo é detectado. Existe uma preferência pelo giro à esquerda mesmo sem ter testado se o lado direito está livre. Esta foi uma escolha arbitrária, sem uma justificativa técnica para isso.

Figura 26 - Fluxograma 1 (Sobre a trajetória original)

Figura 27 - Fluxograma 2 (Contorno à esquerda)

Figura 28 - Fluxograma 3 (Contorno à direita)

3.4 EXPERIMENTOS COM O ALGORITMO

Para a execução dos experimentos, algumas simplificações foram assumidas, sem perder a generalidade do método:

- Espaço Cartesiano: O ambiente em que o Robodeck navegou foi considerado como um espaço cartesiano subdividido em células de tamanho igual ao comprimento do Robodeck.
- Partida na origem do sistema: No início da missão, o Robodeck estará na origem do sistema cartesiano (coordenadas 0,0) orientado no sentido positivo do eixo X
- Chegada sobre o eixo X: O destino foi estabelecido sempre sobre o eixo X e suas coordenadas cartesianas são consideradas como parâmetros conhecidos para a missão
- Movimento quantizado: O Robodeck foi movimentado em unidades múltiplas inteiras de uma célula para facilitar o cálculo de posicionamento.
- Manobras de desvio: Todos os contornos foram feitos com giros em ângulo de 90° porque o Robodeck dispunha de 4 sensores de média distância (sensores ultrassônicos), localizados à frente, atrás, à direita e à esquerda. Com isso, contornos mais detalhados são impossíveis de fazer por falta de informação do que existe nos interstícios dos campos dos sensores.

Os testes foram feitos com o Robodeck na sala S-30 do segundo andar da Engenharia de Transportes da Escola Politécnica.

Para os obstáculos, foram usados caixas de papelão posicionados conforme o teste que estava sendo feito. Todos os obstáculos foram retangulares e seus lados se situaram sempre paralelos aos eixos cartesianos, devido à limitação no número de sensores ultrassônicos.

O controle do Robodeck foi feito por um notebook, operando em Windows 7. O notebook foi posicionado sobre a carenagem do Robodeck e se comunicava diretamente com a placa MCR, através de um conversor USB-TTL3V3.

3.4.1 Desvio de obstáculo Simples

Figura 29 - Desvio de Obstaculo Simples – trajetória esperada

O algoritmo se mostrou eficaz para fazer o contorno do obstáculo e retornar à trajetória original.

Este teste demonstrou a decisão arbitrária do programador de, em caso de ambas as alternativas (giro à esquerda ou giro à direita) estarem disponíveis, o programa dar preferência a fazer o giro no sentido anti-horário (à esquerda).

Foram feitas 4 testes bem sucedidos nesta configuração de obstáculo com o Robodeck contornando o obstáculo e atingindo o destino planejado. A posição final do Robodeck não foi exatamente sobre o ponto de destino esperado. A diferença foi atribuída ao erro acumulado nas manobras de giro e nos deslocamentos em reta.

3.4.2 Desvio de Obstáculo Côncavo

Figura 30 - Desvio de Obstáculo Côncavo

O algoritmo se mostrou eficaz para fazer o contorno do obstáculo e retornar à trajetória original.

Este teste realçou o comportamento de contorno do obstáculo, com os giros sucessivos para fazer o acompanhamento do obstáculo.

O erro acumulado nas manobras de giro fez com que a posição final do Robodeck fosse de aproximadamente 30° em relação à trajetória esperada. Com a origem e o destino distantes de 6,40m, a distância entre o ponto de chegada atingido e o ponto de chegada esperado foi de 1,70 m.

Foi feita apenas uma tentativa nesta configuração devido à falta de mecanismos de recuperação de erros do protocolo do Robodeck.

Este experimento foi de interesse particular por ser um caso que os algoritmos de campos de potencial têm dificuldade para resolver. O centro da parte côncava se caracteriza como um ponto de mínimo local e tais algoritmos podem atingir um ponto de equilíbrio neste local, conforme descrito em 2.1.3.

3.4.3 Objetivo inalcançável

Figura 31 - Objetivo Inalcançável

O algoritmo se mostrou eficaz para fazer os contornos do obstáculo e identificar o retorno à posição inicial de abandono da trajetória original. Ao atingir este ponto, o algoritmo encerra o procedimento por concluir que não há trajetória viável para se atingir o destino.

Foi feita apenas uma tentativa nesta configuração devido à falta de mecanismos de recuperação de erros do protocolo do Robodeck.

4 RESULTADOS OBTIDOS

Foram estudados tres algoritmos de desvio de obstáculos e um deles, o do Inseto 2, foi escolhido e realizado em um programa Java.

Foram feitos testes com os sensores ultrassônicos do Robodeck que se mostraram confiáveis para baixas distâncias (menor que 1,5m), o que foi suficiente para o algoritmo do Inseto 2.

Foram feitos tres tipos de experimentos para validar o algoritmo: um obstáculo simples, um obstáculo convexo e um obstáculo intransponível. Em todos eles o algoritmo atingiu o resultado esperado, fazendo o robô contornar o obstáculo e alcançar o destino pretendido ou concluir pela não viabilidade de alcançar o destino.

Em nenhum dos testes a posição final do Robodeck foi exatamente em cima do ponto de destino. As diferenças eram tanto maiores quanto a distância do destino e o número de giros necessário para contornar o obstáculo.

Os experimentos trouxeram um conhecimento melhor sobre a programação e as capacidades e limitações do Robodeck. Em vista de algumas limitações, alguns testes foram replanejados para se adaptarem a estas restrições.

O principal objetivo, que era verificar a eficiência do algoritmo de desvio de obstáculos, foi atingido. O algoritmo do Inseto-2 se mostrou eficaz para o desvio e contorno de obstáculos que não estavam mapeados previamente. Os erros obtidos se mostraram dentro do esperado a partir dos testes dos sensores, de percursos em linha reta, acurácia da odometria e acurácia angular.

Alguns testes não foram repetidos em quantidades maiores por limitação do Robodeck. O protocolo de comunicação entre o notebook e o MCR é muito pobre e não prevê recuperação de falhas. Assim, sempre que algum erro ocorre na comunicação, o teste tem que ser abortado e reiniciado com uma nova conexão ao Robodeck.

Nos testes mais complexos e longos, a possibilidade de ocorrência destas falhas de comunicação se torna maior. Por esta razão, os testes com um número maior de obstáculos e de contornos foram impossibilitados.

A situação ideal para o término da trajetória seria que o veículo ficasse o mais próximo possível do destino, mesmo com o desvio de diversos obstáculos. O acúmulo nos erros de giro e movimentação foi um fator determinante para não se conseguir atingir este resultado.

A cada manobra de giro, o erro angular era acumulado. Assim, quanto maior o número de manobras de giro e distância percorrida, maior foi o erro final do posicionamento do Robodeck.

O algoritmo se mostrou eficaz em uma situação que o algoritmo de campo vetorial é incapaz de resolver: Um obstáculo convexo, que gera um ponto de mínimo intermediário, foi transposto pelo “inseto 2” sem problema.

A conclusão que se obtém destes experimentos é a de que o algoritmo do inseto 2 faz o desvio dos obstáculos interpostos no caminho do Robodeck com sucesso. A sua eficácia depende de sensores mais acurados que permitam ao algoritmo a sua localização mais exata, tanto em distância quanto em ângulo.

5 CONSIDERAÇÕES FINAIS

Esta dissertação pesquisou sobre alguns algoritmos de desvio de obstáculos e alguns sensores passíveis de serem usados para este fim.

Os experimentos se concentraram na plataforma Robodeck com a realização de um dos algoritmos estudados, o Inseto-2, com o apoio de sensores ultrassônicos. O número de sensores disponíveis na plataforma de teste foi um fator limitante para a qualidade das manobras executadas. No experimento realizado, os desvios foram feitos em ângulo reto, embora isso não desqualifique o teste.

Sensores mais eficientes, tais como o radar ou o lidar, seriam mais complexos para serem integrados à plataforma utilizada, o Robodeck, principalmente pela capacidade de processamento disponível. No caso do lidar, um dos sensores mais mencionados nos veículos autônomos pesquisados, o número de pontos por segundo a serem analisados tornaria impossível o seu uso no Robodeck atual.

A programação dos experimentos foi feita em linguagem Java, o que possibilita o seu transporte para diversas plataformas.

Alguns pontos de melhoria foram identificados durante e ao final da pesquisa e estão apresentados nas seções seguintes deste capítulo.

5.1 MELHORIA NA EXECUÇÃO DOS DESVIOS

A realização de percursos feita na parte experimental se utilizou de 4 sensores ultrassônicos presentes no Robodeck. Com isso, os ângulos de visão foram nos quatro eixos (frente, direita, traseira e esquerda) e limitaram os desvios a ângulos de 90°.

Um aumento na quantidade e na diretividade destes sensores daria uma melhora na detecção dos obstáculos e uma condução mais natural para o Robodeck, com manobras de contorno mais suaves.

A falta de uma fonte para realimentação do posicionamento angular também foi sentida. Com isso, o movimento de giro do Robodeck operou com um sinal de realimentação de baixa acurácia (odometria), causando erros que se acumularam com o número de contornos feitos.

A inclusão de um sensor confiável para realimentação de movimentação angular traria um ganho nas manobras de desvio.

5.2 MELHORIAS NO ROBODECK

Um dos fatores limitantes para os experimentos foi a plataforma de testes que foi utilizada. O Robodeck precisaria ser melhorado para incluir alguns recursos que aumentariam a sua usabilidade e confiabilidade.

5.2.1 Protocolo com recuperação de erros.

O protocolo de comunicação atual entre o MCR e o MAP consiste no envio de comandos para o MCR e o envio de respostas pelo MCR. Os únicos mecanismos de segurança são:

- Uma numeração de pacotes para identificar se a resposta recebida corresponde ao comando enviado
- Uma soma de verificação (*check sum*) para evitar a corrupção de dados por ruído elétrico.

Quando há algum tipo de perda de sincronismo, não há meios de se recuperar a comunicação sem uma intervenção manual que, em geral, é um reset do Robodeck. Durante os experimentos, diversas vezes isso ocorreu, interrompendo o teste e obrigando o seu reinício.

Uma sugestão para melhoria do protocolo é:

- Mensagem de ressincronismo. Sempre que uma perda de sincronismo

for detectada, as partes trocam esta mensagem para restabelecer o sincronismo

- Temporização de mensagens: Se uma mensagem não é completada após um limite de tempo, o receptor aborta a tentativa e volta à fase de espera por um início de mensagem.

5.2.2 Cão de guarda e Indicação de presença

Em diversos sistemas críticos, existe a presença de uma característica chamada de “chave do homem morto” (*dead man switch*). Esta ideia se originou nos sistemas de trem onde um maquinista poderia perder a consciência e deixar de comandar a locomotiva com a chave em aceleração. Nos sistemas eletrônicos existe a figura do “cão de guarda” (*watch dog*) que desempenha uma função similar.

A ideia é que, se houver uma falha de comunicação e ela não for restaurada em um tempo limite, o Robodeck deveria parar por segurança e não ficar executando o último movimento recebido indefinidamente.

Para executar esta função, um acréscimo teria de ser feito ao protocolo entre o MAP e o MCR:

- Sinalização de presença (*keep alive*): uma mensagem periódica trocada entre o MCR e o módulo externo que indique a existência de uma entidade que esteja no controle do sistema.

A falta ou perda desta mensagem por um tempo limite levaria o Robodeck a parar a movimentação ou assumir um comportamento de segurança (ex: ir para um abrigo conhecido).

5.2.3 Interface de comunicação mais imune a ruídos.

A interface atual é um conversor USB para sinal TTL-3V3. A conversão é feita na ponta mais próxima ao notebook. Com isso, o sinal que trafega até o MCR

está em uma tensão mais baixa e mais sujeita a ruídos.

O ambiente do Robodeck contém *drivers* de potência para acionamento dos motores, fontes chaveadas sem blindagem e motores que estão sendo ligados e desligados. Tudo isso provoca um grande ruído elétrico e pode ter sido a causa de tantas perdas de sincronismo na comunicação.

Uma possibilidade é usar uma comunicação mais imune ao ambiente ruidoso (ex: RS-485 ou mesmo RS-232) e fazer a conversão para o nível elétrico do ARM (3.3V) mais perto do ponto de consumo. Isso reduz a captação de ruído.

5.2.4 Parada de Emergência

O Robodeck possui sensores infravermelhos de curto alcance. Se usados juntamente com os sensores ultrassônicos, podem facilitar a criação de um mecanismo de parada de emergência no caso de surgimento de obstáculo imprevisto e súbito (p.ex. uma pessoa passando).

Esta parada de emergência se dará quando um obstáculo fosse detectado a uma distância menor do que a margem de manobra do veículo. Após a parada de emergência, o percurso poderá ser retomado caso o obstáculo venha a ser removido.

Devido à característica de urgência no tratamento desta condição, sugere-se que esta ação seja processada dentro do MCR e informada ao MAP.

APÊNDICE A - EXPERIMENTOS COM LEGO MINDSTORM EV3

Durante a primeira fase de experimentos, foi usado um Lego Mindstorms EV3 com programas feitos em ambiente Labview, da National Instruments. Estes programas permitem criar rotinas e funções através de blocos funcionais de software conectados logicamente. O resultado desta conexão é um arquivo que é descarregado (LEGO GROUP, 2013) no *brick* (módulo de controle do Lego) e executado por ele.

A Figura 32 mostra um código usado para fazer o Lego Mindstorms EV3 executar quatro giros de 90 graus usando a bússola como informação de realimentação angular.

Figura 32 - Programa Lego Mindstorms

O Lego Mindstorms EV3 oferece os seguintes recursos:

- Central de controle (*Brick*) dotado de 4 entradas para sensores e 4 saídas para atuadores
- Programação por blocos funcionais
- Motor de tração
- Sensor ultrassônico 45504 do Mindstorms, capaz de medir distâncias entre 0,03 m e 2,55 m, com uma acurácia de $\pm 0,01$ m (LEGO SYSTEMS A/S, s.d.).
- Sensor inercial (giroscópio)

- Sensor de contato
- bússola
- Sensor de luz/cor
- Atuadores: motores de alta tração independente para cada roda, com contagem de giros. Esta contagem fornece uma informação de odometria.
- motores de baixa tração para movimentação de braços robóticos e outros dispositivos

A programação do *Brick* foi feita em um notebook onde foi instalado o programa Lego Mindstorms EV3 Home Edition, disponível para ser baixado do site da Lego :

(www.lego.com/en-us/mindstorms/downloads/software/ddsoftwaredownload/).

Em seguida o programa é transferido para o *Brick* através de uma conexão USB. O gerenciamento da transferência é feita pelo próprio programa de criação dos blocos.

Quando o programa do Lego é instalado pela primeira vez, alguns sensores não vêm instalados. É o caso do giroscópio e do sensor de ultrassom. Para obter os blocos de programação correspondentes, deve-se ir ao sítio <http://www.lego.com/en-us/mindstorms/downloads> e baixá-los. Em seguida, deve-se acessar a aba de ferramentas do programa e importar os blocos. As instruções para esta operação se encontram no sítio.

O uso do Lego Mindstorms ficou restrito aos testes de conceito devido a algumas limitações:

- Número muito limitado de portas (máximo de quatro)
- Erro nas medições de odometria e giro angular.
- Baixa disponibilidade de sensores.
- Dificuldade de manejar programas mais complexos.

A.1 ACURÁCIA ANGULAR

O programa PRECISAO_ANGULAR.EV3, localizado na pasta LEGO do disco que acompanha esta dissertação, executa uma sequencia de 4 giros de 90 graus com torque de 20%. A bússola foi usada para determinar quando o giro teria atingido o seu término.

Ao final destes giros, a diferença entre o ângulo esperado e o ângulo obtido em cada uma das medições é apresentado na tabela abaixo. Valores positivos são no sentido horário e valores negativos são no sentido anti-horário.

Ângulo Esperado	Medida #1	Medida #2	Medida #3	Medida #4	Medida #5	Erro médio
360	390	372	375	375	375	+15
-360	-365	-365	-365	-365	-365	-5

Tabela 9 - Acuracidade angular do Lego (valores em graus)

Notou-se que a velocidade do giro afeta drasticamente o resultado. Apenas para comparação, o giro de 360 graus, feito em 3 velocidades diferentes, gerou os seguintes resultados.

Experimento	Velocidade 20	Velocidade 50	Velocidade 100
1	-365	-390	-385
2	-365	-385	-385
3	-365	-385	-390
4	-365	-380	-385
5	-365	-390	-380

Tabela 10 - Diferença de angulo com variação da velocidade (em graus)

Nota: As velocidades são percentuais do torque máximo do motor de tracionamento do Lego.

A hipótese levantada é a de que a inércia do veículo, por ser muito leve, faça com que, ao detectar que atingiu o ângulo desejado, leve mais tempo para parar o

giro.

O Lego executando um giro de 360° está no arquivo Lego_Giro.mp4, que está no disco que acompanha esta dissertação.

A.2 PERCURSO EM LINHA RETA

O programa NAV_RET.A.EV3, que está no disco que acompanha esta dissertação faz o Lego Mindstorm andar em linha reta por 1,2 metros, sem nenhuma compensação.

O programa NAV_RET.A_COMP.EV3 que também esta no disco, faz o Lego Mindstorm andar em linha reta por 1,2 metros, fazendo uma compensação com o uso do giroscópio.

Ao final dos experimentos, a distância entre o ponto de chegada esperado e o ponto de chegada atingido foi medido e é apresentado na Tabela 11

Experimento	Afastamento (sem compensação)	Afastamento (com compensação)
#1	0,02	0,04
#2	0,03	0,02
#3	0	0,05
#4	0,01	0,05
#5	-0,01	0,03

Tabela 11 - Afastamento no destino (em metros)

Nota: O valor negativo foi uma convenção adotada para indicar que o veículo terminou o movimento no lado esquerdo ao passo que todos os outros testes tiveram sua parada no lado direito do alvo.

Observou-se que a compensação, ao invés de melhorar a acuracidade das medidas do Lego, piorou-a. Quando se observa o movimento do Lego em linha reta, nota-se que, mesmo sem compensação, o próprio Lego faz alguns ajustes durante o movimento. Estes ajustes seriam um indício de que o próprio programa do Lego já faz algumas compensações internas para melhorar o movimento.

Assumindo-se que haja uma compensação interna no Lego, a compensação feita com o giroscópio pode ter provocado uma sobrecompensação que pode ser a causa da piora nos resultados.

Ambos os movimentos estão nos arquivos Lego_Reta.mp4 e Lego_Reta_Comp.mp4, que estão no disco que acompanha esta dissertação.

APÊNDICE B - FÓRMULAS MATEMÁTICAS E GEOMÉTRICAS

Neste capítulo são apresentadas as expressões matemáticas e geométricas relevantes para o desenvolvimento desta pesquisa. Entretanto, devido à simplificação adotada de giros de 90° e movimentação em células correspondentes ao comprimento do Robodeck, estas expressões foram extremamente simplificadas na realização do algoritmo feitas em Java.

B.1 DISTÂNCIA ENTRE DOIS PONTOS

Sejam dois pontos P1 de coordenadas cartesianas (x1, y1) e P2 de coordenadas cartesianas (x2, y2). O cálculo da distância d (P1,P2), medida em linha reta entre os dois pontos, é dada pela expressão:

$$d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

B.2 EQUAÇÃO DA RETA QUE UNE DOIS PONTOS EM ESPAÇO PLANO

Sejam dois pontos P1 de coordenadas cartesianas (x1, y1) e P2 de coordenadas cartesianas (x2, y2). A equação da reta que passa pelos dois pontos pode ser dada pelo determinante:

$$\begin{vmatrix} x & y & 1 \\ x_1 & x_2 & 1 \\ y_1 & y_2 & 1 \end{vmatrix} = 0$$

Que resulta na expressão $(x_2 - y_2)x + (y_1 - x_1)y + (x_1y_2 - x_2y_1) = 0$

B.3 DISTANCIA DE PONTO A RETA

Sejam um ponto P_1 de coordenadas cartesianas (x_1, y_1) e uma reta de equação $Ax + By + C = 0$. A distância $d(P_1, r)$ entre o ponto e a reta é calculada pela expressão:

$$d = \frac{|Ax_1 + By_1 + C|}{\sqrt{A^2 + B^2}}$$

APÊNDICE C - COMUNICAÇÃO COM O MCR

C.1 FORMATO DAS MENSAGENS

Para a comunicação com o MCR, o formato das mensagens não está descrito no manual do Robodeck e nem é fornecido pelo fabricante. Por isso, uma parte da pesquisa se concentrou na obtenção desta especificação.

Note-se que o formato descrito neste capítulo foi deduzido a partir da monitoração das comunicações efetuadas entre o MAP e o MCR, usando a jiga (programa desenvolvido pelo fabricante para exercitar funções do produto) de testes fornecida junto com o Robodeck. Como este formato não foi fornecido pelo fabricante, alterações podem ocorrer em futuras implementações do Robodeck.

O formato geral das mensagens é o apresentado na Tabela 12.

SYN	SYN	STX	SEQ	LEN	EP1	EP2	ACAO	TAM	ID_H	ID_L	PARM	CKS	ETX
-----	-----	-----	-----	-----	-----	-----	------	-----	------	------	------	-----	-----

Tabela 12 - Formato da mensagem do MCR

Os caracteres de controle da mensagem tem o seguinte valor e significado:

- SYN – Caractere de sincronismo. Valor em hexadecimal é 0x16
- STX – Marca de início de mensagem. Valor em hexadecimal é 0x02
- ETX – Marca de final de mensagem. Valor em hexadecimal é 0x03

Os campos dentro da mensagem têm as seguintes interpretações:

- SEQ – número sequencial usado para as respostas e para as confirmações
- LEN – tamanho da mensagem, iniciando no campo seguinte ao tamanho (EP1) e terminando no último caractere antes do campo de validação (CKS)
- EP1 e EP2 são identificações das entidades que estão se comunicando. Devem ser mantidas durante a sessão de comunicação

- AÇÃO – identifica o tipo de mensagem, segundo a tabela abaixo:
 - 01 – comando
 - 02 – resposta parcial
 - 03 – resposta final
 - 04 – comando com erro
- TAM – tamanho do campo de comando. Identifica o número de caracteres que seguirão, não incluindo o caractere de validação (CKS) e a marca de final de mensagem (ETX)
- ID_H e ID_L – identificador do comando, conforme descrição da apostila de software do Robodeck, seção 3.
- PARM são parâmetros e podem estar presentes ou não, de acordo com o comando enviado
- CKS – Caractere de validação (*Check Sum*) correspondente aos *byte* hexadecimal menos significativo do resultado da soma de todos os caracteres após a marca de início (STX) e antes do próprio caractere de validação.

Para cada mensagem enviada por um dos lados envolvidos na comunicação, uma confirmação ou negação deve ser enviada para fechar a comunicação. Caso contrário, o lado que transmitiu pode efetuar uma retentativa de transmissão.

A mensagem de confirmação tem o formato mostrado na Tabela 13. O valor 0x06 corresponde ao caractere ACK (*acknowledge*) da tabela ASCII

SYN	SYN	0x06
-----	-----	------

Tabela 13 - Mensagem de Confirmação do MCR

A mensagem de negação tem o formato mostrado na Tabela 14. O valor 0x15 corresponde ao caractere NACK (*Negative Acknowledge*) da tabela ASCII.

O campo Razão carrega a causa da negação

SYN	SYN	0x15	RAZAO
-----	-----	------	-------

Tabela 14 - Mensagem de Negação do MCR

C.2 DIVERGÊNCIAS DE DOCUMENTAÇÃO

Durante o desenvolvimento surgiram algumas divergências ou omissões entre a documentação do fabricante e o comportamento real do Robodeck. Os pontos encontrados estão listados nesta seção.

C.2.1 Identificação dos sensores ultrassônicos

Na seção 2 do capítulo II da Apostila de Software do Robodeck, a identificação dos sensores ultrassônicos são as seguintes:

- 0xEE – sensor frontal
- 0xEC – sensor direito
- 0xEA – sensor esquerdo
- 0xE8 – sensor traseiro

Os valores efetivos utilizados na programação foram:

- 0x00 – sensor frontal
- 0x01 – sensor direito
- 0x02 – sensor esquerdo
- 0x03 – sensor traseiro

C.2.2 Unidade de medida da bússola

Na seção 3.4.8 do capítulo II da Apostila de Software do Robodeck, a

medida de orientação da bússola é declarada como sendo expressa em centésimos de grau.

Na realidade, os valores são enviados pelo MCR em décimos de grau.

C.2.3 Identificação dos *encoders* do motor de tração

Na seção 1 do capítulo II da Apostila de Software do Robodeck, a identificação dos *encoders* é declarado como sendo:

- 0x00 – lado esquerdo
- 0xff – lado direito

Durante os testes, foi descoberto que os valores para o MCR são:

- 0x00 – lado esquerdo
- 0x01 – lado direito

APÊNDICE D - LISTA DE DOCUMENTOS EM DISCO

Os documentos listados abaixo estão no disco que acompanha esta dissertação.

- Apostila de hardware Robodeck V1.pdf
- Apostila de software Robodeck V1.1.pdf
- User Guide LEGO Mindstorms EV3 10 All PT.pdf
- Pasta com os arquivos Java dos programas desenvolvidos
- Pasta com os programas LEGO desenvolvidos

REFERÊNCIAS BIBLIOGRÁFICAS

BEKEY, G. A., 2005. *Autonomous Robots - From Biological Inspiration to Implementation and Control*. Cambridge: MIT Press.

BORENSTEIN, J. & KOREN, Y., 1989. Real-Time Obstacle Avoidance for Fast Mobile Robots. *IEE Transactions on Systems, Man and Cybernetics*, Sep/Oct, pp. 1179-1187.

BORENSTEIN, J. & KOREN, Y., 1990. *Real-time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments*. s.l., s.n., pp. 572-577.

BORENSTEIN, J. & KOREN, Y., 1991. The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots. *IEEE Transactions on Robotics and Automation*, Jun, pp. 278-288.

BORENSTEIN, J. & KOREN, Y., 1995. Error Eliminating Rapid Ultrasonic Firing for Mobile Robot Obstacle Avoidance. *IEE Transactions on Robotics and Automation*, Fev, pp. 132-138.

CHOSSET, H. et al., 2005. *Principles of Robot Motion*. Cambridge: MIT Press.

CONTINENTAL Corporation, 2012. *ARS 30X Long Range Radar*. s.l.:s.n.

DEPARTAMENTO AUTÔNOMO DE ESTRADAS DE RODAGEM - RS, 2010. *Acidentalidade em Rodovias Estaduais do Rio Grande do Sul*, Porto Alegre: s.n.

EVERETT, H. R., 1989. Survey of Collision Avoidance and Ranging Sensors for Mobile Robots. In.: *Robotics and Autonomous Systems*. s.l.:Elsevier Science Publishers, pp. 5-67.

FEITEN, W., BAUER, R. & LAWITZKY, G., 1994 (vol 3). Robust Obstacle Avoidance in Unknown and Cramped Environments. *IEEE International Conference on Robotics and Automation*, pp. 2412-2417.

FOX, D., BURGARD, W. & THRUN, S., 1997. The Dynamic Window Approach to Collision Avoidance. *IEEE Robotics & Automation Magazine*, Março, pp. 27-33.

KINSLER, L. E., FREY, A. R., COPPENS, A. B. & SANDERS, J. V., 2000. *Fundamentals of Acoustics*. s.l.:John Wiley & Sons.

LEGO GROUP, 2013. *Mindstorms EV3 - Guia do Usuário*. s.l.:s.n.

LEGO SYSTEMS A/S, s.d. *Ultrasonic Sensor*. s.l.:s.n.

MÁGORI, V. & WALKER, H., 1987. Ultrasonic Presence Sensors with Wide Range and High Local Resolution. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, Mar, pp. 202-211.

SIMMONS, R., 1996. The Curvature-Velocity Method for Local Obstacle Avoidance. *International Conference on Robotics and Automation*, Abril, pp. 3375-3382.

THRUN, S. et al., 2006. Stanley: The Robot that Won the DARPA Grand Challenge. *Journal of Field Robotics*, Issue 23(9).

URMSON, C. et al., 2009. Autonomous Driving in Traffic: Boss and the Urban Challenge. *AI Magazine*, Summer, pp. 17-28.

VELODYNE Inc., s.d. *User's Manual and Programming Guide*. s.l.:s.n.

XBOT, 2011. *RoboDeck - Apostila Mecatronica (versao 1.0)*. s.l.:s.n.

XBOT, 2011. *ROBODECK - Manual do Usuário (ver. 1.0)*. s.l.:s.n.