DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE ENGENHARIA DE COMPUTAÇÃO

Rafael Devitte

ROVER SIM - PLATAFORMA DE VALIDAÇÃO DE ALGORITMOS APLICADOS A VEÍCULOS AUTÔNOMOS PARA DESVIO DE OBSTÁCULOS

Santa Cruz do Sul
2017

Rafael Devitte

ROVER SIM - PLATAFORMA DE VALIDAÇÃO DE ALGORITMOS APLICADOS A VEÍCULOS AUTÔNOMOS PARA DESVIO DE OBSTÁCULOS

Trabalho de Conclusão de Curso II apresentado ao Curso de Engenharia de Computação da Universidade de Santa Cruz do Sul para obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Me. Márcio A. Pacheco

Santa Cruz do Sul

2017

AGRADECIMENTOS

Agradeço imensamente a minha família, meu pai Nelvo, minha mãe Altair, minhas irmãs Vanessa e Diéssica, e a minha namorada Mônica. Estes que sempre me incentivaram, apoiaram e auxiliaram. Que apesar das saudades causadas pela minha ausência, sabiam que era necessário para concluir esta importante etapa.

Aos amigos e colegas, que tornaram felizes os dias de convivência e auxiliaram das mais diferentes maneiras para o cumprimento desta jornada.

Aos professores e funcionários da UNISC, pelo conhecimento e experiências compartilhadas, que contribuíram durante todos estes anos para meu crescimento e amadurecimento.

Agradeço a todos com os quais convivi nestes anos, é graças a todos que me tornei quem seu hoje.

RESUMO

O desenvolvimento de veículos autônomos é um tema contemporâneo, que trata da disponibilidade de facilidades para os condutores e minimização dos problemas de trânsito gerados pelos mesmos. Muitas das pesquisas na área visam resolver problemas relacionados com a locomoção do veículo em ambientes complexos, que podem conter obstáculos estáticos e dinâmicos. Este trabalho apresenta o desenvolvimento de uma plataforma de validação de algoritmos de desvio de obstáculo, aplicados a veículos autônomos. O objetivo é auxiliar na definição da melhor estratégia de desvio de obstáculos para um determinado ambiente, além de auxiliar na definição dos sensores a serem utilizados. Para as validações foram implementados os algoritmos de desvio de obstáculo Inseto Tangencial e Campos Potenciais Artificiais, além da implementação de simulação de sensores ultrassônicos e *laser*. Os testes realizados possuíram como objetivo verificar o comportamento dos dois algoritmos em diferentes ambientes, afim de identificar, a partir de parâmetros como o número de colisões, destinos alcançados e deslocamento realizado, qual dos algoritmos realizou o percurso definido apresentando os melhores resultados. O algoritmo Inseto Tangencial conseguiu realizar todos percursos definidos nos ambientes em que atuou, enquanto o algoritmo Campos Potenciais Artificiais, em um dos ambientes, não conseguiu realizar o percurso. Quanto aos sensores, os sensores *laser* apresentaram resultados superiores aos sensores ultrassônicos quando se utilizou o algoritmo Inseto Tangencial. Na utilização do algoritmo Campos Potenciais Artificiais, ambos sensores demonstraram um desempenho semelhante.

Palavras-chave: Veículos Autônomos, Plataforma de Validação, Desvio de Obstáculo.

ABSTRACT

The development of autonomous vehicles is a contemporary theme, which aims to provide facilities for drivers and minimize traffic problems generated by them. Many of the research in the area aim at solving problems related to vehicle locomotion in complex environments that can contain static and dynamic obstacles. This paper presents the development of a validation platform to obstacle avoidance algorithms, applied to autonomous vehicles. The goal is to help define the best obstacle avoidance strategy for a given environment, and assist in the definition of sensors to be used. For the validations were implemented the Tangent Bug and Artificial Potential Fields obstacle avoidance algorithms, in addition to the implementation of simulation of ultrasonic and laser sensors. The tests performed had as objective to verify the behavior of the two algorithms in different environments, in order to identify, from parameters such as the number of collisions, targets reached and displacement performed, which of the algorithms performed the defined path presenting the best results. The Tangent Bug algorithm was able to perform all defined paths in the environments in which it acted, while the Artificial Potential Fields algorithm, in one of the environments, was not able to perform the course. As for the sensors, the laser sensors presented superior results to the ultrasonic sensors when the Tangent Bug algorithm was used. In the use of the algorithm Artificial Potential Fields, both sensors demonstrated a similar performance.

Keywords: Autonomous Vehicles, Validation Platform, Obstacle Avoidance.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1. INTRODUÇÃO

Veículo autônomo é um veículo capaz de se deslocar em um ambiente sem a necessidade de ser conduzido por uma pessoa. Além de realizar o percurso, o veículo deve ser capaz de desviar de obstáculos que existam no ambiente. Apesar de hoje não existir nenhum modelo disponível comercialmente no mercado, empresas como Google e Uber estão investindo em projetos nesta área e testes em vias públicas estão sendo realizados em cidades como Pittsburgh, no estado da Pensilvânia nos Estados Unidos, e Austin, no estado do Texas nos Estados Unidos (O GLOBO) (REVISTA AUTO ESPORTE).

Quanto às pesquisas na área de veículos autônomos, muitas delas visam resolver problemas relacionados com a locomoção do veículo em ambientes complexos, que podem conter obstáculos estáticos e dinâmicos. Para operar neste tipo de ambiente o veículo deve ser capaz de: interagir com o ambiente, recuperando dele informações para auxiliar em suas tomadas de decisão; estimar sua posição dentro do ambiente; reconhecer obstáculos; e responder, em tempo real, as situações que possam ocorrer neste ambiente (HEINEN, 2002). Sendo que a coleta de dados do ambiente ocorre através de sensores instalados no veículo.

No artigo de Pereira (2006) o autor utiliza um robô provido de um sensor de varredura laser em seus testes para navegação em ambientes semiestruturados. São utilizados controladores baseados na abordagem reativa para testar diferentes métodos de desvio de obstáculos.

No artigo de Gonçalves (2011) é apresentado o desenvolvimento de um veículo autônomo guiado por um sistema de navegação baseado em GNSS (*Global Navigation Satellite System*). Após a implementação de todos os módulos, o veículo é capaz de atingir coordenadas geográficas informadas em uma estação de monitoramento e transmitidas através de comunicação sem fio.

Ferramentas computacionais, como simuladores e plataformas de validação, tem um papel importante no desenvolvimento de veículos autônomos. Estas ferramentas são utilizadas para que se possa projetar, testar e aperfeiçoar os sistemas inteligentes de controle de veículos autônomos, e depois então validá-los em ambientes reais (WOLF; SIMÕES; OSÓRIO; TRINDADE JUNIOR, 2009).

No artigo de Heinen, Osório, Heinen e Kelber (s/d) o simulador SEVA 3D é utilizado para desenvolver um sistema de controle de estacionamento utilizando veículos autônomos. A

utilização de um simulador, ao invés de um veículo real, apresenta diversas vantagens, entre elas estão o menor custo no desenvolvimento e a facilidade e segurança que a simulação proporciona, mesmo em situações extremas.

Dentre as principais informações que um veículo autônomo precisa processa está a definição do percurso que o veículo deve realizar. Deste modo, percebeu-se que caso pretenda ser desenvolvido um projeto de veículo autônomo, possuir uma ferramenta para auxiliar nas definições de estratégia de desvio de obstáculo seria de grande ajuda

Com base nas informações apresentadas, este trabalho possui como objetivo principal desenvolver uma plataforma de validação de algoritmos aplicados a veículos autônomos com ênfase na detecção e desvio de objetos. A plataforma permitirá realizar testes e validações onde, através dos resultados obtidos, será possível definir o algoritmo mais indicado para um determinado ambiente, bem como as características que os sensores de navegação utilizados no projeto deverão possuir para que haja um bom desempenho.

Como justificativa social e empresarial para o desenvolvimento deste tema, pode-se citar que os veículos autônomos poderão auxiliar na questão de segurança no trânsito. Segundo estudos do Observatório Nacional de Segurança Viária, 90% dos acidentes de trânsito que ocorrem são causados pelos próprios condutores (Observatório Nacional de Segurança Viária - ONSV). Além disso, os veículos autônomos podem proporcionar maior acessibilidade a pessoas portadoras de necessidades especiais.

Como justificativa científica para o desenvolvimento deste tema, pode-se citar o aprimoramento das técnicas de desvio de obstáculos para a utilização em sistemas de navegação de veículos autônomos. Além do auxílio na otimização dos algoritmos, a plataforma irá auxiliar na escolha do *hardware* que deverá ser utilizado em determinado projeto, através de testes que poderão ser feitos com configurações variadas.

Este trabalho está estruturado do seguinte modo: no capítulo 2, é apresentada a fundamentação teórica, contendo os principais conceitos necessários para a elaboração deste trabalho; o capítulo 3 contém dados e informações a respeito de trabalhos relacionados ao assunto abordado nesse trabalho; no capítulo 4 encontra-se as informações quanto a solução que será desenvolvida; e no capítulo 5 estão as considerações finais do trabalho.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados os principais conceitos para o entendimento do trabalho a ser desenvolvido. Alguns dos conceitos, que serão apresentados, não farão parte do escopo do trabalho, mas sua compreensão ajudará no entendimento do mesmo.

A apresentação dos conceitos, do capitulo 2.1, será feita através de uma abordagem *top-down*, ou seja, será apresentado um conceito de aplicação final e gradativamente serão apresentados conceitos mais específicos.

2.1. VEÍCULO AUTÔNOMO

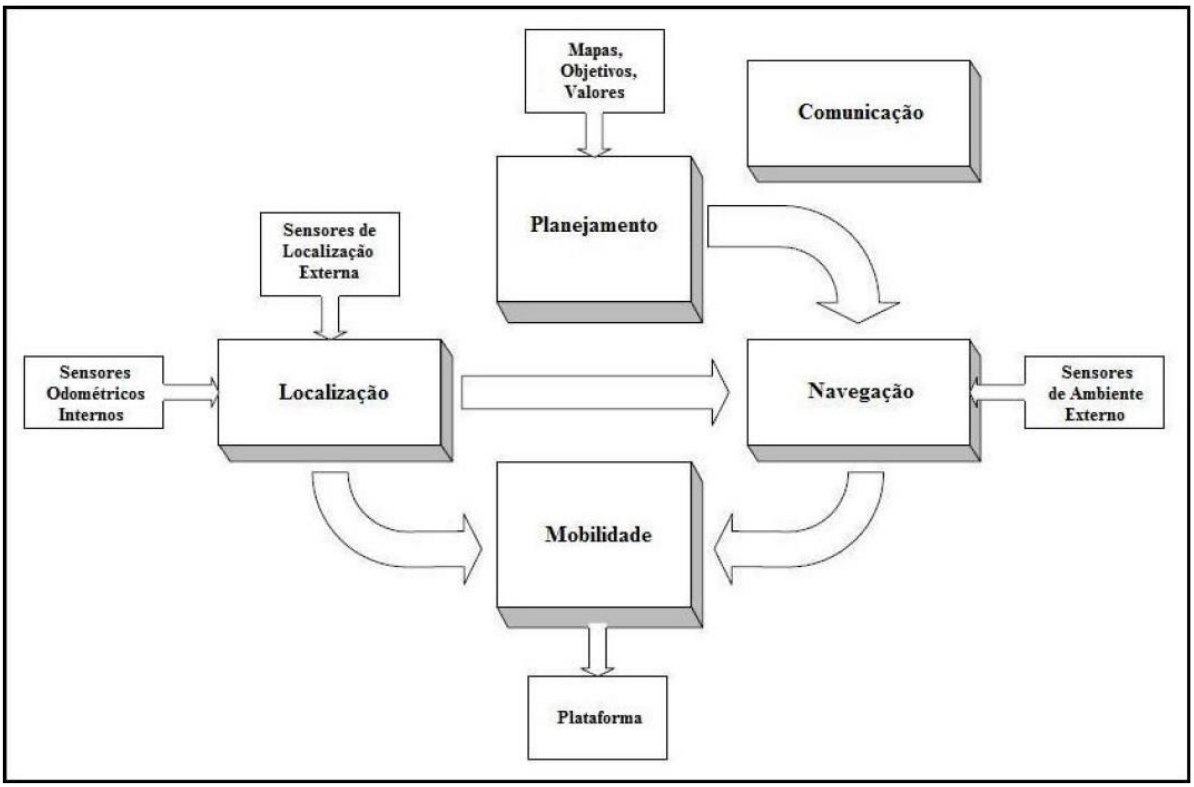
O termo “Veículo Autônomo” é atribuído a veículos dotados de sistema de controle computacional. Desta forma, um veículo deste tipo deve “Ser capaz de sensoriar, processar e responder a eventos dinâmicos e estáticos do ambiente em um tempo adequado de forma similar ou superior ao desempenho desenvolvido por condução humana” (PISSARDINI; CHIN MIN WEI; FONSECA JÚNIOR, s/d, p. 1).

Os principais elementos de projeto para veículos autônomos terrestres são (GONÇALVEZ, 2011):

- **Planejamento:** Responsável por planejar o percurso a ser seguido;
- **Navegação:** Responsável pela tomada das decisões e execução das ações;
- **Localização:** Responsável por mensurar a movimentação e o posicionamento do veículo;
- **Comunicação:** Responsável por comunicar o veículo com uma estação de monitoramento e controle;
- **Sensoriamento:** Responsável por sensoriar o ambiente de navegação, detectar obstáculos e medir a distância até estes;
- **Mobilidade:** Responsável por controlar os movimentos do veículo;

A Figura 1 demonstra o diagrama de relacionamento entre os elementos de projeto.

Figura 1 - Diagrama de Relacionamento dos Elementos de Projeto.



Fonte: Gonçalves (2011)

Nos próximos capítulos (2.1.1, 2.1.2 e 2.1.3) será explicado: como é feito o controle da movimentação do veículo; como é feito o desvio de obstáculos pelo veículo; e sensores que podem ser utilizados para detecção de obstáculos durante a movimentação.

2.1.1. MOVIMENTAÇÃO

A movimentação de um veículo autônomo utiliza duas ações similares que seriam o planejamento de rota e a manobra local. O planejamento de rota considera a localização atual do veículo e seu ponto de destino. Com essas duas localizações é gerada uma sequência de pontos de destino intermediários que devem ser percorridos para se chegar ao destino desejado. A manobra local é utilizada para o deslocamento entre um ponto intermediário e outro, evitando obstáculos que obstruam a trajetória (CHIN MIN WEI, 2015).

O veículo tem sua operação limitada por suas propriedades, sendo as principais: graus de liberdade, como ângulo máximo de esterção[1] e as direções de movimentação possíveis; características estáticas, como formato, tamanho e peso; e características dinâmicas, como limite de aceleração, velocidade e frenagem (CHIN MIN WEI, 2015).

[1] Ângulo máximo que as rodas viram para cada lado.

Em um ambiente ideal não existiriam obstáculos a serem transpostos e o ambiente seria plano. Deste modo o planejador de rota traçaria uma linha reta, entre a origem e o destino, e bastaria ao veículo percorrer este caminho para atingir o destino (CHIN MIN WEI, 2015).

Todavia, em um ambiente real podem existir obstáculos, previstos ou imprevistos, no caminho, que exijam uma rota de contorno. Obstáculos previstos, contidos em um mapa do ambiente por exemplo, podem ser considerados pelo planejador de rota, antes de iniciar o deslocamento do veículo, para gerar uma rota que evite estes obstáculos. Obstáculos imprevistos são detectados pelo veículo durante o percurso, exigindo o desvio dos mesmos (CHIN MIN WEI, 2015).

Em um ambiente real, além dos obstáculos presentes, o destino pode ser inatingível, seja por causa de obstáculos intransponíveis ou pela inexistência de rota viável dentro das condições do veículo (CHIN MIN WEI, 2015).

2.1.2. DESVIO DE OBSTÁCULOS

Durante a movimentação do veículo autônomo, é função do planejador de rota definir a trajetória por onde o veículo irá se deslocar. O planejador pode atuar de dois modos: *offline* ou *online*. O planejador *offline* atua antes do veículo iniciar sua trajetória, utilizando modelos do ambiente, como mapas, para gerar a trajetória que o veículo irá precisar percorrer de sua posição atual até o seu ponto de destino. O planejador *online* atua durante a execução da trajetória, usando dados obtidos por sensores, sendo este o responsável pelo desvio dos obstáculos imprevistos. Normalmente os veículos unem um planejamento *offline* com um planejamento *online* para poder alcançar seu destino (RIBEIRO, 2015).

O planejador *online* utiliza algoritmos de desvio de obstáculos para a construção da trajetória de desvio. Alguns algoritmos consideram um subconjunto das propriedades do veículo 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 (CHIN MIN WEI, 2015).

A seguir são apresentados alguns algoritmos e estratégias para desvio de obstáculos.

2.1.2.1. ALGORITMO INSETO (*BUG ALGORITHM*)

Este algoritmo é baseado no instinto natural apresentado por alguns insetos, como as formigas, que andam em linha reta em direção ao objetivo. Caso um obstáculo se encontre no caminho, o mesmo é contornado e posteriormente se retoma o curso ao objetivo. A

movimentação produzida por este algoritmo pode ser dividida em duas fases: a movimentação em direção ao alvo e a movimentação de contorno de borda (*boundary following*) (CHIN MIN WEI, 2015).

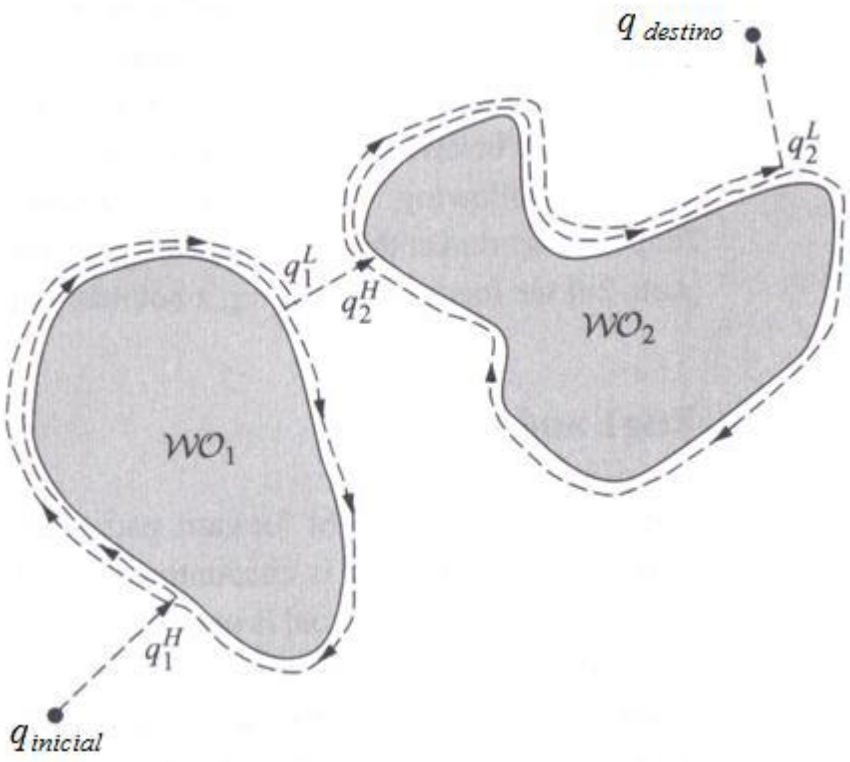
A implementação deste algoritmo supõe que o veículo seja dotado de sensores que possam detectar o obstáculo, e seus limites, e possua algum meio para determinar a distância entre a sua posição atual e o ponto de destino. É um algoritmo que garante o sucesso em chegar ao destino, desde que ele seja alcançável (CHIN MIN WEI, 2015).

A descrição do algoritmo Inseto 1 é a seguinte (CHIN MIN WEI, 2015):

1. Mover-se em direção ao destino até alcançá-lo ou encontrar um obstáculo.
2. Se um obstáculo for encontrado durante o trajeto, inicia-se um movimento de contorno. O contorno deve percorrer todo o obstáculo. Durante o contorno do obstáculo, duas informações são constantemente obtidas:
    a. Distância entre o ponto no contorno e o ponto de destino;
    b. Se, em algum ponto, o caminho até o destino fica desobstruído.
3. Quando o contorno completo for concluído, o veículo estará de volta ao ponto inicial, onde o obstáculo foi detectado. Haverá então, duas possibilidades:
    a. Existem pontos de partida dos quais é possível seguir em direção ao destino, sem ser obstruído pelo obstáculo. Neste caso o algoritmo identifica, dentre os pontos de partida possíveis, o ponto que tenha a menor distância até o destino e retoma o movimento em direção ao destino a partir deste 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.

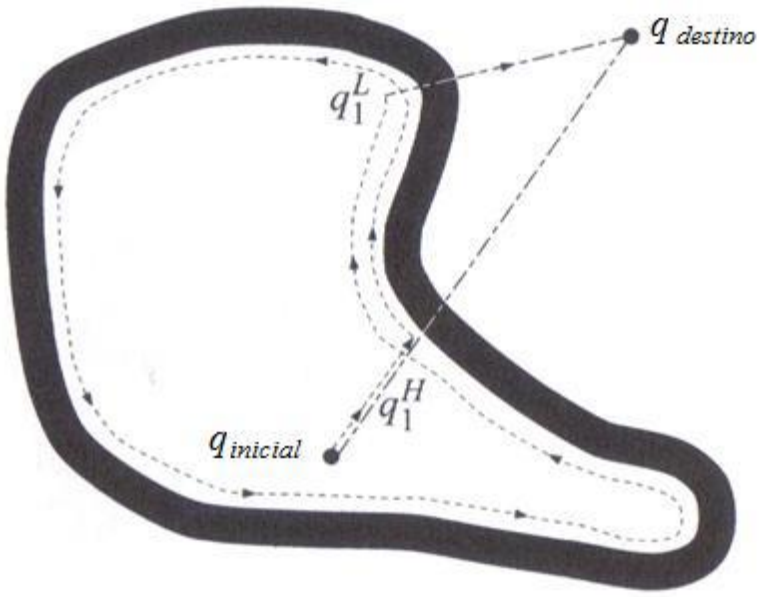
As Figuras 2 e 3 demonstram o comportamento do algoritmo, Inseto 1, quando o destino é alcançável e quando o destino é inalcançável, respectivamente.

Figura 2 - Algoritmo Inseto 1 com destino alcançável.



Fonte: Chin Min Wei (2015)

Figura 3 - Algoritmo Inseto 1 com destino inalcançável.



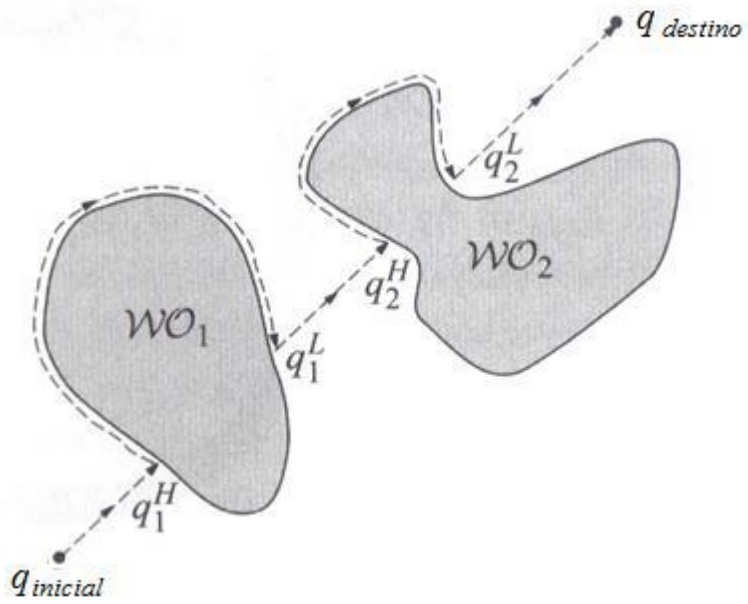
Fonte: Chin Min Wei (2015)

O algoritmo Inseto 2 é uma variação do algoritmo Inseto 1, apresentado anteriormente. A diferença é o ponto em que o veículo deixa de contornar o obstáculo e retoma seu caminho em direção ao ponto de destino (ponto de relargada). No algoritmo Inseto 1, o ponto de relargada de um obstáculo é aquele que está desobstruído e que oferece a menor distância até o destino. Para que este ponto seja obtido, o veículo precisa contornar todo o obstáculo pelo menos uma vez, afim de mapear todos os pontos viáveis para a relargada (CHIN MIN WEI, 2015).

A proposta do Inseto 2 é determinar um trajeto em linha reta, entre o ponto de origem e o ponto de destino, e que o veículo siga este percurso o mais fielmente possível. Quando um obstáculo é encontrado, o movimento de contorno é iniciado, igual ao Inseto 1. Mas, assim que encontrar um ponto no caminho de contorno que seja desobstruído e que pertença à linha de trajeto determinada inicialmente, o veículo abandona o contorno e retoma o trajeto inicial (CHIN MIN WEI, 2015).

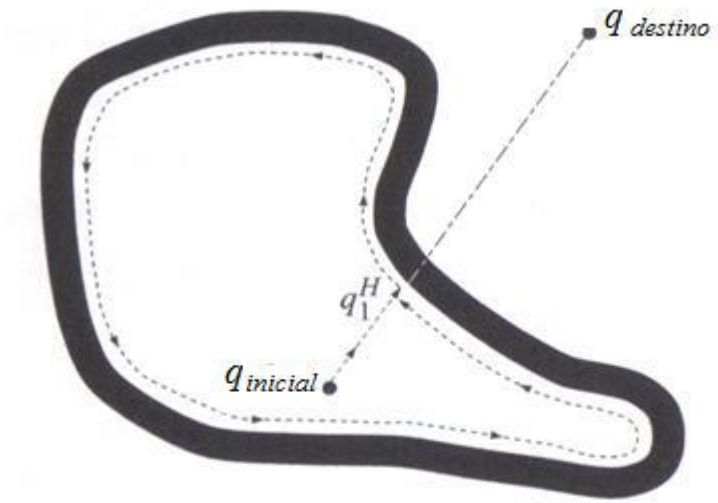
As Figuras 4 e 5 demonstram o comportamento do algoritmo, Inseto 2, quando o destino é alcançável e quando o destino é inalcançável, respectivamente.

Figura 4 - Algoritmo Inseto 2 com destino alcançável.



Fonte: Chin Min Wei (2015)

Figura 5 - Algoritmo Inseto 2 com destino inalcançável.



Fonte: Chin Min Wei (2015)

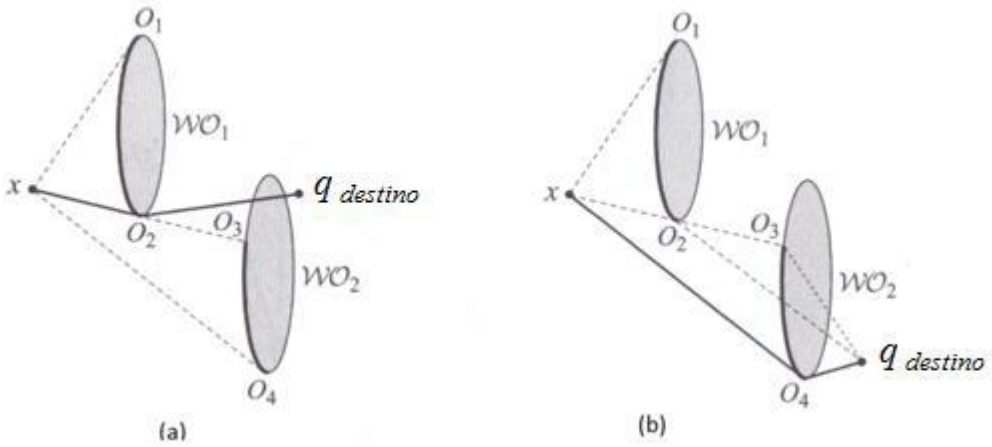
O algoritmo Inseto Tangencial (*Tangent Bug*) é mais uma variação do algoritmo Inseto e conta com a melhoria do sensor. Nos algoritmos anteriores, um sensor de contato, ou de curtíssimo alcance, bastava para fazer o algoritmo funcionar. Nesta variação, o alcance do sensor é utilizado para melhorar a eficiência do algoritmo (CHIN MIN WEI, 2015).

A implementação deste algoritmo supõe que o veículo seja dotado de um sensor com um alcance limitado R e com 360º de cobertura com alta resolução. Quando um obstáculo é detectado, seu contorno é mapeado para conhecer seus limites (CHIN MIN WEI, 2015).

O objetivo do algoritmo é fazer com que o veículo tangencie os obstáculos, sem precisar contorná-los. Para tanto, a direção de movimento apontará para o ponto mais extremo do contorno conhecido do obstáculo, este ponto também deve apresentar a menor distância para o ponto de destino (CHIN MIN WEI, 2015).

A Figura 6 apresenta o planejamento de trajetória do algoritmo, Inseto Tangencial, para dois pontos de destino distintos e considerando que o alcance do sensor englobe os dois obstáculos.

Figura 6 - Planejamento de trajetória do algoritmo Inseto Tangencial



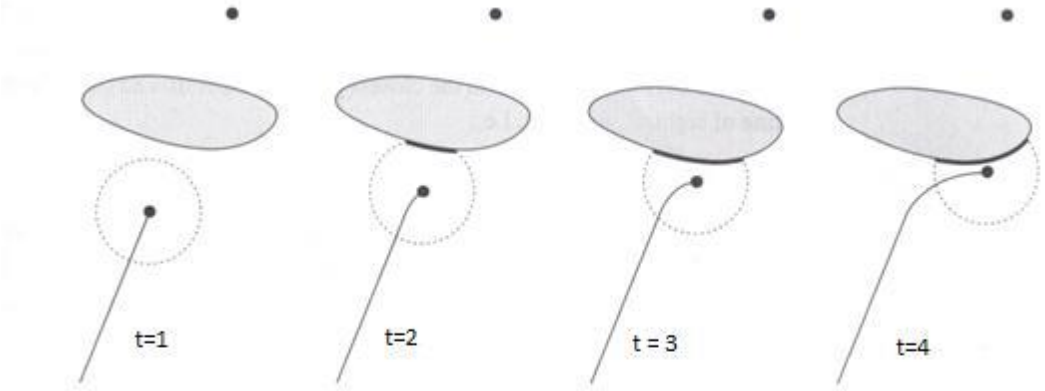
Fonte: Chin Min Wei (2015)

No planejamento para o primeiro ponto de destino (Figura 6-a), a trajetória inicialmente proposta passará por dentro do obstáculo WO2, já que o veículo não tem o conhecimento do tamanho total do obstáculo. Sendo assim a trajetória deverá ser corrigida, conforme o contorno do obstáculo WO2 for descoberto.

No planejamento para o segundo ponto de destino (Figura 6-b), a trajetória inicialmente proposta será muito próxima a trajetória efetivamente realizada. Tendo em vista que a trajetória planejada não irá colidir com nenhum outro obstáculo.

A Figura 7 ilustra o movimento de contorno de obstáculo do algoritmo Inseto Tangencial, utilizando um sensor com alcance finito.

Figura 7 - Movimento de contorno de obstáculo do algoritmo Inseto Tangencial

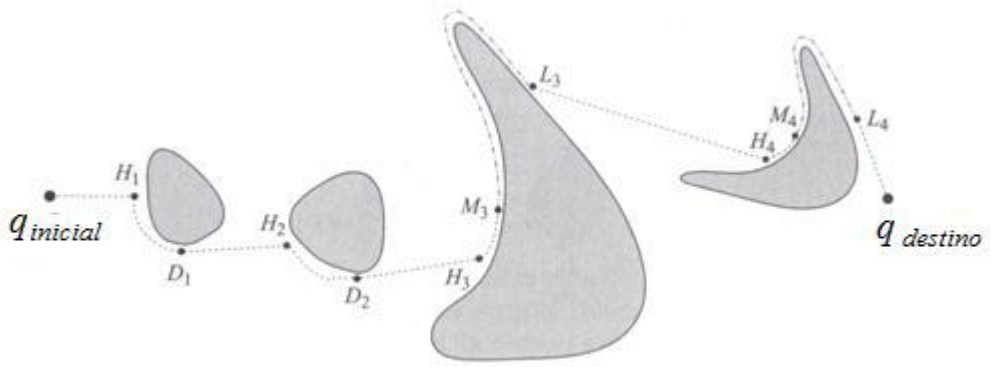


Fonte: Chin Min Wei (2015)

A Figura 7 representa a seguinte situação: enquanto o obstáculo não é detectado (t=1), o curso seguido é uma linha reta para o destino; ao detectar o obstáculo (t=2) são determinados 2 pontos de tangência: um à esquerda e outro à direita. 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 (t=3 e t=4). Em cada momento, à medida que o sensor determina o contorno do obstáculo, um novo ponto mais próximo do destino é definido e o curso é mudado para este novo ponto (CHIN MIN WEI, 2015).

A Figura 8 apresenta o comportamento do algoritmo Inseto Tangencial quando o alcance do sensor é zero, ou muito baixo. Pode-se perceber o comportamento semelhante ao dos algoritmos Inseto 1 e Inseto 2.

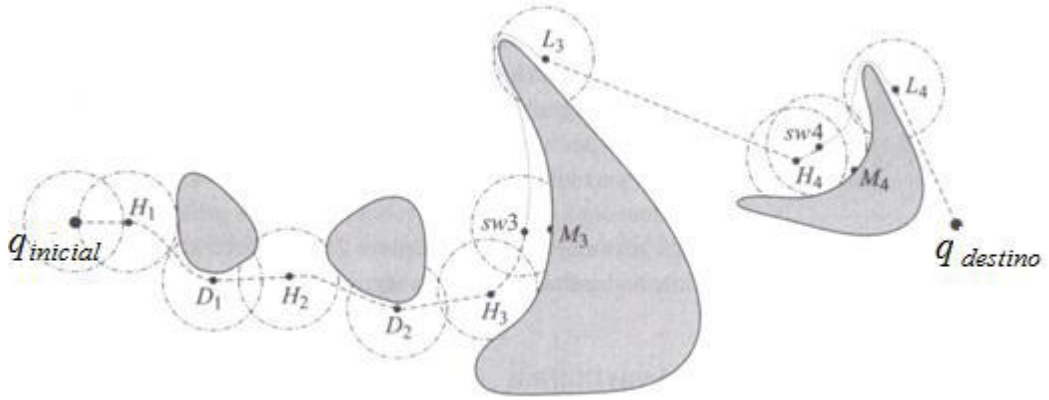
Figura 8 - Algoritmo Inseto Tangencial com sensor de alcance zero.



Fonte: Chin Min Wei (2015)

A Figura 9 apresenta o comportamento do algoritmo Inseto Tangencial quando o alcance do sensor é maior que zero, mas ainda limitado. Pode-se perceber a suavização nas mudanças de rota.

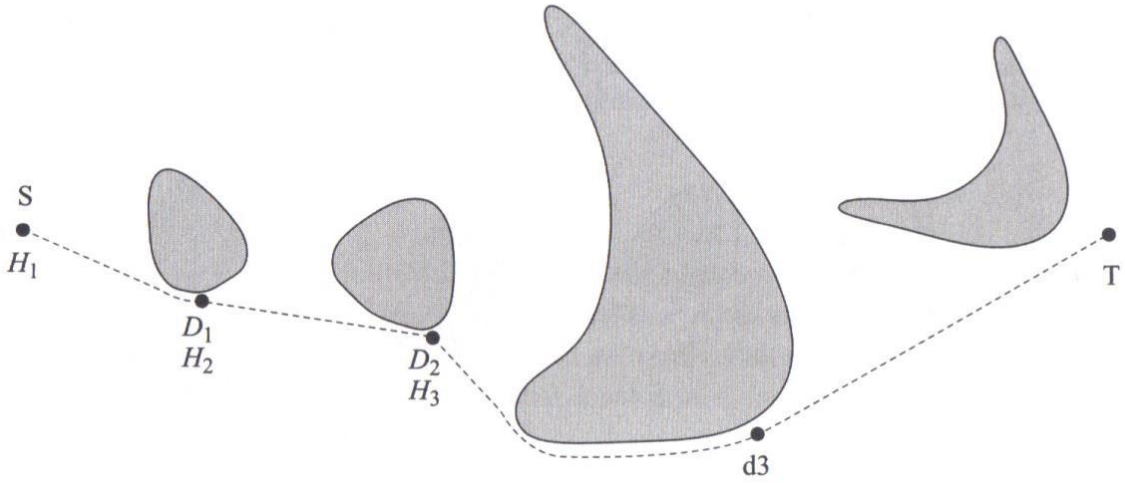
Figura 9 - Algoritmo Inseto Tangencial com sensor de alcance finito.



Fonte: Chin Min Wei (2015)

A Figura 10 apresenta o comportamento do algoritmo Inseto Tangencial quando o alcance do sensor é infinito.

Figura 10 - Algoritmo Inseto Tangencial com sensor de alcance infinito.



Fonte: Chin Min Wei (2015)

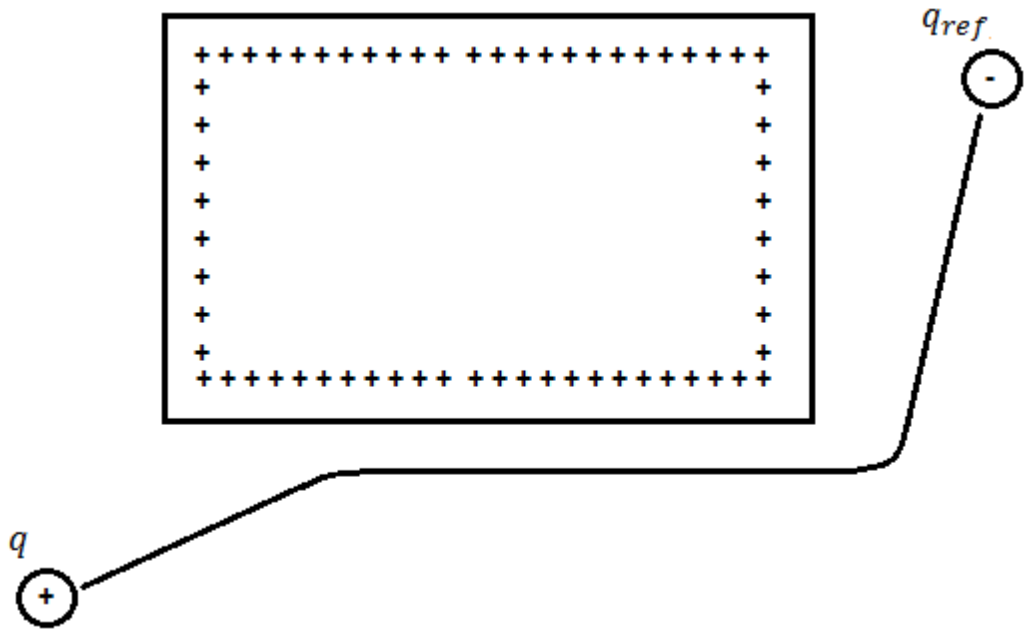
2.1.2.2. CAMPOS POTENCIAIS ARTIFICIAIS (*ARTIFICIAL POTENTIAL FIELDS*)

O algoritmo dos Campos Potenciais Artificiais, trata o veículo como uma partícula que se move em um campo potencial. Neste contexto o veículo é uma partícula com carga positiva e o ponto de destino é uma partícula com carga negativa que, consequentemente, atrai o veículo para si devido ao seu campo potencial atrativo. Os obstáculos são vistos como cargas positivas

que, por terem o mesmo sinal que o veículo, exercem um campo potencial repulsivo sobre ele, fazendo com que se afaste (RIBEIRO, 2015).

A Figura 11 representa a trajetória do veículo utilizando o algoritmo dos Campos Potenciais Artificiais.

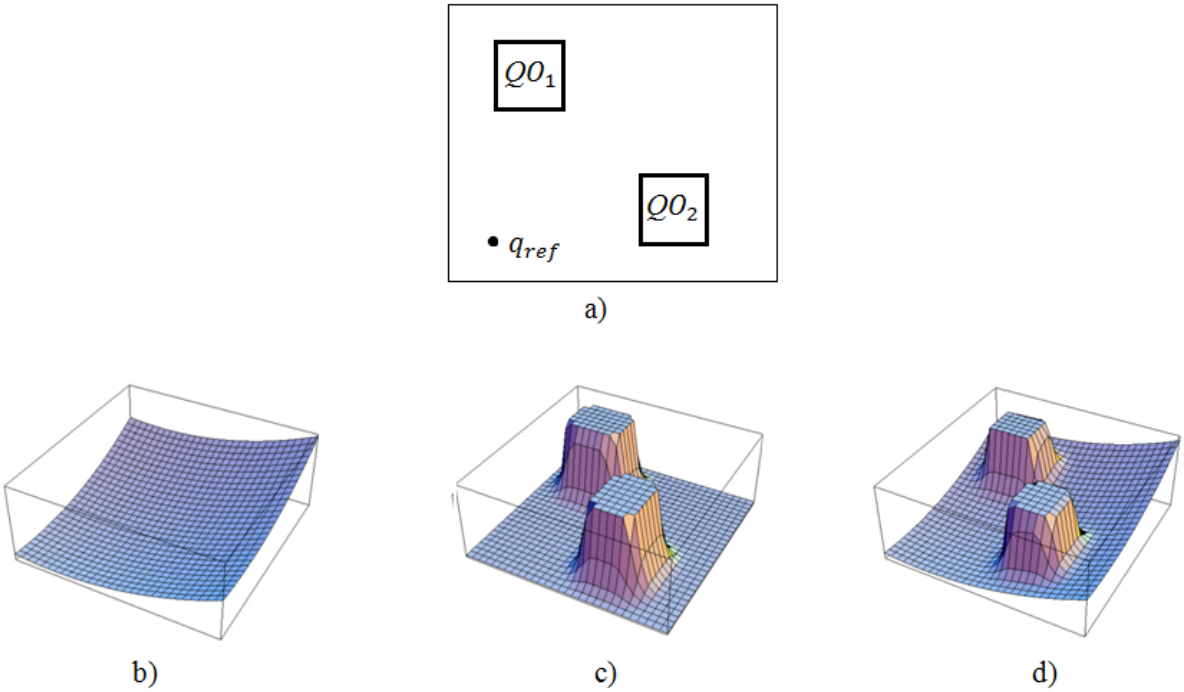
Figura 11 – Algoritmo dos Campos Potenciais Artificiais.



Fonte: Ribeiro (2015)

Este método pretende que o veículo chegue ao ponto de destino, através da soma dos potenciais repulsivo e atrativo, sem que o veículo atinja qualquer obstáculo. Essa técnica é conhecida como descida do gradiente (*Gradient Descent*) (RIBEIRO, 2015).

Figura 12 - Forças no algoritmo dos Campos Potenciais Artificiais.

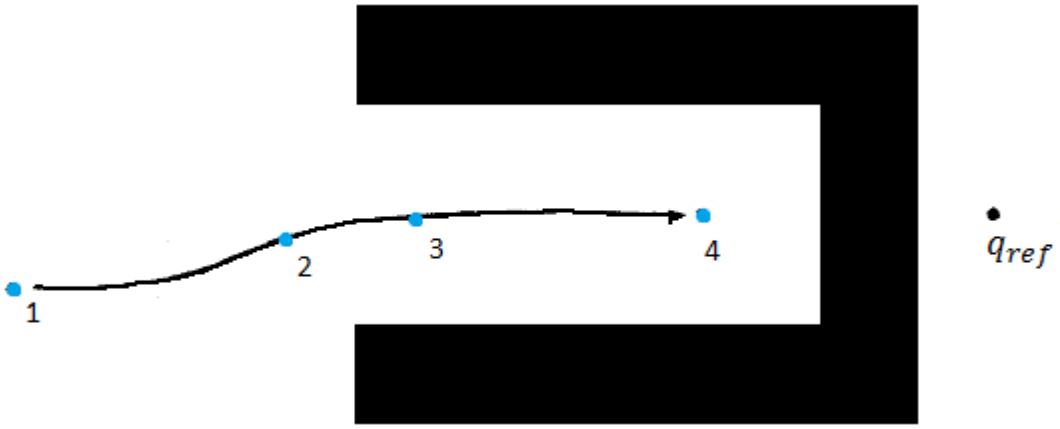


Fonte: Ribeiro (2015)

A Figura 12 representa as forças de atração e repulsão que atuam sobre o veículo. As informações foram dispostas da seguinte forma: Figura 12-a: ambiente proposto; Figura 12-b: campo atrativo do ponto de destino; Figura 12-c: campos repulsivos dos obstáculos; e Figura 12-d: soma dos Campos (RIBEIRO, 2015).

O algoritmo dos Campos Potenciais Artificiais é de simples implementação, exige pouco esforço computacional e funciona para obstáculos estáticos e dinâmicos. Contudo, existe uma falha. Dependendo da posição inicial do veículo e da disposição dos obstáculos no ambiente, pode ser que o ponto de destino não seja atingido, mesmo que exista uma trajetória possível. Os pontos no ambiente que podem fazer com que o veículo fique preso são conhecidos como Mínimos Locais. A Figura 13 representa este tipo de situação (RIBEIRO, 2015).

Figura 13 - Mínimo Local com obstáculo côncavo.

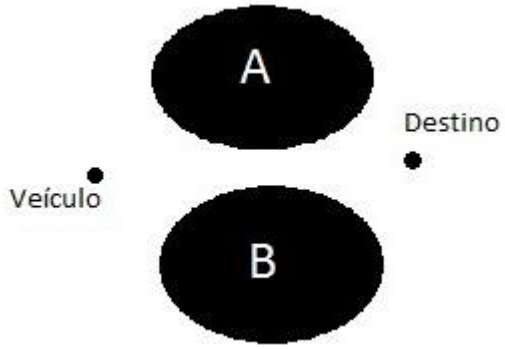


Fonte: Ribeiro (2015)

Na Figura 13, inicialmente o veículo é atraído apenas pelo ponto de destino (ponto 1), seguindo em direção a ele até que começa a ser repelido pela parte inferior do obstáculo (ponto 2). O veículo é "empurrado" para cima até que seja influenciado igualmente pela repulsão das partes inferior e superior do obstáculo (ponto 3), ele continua deslocando-se horizontalmente para o ponto de destino. Em um momento, o veículo começa a ser repelido, também, pela parte central do obstáculo, que empurra o veículo no sentido contrário do ponto de destino. A ação de todas estas forças acaba fazendo com que, em certo momento, o gradiente resultante, do campo atrativo, seja nulo em um local que não é o ponto de destino e o veículo para (ponto 4) (RIBEIRO, 2015).

O problema de Mínimo Local não acontece apenas com obstáculos côncavos. A Figura 14 apresenta mais um problema de Mínimo Local. A força de repulsão dos dois obstáculos não permite que o veículo consiga alcançar seu destino e o deixa "preso".

Figura 14 - Mínimo Local com dois obstáculos.



Fonte: Autor

2.1.3. SENSORES

Na estrutura de um veículo autônomo, a parte de sensoriamento pode ser dividida em três grupos básicos de sensores: sensores para reconhecimento de rota; sensores para reconhecimento de obstáculos; e sensores de navegação. É a atuação destes três grupos de sensores que permitem que a rota traçada seja seguida com segurança (JUNG; OSÓRIO; KELBER; HEINEN, s/d).

Tendo em vista o objetivo deste trabalho, aqui serão abordados apenas os sensores para reconhecimento de obstáculos. "Através dos sensores de objetos, como *scanners* a laser, ultrassom, radar e visão estereoscópica, diferentes obstáculos podem ser detectados, fazendo com que o veículo pare ou desvie, evitando assim colisões" (JUNG; OSÓRIO; KELBER; HEINEN, s/d, p. 1368).

A seguir é explicado o funcionamento de alguns dos sensores que podem ser utilizados na tarefa de detecção e desvio de obstáculos.

2.1.3.1. SENSOR ULTRASSÔNICO

Ultrassom é a termologia utilizada para definir as ondas sonoras com frequências acima da capacidade de audição humana, em torno de 18kHz. As principais aplicações de ultrassom dividem-se em: utilização de potência elevada ou; utilização de vibrações de pequena amplitude (MARTINS; ANDRADE, 2005). As potências elevadas de ultrassom promovem alterações no meio de propagação das ondas e suas aplicações incluem limpeza, perfuração, processos químicos, produção de emulsões, entre outros (MARTINS; ANDRADE, 2005). Tais aplicações não são o foco do presente trabalho.

A utilização de vibrações de pequena amplitude visa o efeito que meio tem sobre as ondas, como reflexão, refração, velocidade de propagação ou atenuação. Estes efeitos, através de técnicas de medição, são utilizados nas mais diversas aplicações como: determinação de módulos de elasticidade em metais; sonares; obtenção de imagens de partes do corpo humano; entre outras (MARTINS; ANDRADE, 2005).

Sensores ultrassônicos normalmente utilizam cristais piezoelétricos para emissão e recepção, estes cristais têm capacidade de converter energia mecânica em energia elétrica e vice-versa. Os transmissores ultrassônicos utilizam o efeito piezoelétrico inverso, isto é, se uma tensão senoidal é aplicada no cristal transmissor este produzirá uma deformação senoidal correspondente. Esta vibração é transmitida para as partículas no início do meio de transmissão

num movimento senoidal, causando a vibração das outras partículas até que a perturbação é transmitida ao final do meio. Esta então é detectada pelo receptor ultrassônico que utiliza o efeito piezoelétrico direto, onde a força gerada pelas ondas de pressão do meio sobre a área do cristal produz uma variação correspondente na carga e corrente (MARTINS; ANDRADE, 2005).

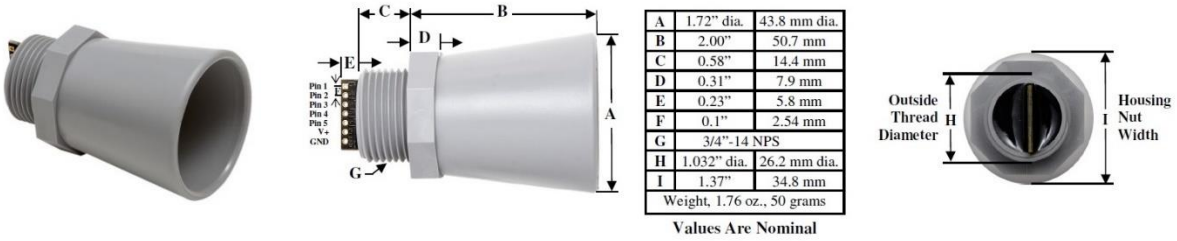
Pereira (2006) explica como obter a informação da distância de um obstáculo através de um sensor ultrassônico:

> Com um sensor ultrassônico pode-se obter a informação de distância a partir da técnica de tempo de voo (TOF, do inglês *Time of Flight*). Mede-se o tempo entre a emissão do pulso e o retorno do eco. Este valor de tempo é multiplicado pela velocidade do som no ar para se obter um valor de distância. Entretanto, esta distância equivale ao dobro da distância real ao obstáculo. Logo, basta dividir o resultado obtido por dois para determinar a distância ao obstáculo (PEREIRA, 2006, p. 22-23).

2.1.3.1.1. MB7363 MAXSONAR

O sensor MB7363, da série HRXL-MaxSonar® - WR™, produzido pela empresa MaxSonar, é um sensor ultrassônico que pode ser utilizado para a medição de distância de obstáculos. Neste trabalho suas características serão utilizadas como base para o sensor ultrassônico que será implementado e utilizado na plataforma de validação a ser desenvolvida. A Figura 15 ilustra o sensor e suas dimensões.

Figura 15 - Sensor ultrassônico MB7363.



A	1.72" dia.	43.8 mm dia.
B	2.00"	50.7 mm
C	0.58"	14.4 mm
D	0.31"	7.9 mm
E	0.23"	5.8 mm
F	0.1"	2.54 mm
G	3/4"-14 NPS	
H	1.032" dia.	26.2 mm dia.
I	1.37"	34.8 mm
Weight, 1.76 oz., 50 grams		

Values Are Nominal

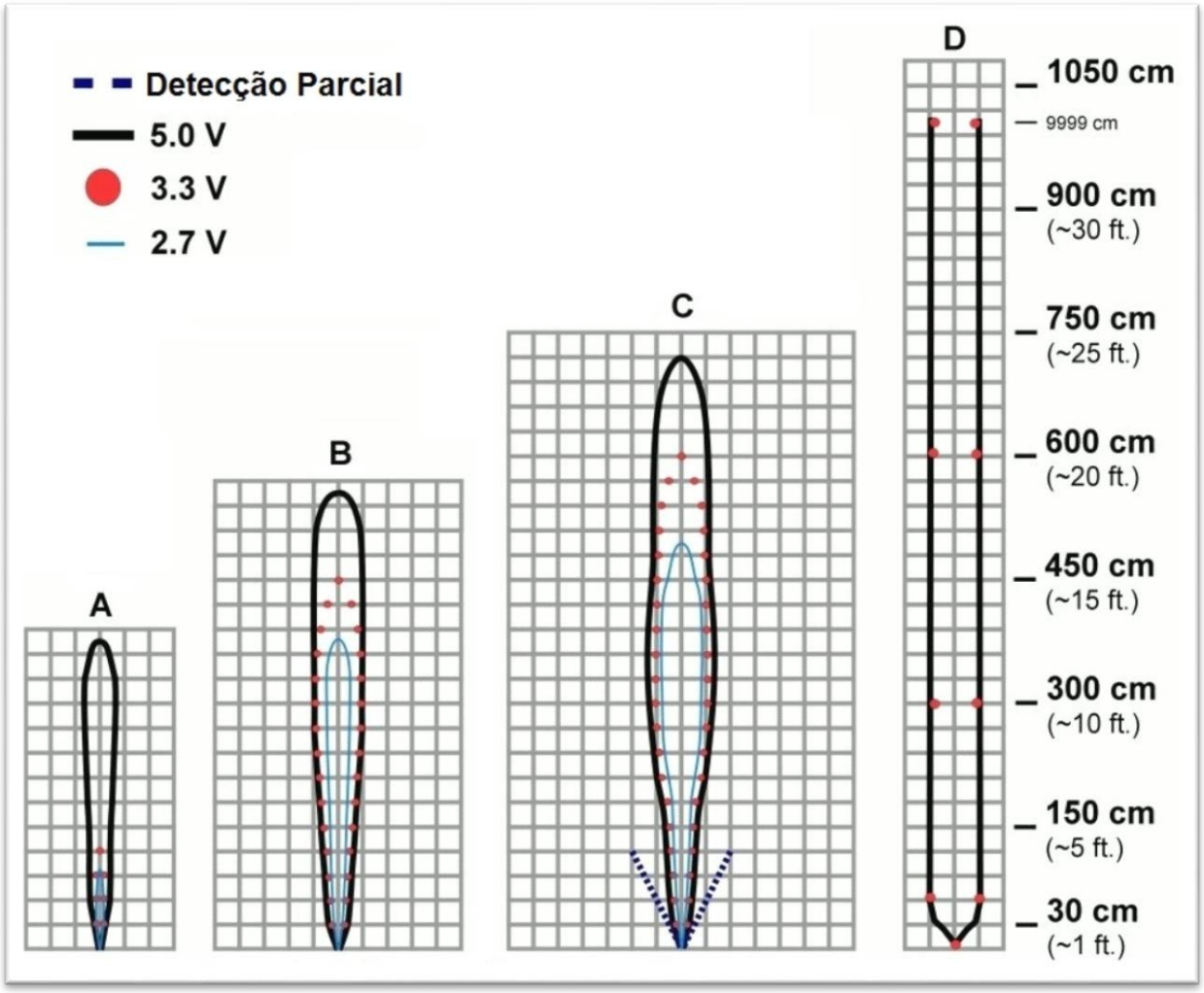
Fonte: HRXL-MaxSonar-WR Datasheet.

O sensor MB7363 apresenta as seguintes características:

- Alcance de 50cm à 10m. Objetos a uma distância menor que 50cm do sensor terão a indicação de estarem a 50cm;
- Comunicação serial RS232;
- Trabalha com as tensões 2.7V, 3.3V ou 5.0V;
- Precisão de 1mm;

A Figura 16 demonstra a área de detecção do sensor, para diferentes objetos e com variadas tensões.

Figura 16 - Área de detecção do sensor MB7363.



Fonte: HRXL-MaxSonar-WR Datasheet.

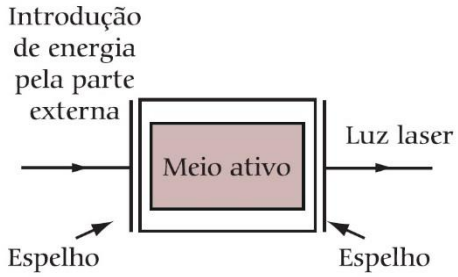
Na Figura 16 todas as medições são apresentadas em grades onde cada quadrado possui 30cm de lado. As Figuras 16-A, 16-B e 16-C representa a capacidade de detecção de cilindros com diâmetro de 6.1mm, 2.54cm e 8.89cm, respectivamente. A Figura 16-D representa a capacidade de detecção de uma placa com 27.94cm de largura, esta placa fica virada para o sensor, e é deslocada na frente dele da esquerda para a direita, para demonstrar o alcance do sensor.

Para determinar a capacidade de detecção de uma pessoa, conforme indicação do fabricante do sensor pode-se considerar a Figura 16-B, assim sendo, este sensor é capaz de detectar uma pessoa a, aproximadamente, 5.5m de distância.

2.1.3.2. SENSOR LASER

Laser (*Light Amplification by Stimulated Emission of Radiation*) é um feixe de luz direcional de alta intensidade. A criação de um raio-laser dá-se do seguinte modo: um meio ativo recebe estímulos de uma fonte externa (uma descarga elétrica, outro laser, etc.) deixando a maioria dos átomos em seus estados excitados. Essa excitação gera a emissão espontânea de fótons a partir desses átomos, adicionando mais luz à porção já existente. Esses fótons se refletem em espelhos que ficam ao redor do meio, voltando para a amostra e provocando mais emissão estimulada. Uma porção dessa luz emerge do sistema, através de uma abertura em um dos espelhos, constituindo o feixe da luz *laser* (BAGNATO, 2001). A Figura 17 ilustra de modo simplificado o processo para a emissão de um *laser*.

Figura 17 - Esquema simplificado das partes que constituem um *laser*.



Fonte: *Bagnato* (2001)

Os *lasers* são utilizados nas mais diversas áreas e com diferentes finalidades como: cirurgias; leitor de código de barras; depilação; impressoras; reprodutores de CD; etc. Os *lasers* também podem ser utilizados em sensores para a medição de distância (Pereira, 2006). Estes sensores *laser* para determinação de distância são os que serão abordados neste trabalho.

Os sensores *laser* são capazes de determinar a distância de um objeto através de duas técnicas distintas: técnica de Diferença de Fase; e técnica de Tempo de Voo (TOF, do inglês *Time Of Flight*) (PEREIRA, 2006).

A técnica de Diferença de Fase utiliza a defasagem (deslocamento de fase) entre dois feixes de laser, um emitido e outro refletido, contudo esta técnica só pode ser utilizada para medir distância com no máximo metade do comprimento de onda do *laser* (PEREIRA, 2006).

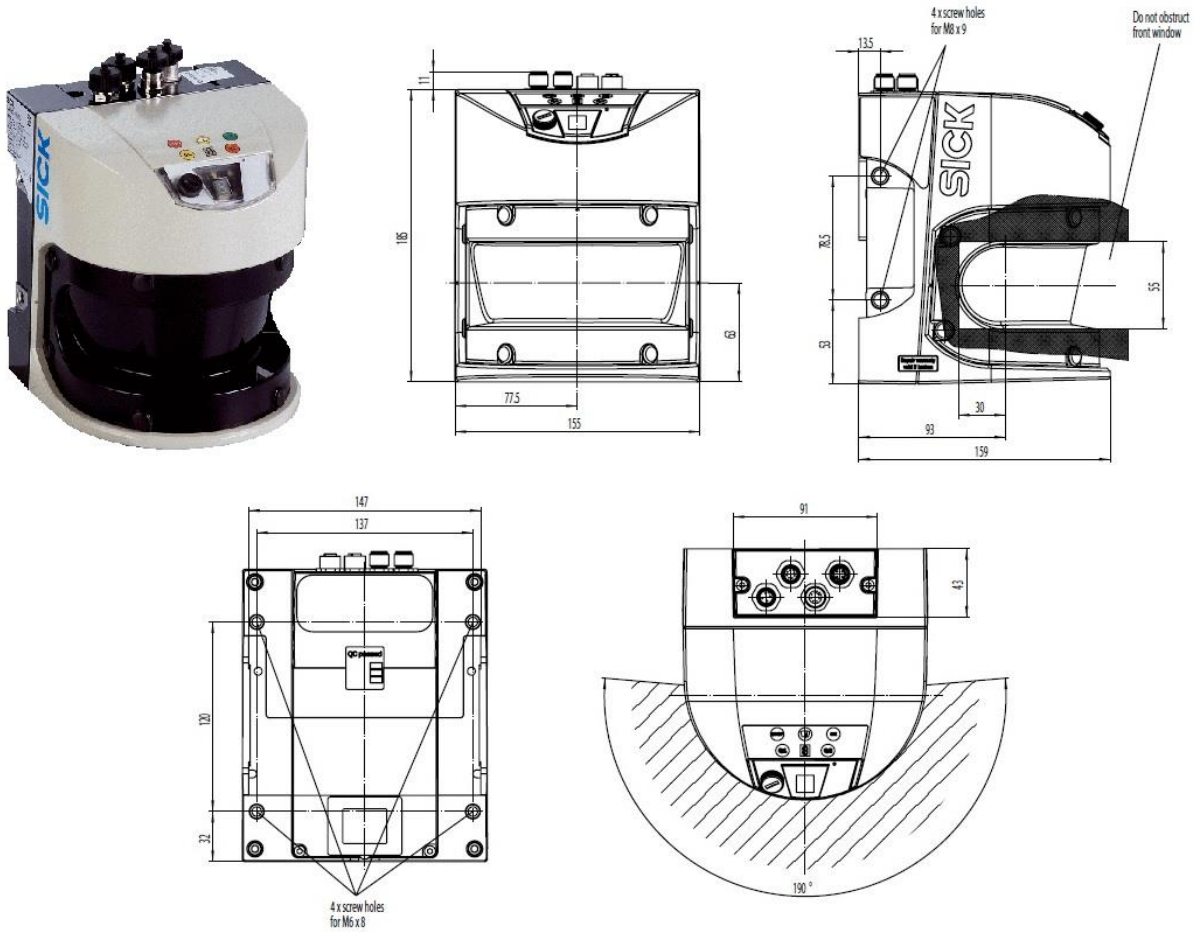
Na técnica de Tempo de voo um *laser* é emitido pelo transmissor e ao encontrar uma superfície, retorna até o receptor. A medida de distância entre o sensor e a superfície refletora é determinada pelo tempo que este levou para sair do transmissor até chegar ao receptor. Este valor de tempo é multiplicado pela velocidade do *laser* no ar (velocidade da luz) para se obter

um valor de distância. Entretanto, esta distância equivale ao dobro da distância real ao obstáculo. Logo, basta dividir o resultado obtido por dois para determinar a distância ao obstáculo (PEREIRA, 2006).

2.1.3.2.1. LMS511 SICK

O sensor LMS511 PRO *Standard Resolution*, da série LMS5xx, produzido pela empresa Sick, é um sensor *laser* que pode ser utilizado para a medição de distância de obstáculos. Neste trabalho suas características serão utilizadas como base para o sensor *laser* que será implementado e utilizado na plataforma de validação a ser desenvolvida. A Figura 18 ilustra o sensor e suas dimensões.

Figura 18 - Sensor *laser* LMS511.



Fonte: LMS5xx Laser Measurement Technology - Product Information.

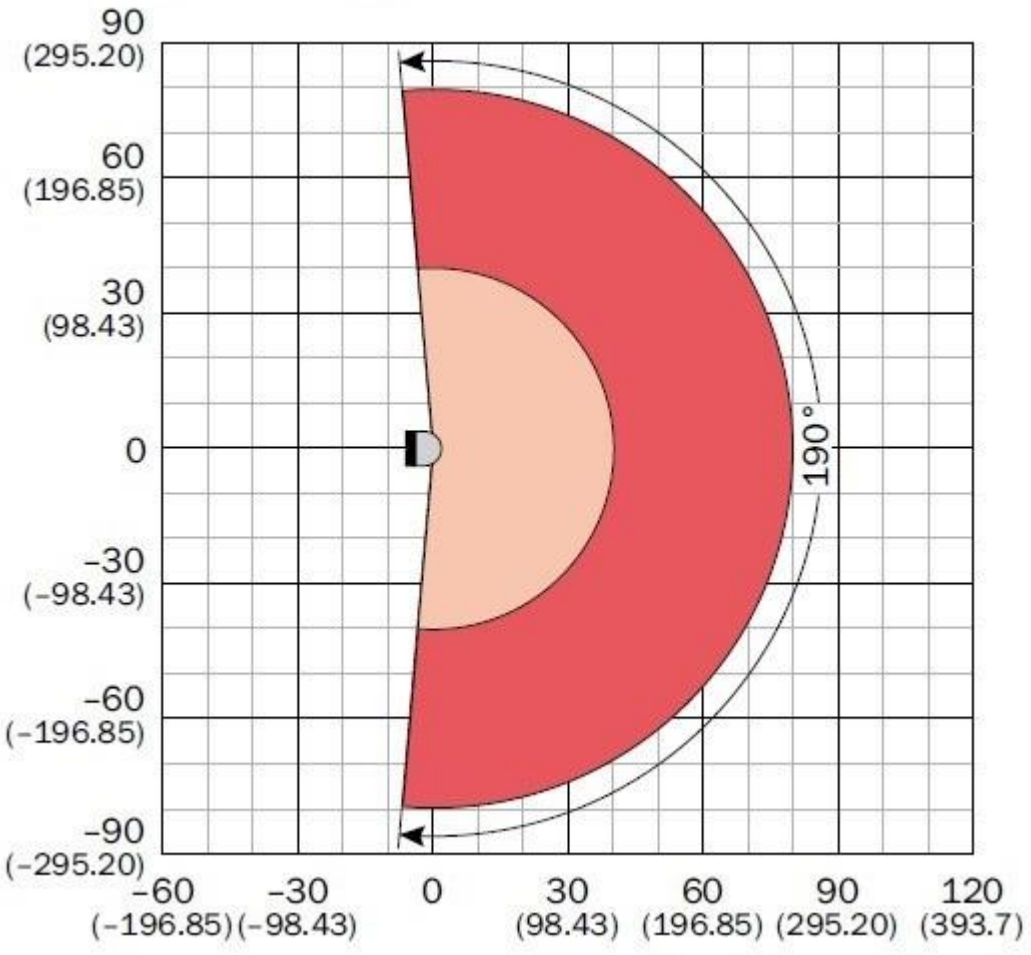
O sensor LMS511 PRO *Standard Resolution* apresenta as seguintes características:

- Alcance de 0m à 80m. O alcance para objetos com baixa reflexão (10%) é de 40m;
- Ângulo de visão de até 190º;
- Precisão angular de 0.167º à 1º;

- Trabalha com frequências de 25Hz à 100Hz;
- Comunicação serial RS232 ou RS422;
- Trabalha com tensão de 24V;
- Trabalha em ambientes com temperaturas de -30ºC à +50ºC;

A Figura 19 demonstra a área de detecção do sensor.

Figura 19 - Área de detecção do sensor LMS511.



Fonte: LMS5xx Laser Measurement Technology - Product Information.

Na Figura 19 há dois valores para cada linha do gráfico, o primeiro valor representa metros e o segundo valor, entre parênteses, representa pés. O arco maior, em vermelho forte, representa o alcance máximo do sensor, que equivale a 80m. O arco menor, em vermelho fraco, corresponde ao alcance máximo do sensor para objetos com taxa de reflexão de até 10%, que equivale a corresponde a 40m.

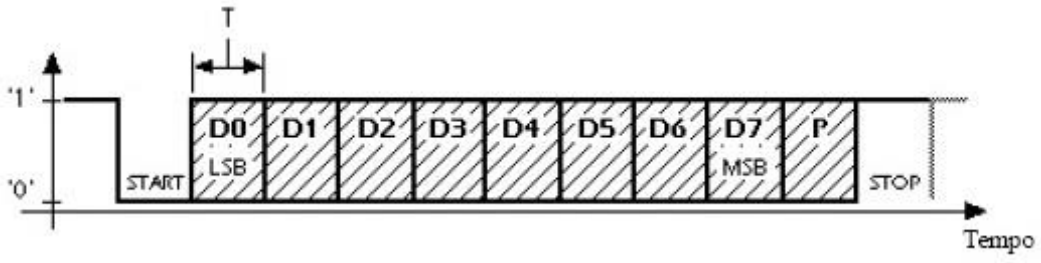
2.1.3.3. COMUNICAÇÃO RS232

A necessidade de explicar a comunicação RS232 dá-se pelo fato de que os dois sensores citados anteriormente, capítulos 2.1.3.1.1 e 2.1.3.2.1, utilizam este tipo de comunicação. E essa comunicação também será simulada na plataforma a ser desenvolvida.

A comunicação RS232 é uma interface serial, ponto a ponto (2 nós) e de baixa velocidade[2]. Foi definida em 1950, com o objetivo de padronizar a comunicação com equipamentos eletrônicos da rede, como modens e repetidores. O padrão RS232 define: especificações elétricas; interface mecânica (conectores); e descrição funcional (Pinheiro, 2011).

Quanto a parte elétrica do padrão, a faixa de tensão de -3V à -15V representa o 1 (*OFF*) e a faixa de +3V à +15V representa o 0 (*ON*). A faixa de tensão entre -3V e +3V é indefinida (Pinheiro, 2011). A Figura 20 representa a estrutura assíncrona da mensagem transmitida no padrão RS232.

Figura 20 – Mensagem assíncrona RS232.



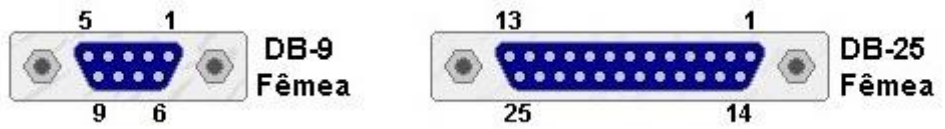
Fonte: Pinheiro 2011.

Na estrutura da mensagem (Figura 20): o primeiro bit é o *start*, para inicializar a transmissão; do segundo ao nono bit (8 bits) são os dados da mensagem; o décimo é um bit de paridade, para checar a transmissão da mensagem; o décimo primeiro bit é o *stop*, que finaliza a mensagem.

Na parte mecânica, por padrão são usados os conectores DB-9 ou DB-25. Na maioria das aplicações, quase todos os 25 pinos são desnecessários então o conector DB-9 se tornou um "padrão de fato" (Pinheiro, 2011). A Figura 21 expõe os conectores DB-9 e DB-25.

[2] Originalmente, a velocidade havia sido estipulada em até 20kbps, contudo hoje existem aparelhos que se comunicam neste padrão com velocidade de 1,5Mbps (PC16550D Universal Asynchronous Receiver/Transmitter With FIFOs).

Figura 21 - Conectores fêmea DB-9 e DB-25.



Fonte: Pinheiro 2011.

No padrão RS232, para o conector DB-9 os pinos responsáveis pela recepção e transmissão de dados são os pinos 2 e 3 respectivamente. A função de cada um dos pinos do conector DB-9, no padrãoRS232, é a seguinte:

1. *Received Line Signal Detector* (Portadora detectada)
2. *Received Data* (Recepção de dados)
3. *Transmitted Data* (Transmissão de dados)
4. *Data Terminal Ready* (Terminal de dados pronto)
5. *Signal Ground* (Terra)
6. *Data Set Ready* (Conjunto de dados pronto)
7. *Request to Send* (Pronto para enviar)
8. *Clear to Send* (Envie os dados)
9. *Ring Indicator* (Indicador de chamada)

2.2. SIMULAÇÃO

Uma plataforma de validação pode ser entendida como um simulador com funcionalidades reduzidas ou simplificadas. Deste modo, e visando o objetivo deste trabalho, faz-se necessário explicar alguns conceitos da simulação.

Segundo Pegden (1990), citado por Souza (2009, p. 1), “simulação é o processo de projetar um modelo computacional de um sistema real e conduzir experimentos com este modelo com o propósito de entender seu comportamento e/ou avaliar estratégias para sua operação”. Assim sendo, a simulação consiste em: construir modelos computacionais; descrever o comportamento do sistema; construir teorias e hipóteses considerando as observações efetuadas; e usar o modelo para prever o comportamento futuro, isto é, os efeitos produzidos por alterações no sistema ou nos métodos empregados em sua operação.

Os simuladores são ferramentas (*hardware* ou *software*) utilizados para gerar as simulações. Alguns conceitos chave precisam ser entendidos para o bom entendimento dos simuladores. Biscotto (2009) os descreve do seguinte modo:

- **Sistema**: Um conjunto de entidades que interagem ao longo do tempo para obter resultados comuns.

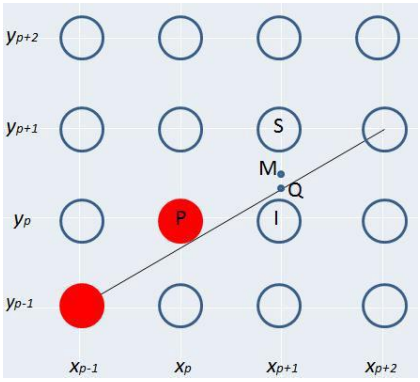
- **Modelo**: Uma representação abstrata de um sistema, geralmente contendo relações estruturais, lógicas e matemáticas que descrevem o sistema.
- **Estado de sistema**: Uma coleção de variáveis que contêm toda informação necessária para descrever o sistema em um determinado instante.
- **Entidade**: Qualquer objeto ou componente do sistema que exija representação explícita no modelo.
- **Atributos**: As propriedades de uma entidade, ou seja, características da entidade que influem de alguma forma no sistema.
- **Fila**: Uma coleção permanente ou temporária de entidades associadas, ordenadas de forma lógica.
- **Evento**: Uma ocorrência instantânea que altera o estado do sistema.
- **Aviso de Evento**: Registro que carrega as informações necessárias para a ocorrência futura de um evento.
- **Fila de Eventos**: Uma lista de Avisos de Eventos ordenada de forma lógica contendo registros da sequência dos eventos futuros.
- **Atividade**: Intervalo de tempo de comprimento conhecido a partir do momento em que é iniciada. Sua duração pode ser definida de forma determinística, em termos de distribuição estatística, em função de atributos de variáveis e entidades ou em função do estado do sistema no instante do seu início.
- **Tempo de Espera**: Intervalo de tempo de comprimento desconhecido até o momento em que termina. Ao contrário da atividade, sua duração não é determinada, mas depende das condições apresentadas pelo sistema em seu decorrer.
- ***Clock***: Variável que representa o próprio tempo simulado.

2.3. ALGORITMO DE BRESENHAM

O Algoritmo de Bresenham foi proposto em 1965 por Jack E. Bresenham, um então funcionário da IBM. A função deste algoritmo é definir os pontos que compõem uma linha, seja esta linha reta ou curva (MORO).

Este algoritmo utiliza apenas variáveis inteiras, para minimizar o esforço computacional e assim tornar o processo mais rápido. Outra vantagem do algoritmo é permitir que o cálculo de um próximo ponto seja feito de forma incremental, usando os cálculos já feitos para o ponto anterior (MORO). A Figura 22 ilustra o comportamento do algoritmo.

Figura 22 – Algoritmo de Bresenham.



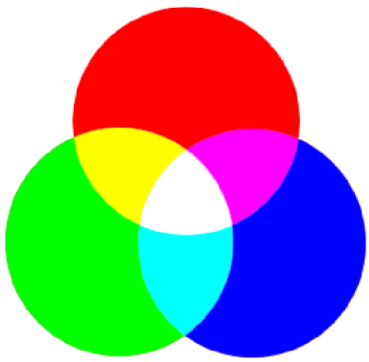
Fonte: Moro

Na Figura 22, assumindo que o pixel que acabou de ser selecionado é P, e o próximo deve ser escolhido entre o pixel à direita superior a M (pixel S) e o pixel à direita inferior a M (pixel I). Seja Q o ponto de intersecção entre a reta e a coluna (*xp + 1*) da malha, e M o ponto intermediário entre os pixels S e I, o que se faz é observar de que lado da reta está o ponto M. É fácil verificar que se M está acima de Q, o pixel I está mais próximo da reta; se M está abaixo de Q, S está mais próximo. Dessa forma, o teste do ponto-médio permite a escolha do pixel mais próximo da reta (MORO).

2.4. SISTEMA DE CORES RGB

O sistema que regula as cores dos corpos que emitem luz é conhecido como RGB (*Red, Green and Blue* em inglês, ou seja, vermelho, verde e azul), ou sistema de Cor Luz. Este sistema é utilizado em fotografia, cinema, vídeo, televisão, fotografia digital e na tela dos computadores (ROCHA). A Figura 23 representa a composição de cores do sistema.

Figura 23 - Círculos Cromáticos RGB.



Fonte: Rocha

O sistema RGB trabalha por adição de cores, ou seja, elas são somadas para produzir novas cores. As cores primárias do sistema são vermelho, verde e azul, as secundárias são ciano,

magenta e amarelo. Se somadas as três cores básicas, nas proporções corretas, obtém-se a cor branca, sendo esta uma cor terciária (ROCHA).

3. TRABALHOS RELACIONADOS

Neste capítulo serão apresentados alguns trabalhos relacionados aos temas Desvio de Obstáculos, Robótica Móvel e Plataforma de Validação. Estes documentos foram utilizados como apoio fundamental para o desenvolvimento do trabalho proposto.

3.1. NAVEGAÇÃO E DESVIO DE OBSTÁCULOS USANDO UM ROBÔ MÓVEL DOTADO DE SENSOR DE VARREDURA LASER (PEREIRA, 2006)

Em Pereira (2006) o artigo relata a utilização de um robô provido de um sensor de varredura laser para testes de navegação em ambientes semiestruturados. O objetivo do trabalho é demonstrar as vantagens da utilização de sensores *laser* na detecção e desvio de obstáculos.

O trabalho trata por "controladores" os algoritmos utilizados para as tomadas de decisão do robô. No trabalho são utilizados controladores baseados na abordagem reativa. Tendo em vista que a abordagem reativa não utiliza nenhum conhecimento prévio do meio, a não ser por informações genéricas como o ambiente ser plano e fechado, este tipo de abordagem permite a utilização destes controladores em ambientes dinâmicos.

No trabalho são utilizados três tipos de controladores. Em todos os controladores o deslocamento do robô deve ocorrer de forma suave, ou seja, sem mudanças bruscas de direção. Um dos controladores é utilizado para a navegação em corredores e o objetivo é que o robô percorra o corredor sempre estando no meio do mesmo, mantendo a mesma distância de ambas as paredes.

Os outros dois controladores são para desvio de obstáculos. O primeiro utiliza o desvio tangencial dos obstáculos sem considerar a posição final de destino. O segundo procura um caminho onde seja possível a passagem do robô e que possua a orientação mais próxima a posição final de destino

Quanto aos testes eles são executados em diversos tipos de ambiente: corredor em que a distância entre as paredes não varia; corredor com afunilamento, ondem em determinado momento a distância entre as paredes corredor diminui; corredor com curvas; e em ambiente aberto. Nestes ambientes foram realizados testes com e sem obstáculos.

Nos testes realizados em corredores retos, o robô é guiado pelo controlador de navegação em corretor e, caso detectado um obstáculo, o controlador de desvio não tangencial desvia do obstáculo. Nos testes realizados em corredores com curva o robô é guiado pelo controlador de desvio tangencial. No ambiente aberto foram testados os dois controladores de

desvio de obstáculo. Durante a execução dos testes o robô executou todos os percursos da forma desejada.

3.2. DESENVOLVIMENTO DE SISTEMA DE NAVEGAÇÃO POR GNSS (GONÇALVES, 2011)

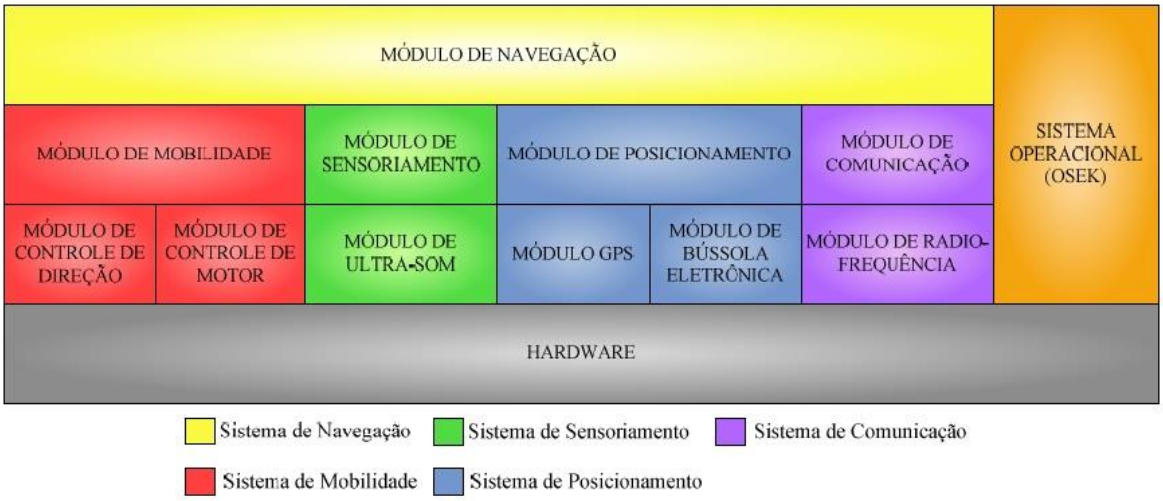
Em Gonçalves (2011) o autor adapta um veículo para receber a infraestrutura e as tecnologias necessárias para proporcionar: mobilidade; detecção de posicionamento; comunicação sem fio; e sensoriamento do ambiente. Além disso é desenvolvido um sistema remoto para monitorar e instruir o veículo à distância e em tempo real. O objetivo é desenvolver um veículo que seja capaz de navegar, de forma autônoma, até um ponto determinado fazendo uso de GPS.

O projeto é realizado em um veículo em escala reduzida, ou seja, um veículo de brinquedo que no modelo original é rádio controlado. Todo o *hardware* foi embarcado neste veículo como intuito de torna-lo autônomo.

O ambiente de desenvolvimento do *software* do veículo foi elaborado sobre a plataforma de hardware DEMO9S12XEP100, produzida pela empresa *Freescale*. Como plataforma de *software* foi escolhido o sistema operacional de tempo real OSEK (*Offene Systeme und deren Schnittstellen für die Elektronik in Kraftfahrzeugen*). Em tradução livre, o nome significa “Sistemas abertos e suas interfaces para eletrônica em veículos motorizados”.

O algoritmo de *software* do veículo foi desenvolvido de forma modular com finalidades específicas em cada módulo. A Figura 24 ilustra a arquitetura de *software* elaborada para a aplicação do veículo.

Figura 24 - Arquitetura de software do sistema do veículo.



Fonte: Gonçalves (2011)

Após o termino da implementação do trabalho o veículo passou a contar com dois modos de operação, um Remoto e outro Autônomo, além de um modo Inicial que apenas é utilizado para a inicialização do sistema.

O Modo Remoto permite o controle do veículo através do *software* de monitoramento e controle desenvolvido. O Modo Autônomo aguarda uma missão, que é um ponto de destino, e executa automaticamente todos os comandos que o veículo precisa executar para concluir a missão. Todos os três modos permitem o monitoramento do veículo.

Os testes neste trabalho são divididos em três partes. As duas primeiras partes são executadas com o veículo em Modo Autônomo, já a terceira parte é executada em Modo Remoto.

Na primeira bateria de testes tem o objetivo de verificar a capacidade do veículo de atingir um ponto previamente determinado, bem como verificar a acurácia com a qual se chegou ao ponto. Em todos os 28 testes realizados, o veículo conseguiu atingir o ponto de destino.

Na segunda etapa dos testes o intuito é avaliar a capacidade do veículo em cumprir uma missão, a qual se trata de alcançar subsequentemente 5 pontos determinados. Nestes testes também é avaliada a trajetória e capacidade de manobra do veículo. Em todos os 12 testes realizados o veículo conseguiu atingir os 5 pontos da missão.

A terceira parte dos testes, executado em Modo Remoto, visa traçar um caminho conhecido com o veículo, para posterior comparação com o caminho representado após o processamento dos dados de monitoramento. Nestes testes foram comparadas as direções que

o veículo estava, em vários pontos do percurso, e também o traçado do percurso completo. As comparações entre as direções foram bastante similares, e o traçado executado também.

3.3. USO DE REALIDADE VIRTUAL NO DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE DO ESTACIONAMENTO DE VEÍCULOS (HEINEN; OSÓRIO; HEINEN; KELBER, S/D)

No artigo de Heinen, Osório, Heine e Kelber (s/d) os autores utilizam o sistema SEVA3D, que implementa um ambiente virtual, para controlar um veículo. O objetivo é demonstrar o uso de Realidade Virtual no desenvolvimento de sistemas de controle inteligente de veículos, através do simulador SEVA3D.

Com o sistema SEVA3D é feito o controle do carro através da leitura de um conjunto de sensores do tipo sonar, gerando os comandos de aceleração e de giro de direção, de modo a estacionar o carro em uma vaga paralela.

Os sonares são simulados através da definição de uma seção cônica no espaço virtual (simulando o cone no qual as ondas sonoras se propagam em um sonar real), onde os objetos que estiverem dentro do volume formado pelo cone são detectados. Diversas linhas de detecção de objetos (raios) são geradas a partir da posição do sensor e direcionados de acordo com a orientação espacial do sonar, permanecendo dentro do volume formado pelo cone. Se algum dos raios colidir com algum dos objetos presentes no ambiente, a distância do sensor até o ponto de colisão é informada.

O controle do veículo é feito por meio de um autômato finito, ou seja, uma máquina de estados. Para a tarefa de estacionar o veículo foi necessário um autômato com nove estados: Parado; Procurando Vaga; Posicionando; Afastando; Entrando Vaga; Abortando; Posicionando Vaga; Otimizando Vaga; e Alinhando. Além da tarefa de estacionar o veículo também foi desenvolvida a tarefa de retirada do veículo, para esta foram utilizados quatro estados: Parado; Preparando Retirada; Retirando; e Retornando.

Durante os testes da manobra de estacionar foi obtida uma taxa de sucesso de 96,83% com um estacionamento considerado bastante satisfatório. A distâncias do carro estacionado em relação ao meio-fio ficando meio-fio tiveram uma média de 26,16cm e um desvio padrão de 5,92cm. Durante os testes da manobra de retirada obteve-se uma taxa de sucesso de 100%.

3.4. MÉTODO DE DESVIO DE OBSTÁCULOS APLICADO EM VEÍCULO AUTÔNOMO (CHIN MIN WEI, 2015)

Em Chin Min Wei (2015) o artigo destaca técnicas de desvio de obstáculos e sensores para detecção de obstáculo. O objetivo do trabalho é verificar a viabilidade e eficiência de um método de desvio de obstáculos baseado em sensores, utilizando a plataforma Robodeck.

O trabalho aborda as seguintes técnicas de desvio de obstáculo: Algoritmo do Inseto (*Bug Algorithm*) 1, 2 e Tangencial; e Algoritmo de campo de força virtual (*Virtual Force Field*). Quanto os sensores o trabalho estuda os seguintes tipos: ultrassônicos; lidar (*Light Detection and Ranging*); e radar (*Radio Detection and Ranging*). Tanto nas técnicas de desvio quanto nos sensores, é feita a explicação do seu funcionamento e são destacadas suas vantagens e desvantagens.

Para a execução do trabalho o autor optou pela utilização do Algoritmo do Inseto 2 e o sensor ultrassônico, sensor este que vem integrado à plataforma robótica utilizada. O programa do algoritmo foi escrito em linguagem Java e com o formato de uma máquina de estados, com funções definidas para cada situação.

Foram feitos três 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.

Vale destacar que em nenhum dos testes a posição final do Robodeck foi exatamente em cima do ponto de destino. Quanto mais manobras o robô precisou fazer durante o percurso, maior foi a diferença entre o final da movimentação e o ponto de destino esperado.

3.5. COMPARAÇÃO DOS TRABALHOS RELACIONADOS

Neste capitulo é disposta uma tabela comparativa entre os trabalhos relacionados. A Tabela 1 contém os dados comparativos.

Tabela 1 - Quadro comparativo dos trabalhos relacionados estudados.

Autor	Objetivo	Tipo de Sensor de Detecção de Obstáculo	Técnica para Evitar Colisão	Resultados
Pereira (2006)	Demonstrar as vantagens da utilização de sensores laser na detecção e desvio de obstáculos.	Laser	Desvio tangencial de obstáculo.	Os testes foram realizados em corredor e áreas abertas, com e sem obstáculos. Durante a execução dos testes o robô executou todos os percursos da forma desejada.
Gonçalves (2011)	Desenvolver um veículo que seja capaz de navegar, de forma autônoma, até um ponto determinado fazendo uso de GPS.	Ultrassônico	Parada do veículo ao detectar um obstáculo.	Os testes foram divididos em três partes: verificar a capacidade do veículo de atingir um ponto previamente determinado; avaliar a capacidade do veículo em cumprir uma missão; e traçar um caminho conhecido com o veículo. Em todos testes o veículo executou a tarefa corretamente.
Heinen; Osório; Heinen; Kelber (S/D)	Demonstrar o uso de Realidade Virtual no desenvolvimento de sistemas de controle inteligente de veículos, através do simulador SEVA3D.	Ultrassônico	Durante as manobras pré-definidas, o ultrassom verifica a distância de obstáculos, para evitar colisões.	Durante os testes da manobra de estacionar foi obtida uma taxa de sucesso de 96,83% com um estacionamento considerado bastante satisfatório. A manobra de retirada obteve-se uma taxa de sucesso de 100%.
Chin Min Wei (2015)	Verificar a viabilidade e eficiência de um método de desvio de obstáculos baseado em sensores.	Ultrassônico	Algoritmo de desvio de obstáculo Inseto 2.	Foram feitos três tipos de teste: um obstáculo simples, um obstáculo convexo e um obstáculo intransponível. Em todos eles o algoritmo atingiu o resultado esperado.
Devitte (2017)	Desenvolver uma plataforma para auxiliar na definição da melhor estratégia de desvio de obstáculos para um determinado ambiente.	Laser e Ultrassônico	Algoritmo Inseto Tangencial e Algoritmo Campos Potenciais Artificiais.	A definir

Fonte: Autor.

Entre os trabalhos relacionados, nenhum deles apresenta a mesma finalidade que a proposta descrita neste documento. O trabalho que será desenvolvido é uma plataforma de validação de algoritmos aplicados à veículos autônomos com ênfase na detecção e desvio de objetos. Para tanto, foram estudados trabalhos relacionados a veículos autônomos e técnicas de desvio de obstáculos.

Foram selecionados trabalhos que somados apresentam a fundamentação do que se pretende desenvolver. Em Gonçalves (2011) pode-se verificar o modelo de estrutura de veículo autônomo que será utilizado. Os trabalhos de Pereira (2006) e Chin Min Wei (2015) apresentam técnicas de desvio, utilizando diferentes tipos de sensores, que também serão aplicados no desenvolvimento do trabalho. O trabalho de Heinen, Osório, Heinen e Kelber (S/D) utiliza um simulador para o desenvolvimento de uma funcionalidade para veículo autônomo, demostrando o auxílio que a plataforma de validação pode trazer ao desenvolvedor.

4. TRABALHO DESENVOLVIDO

Este capítulo descreve a plataforma de validação que foi implementada, todas as escolhas tomadas são baseadas na pesquisa já realizada ou em pesquisas pontuais para algum aspecto em específico.

4.1. DEFINIÇÃO

A plataforma desenvolvida possui como objetivo a validação de algoritmos de desvio de obstáculos aplicáveis a veículos autônomos. O intuito é testar os algoritmos em diferentes ambientes e com variados tipos e combinações de sensores. Os resultados obtidos auxiliarão a definir o algoritmo mais indicado para um determinado ambiente, bem como as características que os sensores de navegação utilizados no projeto devem possuir para que haja um bom desempenho.

4.2. ROVER SIM

O nome atribuído a plataforma é Rover Sim, uma abreviação de Rover Simulator. Este nome foi escolhido com base no agente simulado pela plataforma, sendo que a definição de "Rover" é atribuída a veículos de exploração espacial. Como o agente simulado não tem o conhecimento prévio do ambiente que irá atuar, ele deve explorar o ambiente e tomar decisões a partir das informações disponíveis.

A Rover Sim foi desenvolvida em linguagem Java com a utilização da IDE Eclipse, devido a maior afinidade com estas ferramentas.

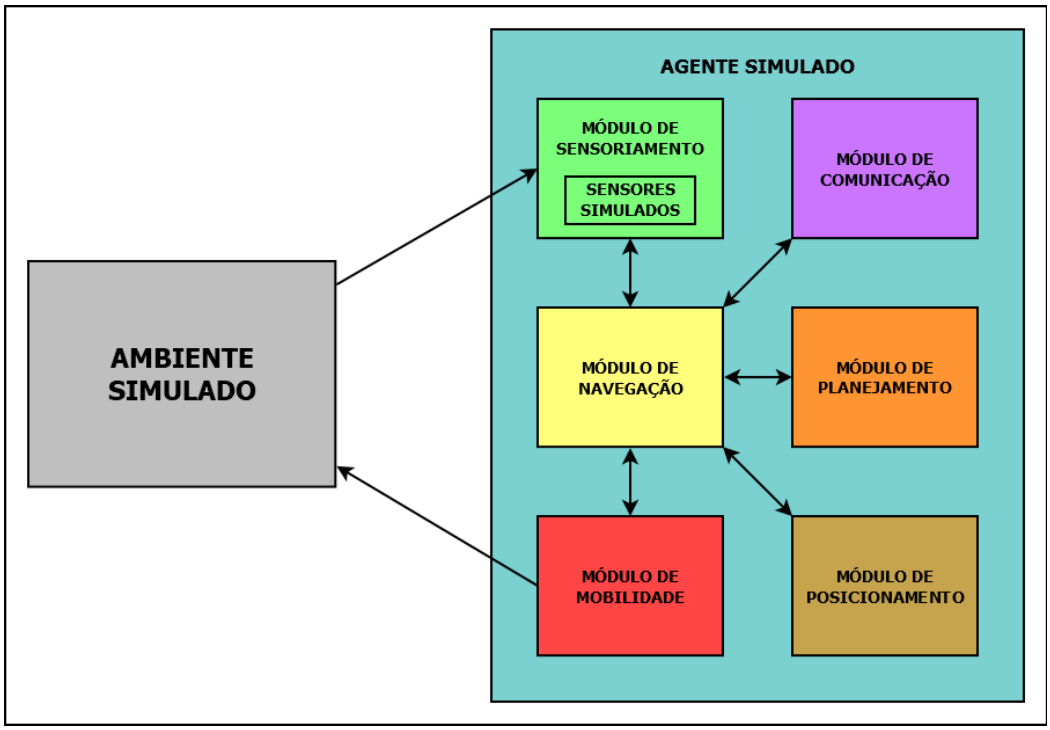
O principal componente da Rover Sim é o agente simulado, sendo este o responsável por demostrar para o usuário o comportamento do algoritmo que se pretende validar. Através do agente são coletadas todas as informações que posteriormente servirão como base para a tomada de decisão do usuário.

Trabalhando como parte integrada do agente estão os sensores simulados. Estes por sua vez possuem grande importância para os resultados que serão gerados pela Rover Sim, tendo em vista que é através da leitura que eles farão do ambiente que o algoritmo irá tomar suas decisões.

Por último, englobando os componentes citados anteriormente, está o ambiente simulado. Neste ambiente será realizada a movimentação do agente e a verificação de obstáculos através dos sensores.

A interação dos componentes da Rover Sim pode ser vista na Figura 25.

Figura 25 - Arquitetura da Rover Sim.



Fonte: Autor.

4.2.1. AGENTE SIMULADO

O desenvolvimento do agente foi realizado de maneira modular, conforme visto na Figura 25, onde cada módulo possui funções específicas que se somam para o pleno funcionamento do agente. Os módulos que o compõem são de comunicação, posicionamento, sensoriamento, planejamento, mobilidade e navegação.

Estes módulos foram definidos com base no projeto de veículos autônomos terrestres descrito no capítulo 2.1. Tendo em vista o objetivo da plataforma, alguns destes elementos foram simplificados, conforme será explicado a seguir:

- O módulo de navegação é responsável pelas tomadas de decisões do agente, a partir dos dados fornecidos pelos demais módulos.
- O módulo de comunicação transmite dados entre o agente e sua estação de controle. É através deste módulo o agente recebe os novos destinos que deve alcançar e também pode receber a definição de que o destino atual não é alcançável.
- O módulo de posicionamento informa onde o agente se encontra dentro do ambiente. Foi considerado que o agente sempre sabe sua posição exata no ambiente.

- O módulo de sensoriamento é responsável por sensoriar o ambiente próximo ao agente. As leituras do ambiente, realizadas pelos sensores, são repassadas ao agente através deste módulo.
- O módulo de planejamento, por meio dos planejadores (item. 2.1.2, cap. 2), gera o trajeto que deve ser executado pelo agente. O planejador *offline* apenas irá definir um trajeto reto entre a posição atual do agente e o ponto de destino, enquanto o planejador *online*, através do controlador definido para o agente, irá definir o restante das ações durante o deslocamento.
- O módulo de mobilidade executa o deslocamento do agente dentro do ambiente.

A movimentação do agente no ambiente é realizada através de uma sequência de tarefas: o agente verifica, em seu módulo de comunicação, se um novo ponto de destino foi definido; verifica sua posição atual no ambiente, através do módulo de localização; analisa os obstáculos próximos a si, pelo módulo de sensoriamento; no módulo de planejamento, define o trajeto a ser realizado; e, através do módulo de mobilidade, realiza sua locomoção no ambiente. O módulo de navegação faz a integração de todos os módulos.

Para a representação do agente foi utilizado um círculo, sendo este tratado como uma partícula omnidirecional, ou seja, pode se mover para qualquer direção independente de seu posicionamento atual. Os atributos considerados na simulação do agente são: tamanho; velocidade máxima; distância segura; e controlador.

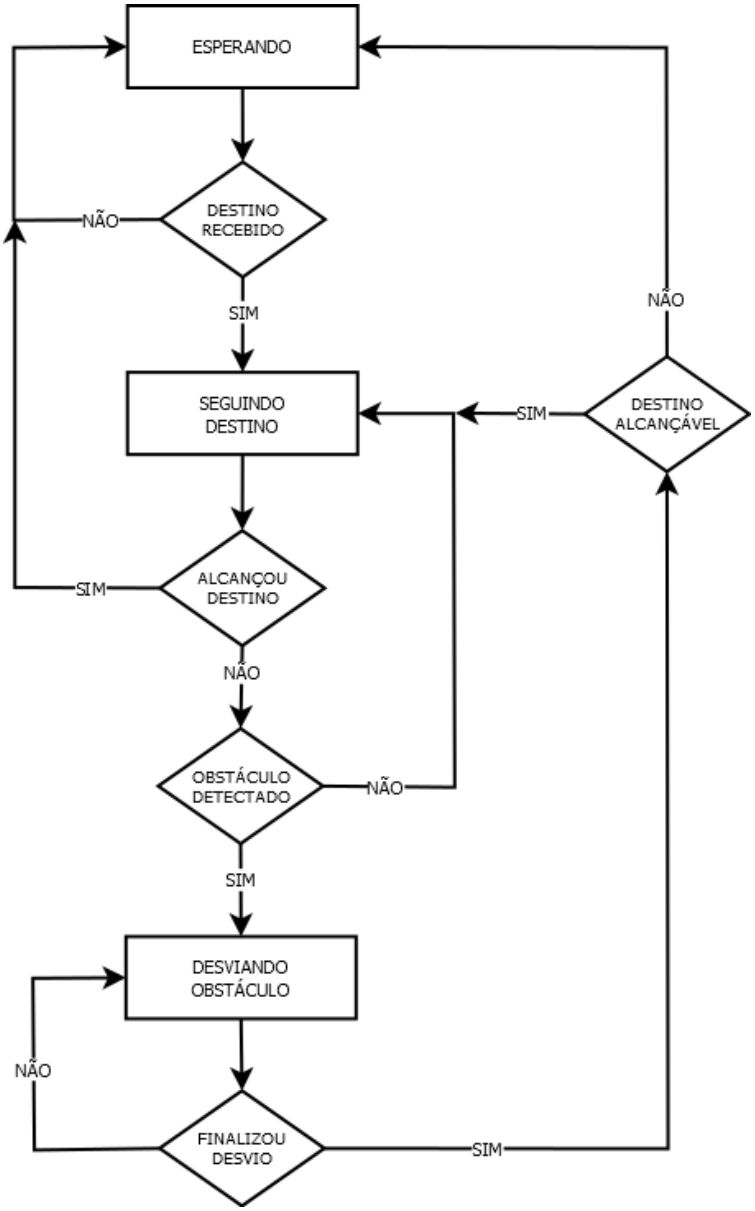
- Tamanho (cm): Pelo fato de o agente ser tratado como um círculo, apenas uma dimensão define seu tamanho. Esta medida é utilizada como diâmetro do círculo que lhe representa, sendo este considerado para verificação de colisão, cálculo do alcance dos sensores, e cálculo da distância segura;
- Velocidade máxima (m/s): Maior velocidade que o agente é capaz de alcançar. Na Rover Sim a velocidade é considerada constante e os efeitos da aceleração e inércia foram ignorados;
- Distância segura (cm): É utilizada para expressar a distância que o agente pode se aproximar dos obstáculos, sem que ele corra risco de colidir. Esta informação é considerada pelo controlador para evitar colisões;
- Controlador: Refere-se ao algoritmo de desvio de obstáculo que será utilizado pelo agente durante a simulação. Estão disponíveis para utilização o algoritmo Inseto

Tangencial e o algoritmo de Campos Potenciais Artificiais (itens 2.1.2.1 e 2.1.2.2, cap. 2).

4.2.1.1. DIAGRAMA DE ESTADOS

O controle do agente se dará através de uma máquina de estados finitos, onde cada estado apresenta um comportamento distinto. A Figura 26 demostra o diagrama de estados.

Figura 26 - Diagrama de estados do agente



Fonte: Autor.

O agente inicia a simulação no estado “Esperando”, ele permanece neste estado até receber um destino que deva alcançar. Ao receber um novo destino seu estado é alterado para “Seguindo Destino”. No estado “Seguindo Destino” o agente se desloca, em linha reta, em

direção ao ponto de destino. Durante seu deslocamento o agente verifica se o destino foi alcançado ou se algum obstáculo foi detectado.

Ao alcançar o ponto de destino, o agente retorna ao estado “Esperando”. Caso um obstáculo seja detectado, antes do agente alcançar seu destino, o estado do agente é alterado para “Desviando Obstáculo”. O tipo de movimentação que o agente irá realizar no estado “Desviando Obstáculo”, depende do algoritmo de desvio que estará sendo utilizado.

Ao finalizar o movimento de desvio haverá duas possibilidades, o destino é alcançável ou inalcançável. Caso o destino seja alcançável, o estado do agente é alterado para “Seguindo Destino” e ele volta a se deslocar em direção ao ponto de destino. Caso o destino seja inalcançável, o agente retorna ao estado “Esperando”.

4.2.2. SENSORES SIMULADOS

A função dos sensores é realizara leitura do ambiente e retornar a distância dos obstáculos em relação ao agente. Os sensores simulados consideram as seguintes variáveis para seu funcionamento: tipo; alcance; distância mínima; ângulo de detecção; grau de precisão; erro médio; e direção. As variáveis do sensor são identificadas a seguir:

- Tipo: Existem dois tipos de sensores disponíveis na Rover Sim, sensor ultrassônico e sensor laser (itens 2.1.3.1 e 2.1.3.2, cap. 2);
- Alcance (cm): O alcance do sensor se refere a distância máxima que o sensor tem capacidade de detectar um obstáculo;
- Distância mínima (cm): Alguns sensores possuem distância mínima de detecção. Se um sensor possuir uma distância mínima de 5 cm, mesmo que um obstáculo esteja a 3 cm do sensor ele irá reportar que o obstáculo está a 5 cm.
- Ângulo de detecção (grau de ângulo): O ângulo de detecção representa a área de detecção do sensor. Um sensor com ângulo de detecção de 90º localiza apenas obstáculos que estejam posicionados a sua frente, enquanto um sensor com 360º de ângulo de detecção é capaz de localizar obstáculos em qualquer posição a sua volta.
- Grau de precisão (grau de ângulo): A precisão do sensor está vinculada a seu grau de precisão, quanto menor o grau de precisão do sensor, mais preciso é o sensor. Um sensor com ângulo de detecção de 90º e com grau de precisão de 2º apresenta uma leitura de 45 posições, enquanto o mesmo sensor com 1º de precisão apresenta 90 posições de leitura.

- Erro médio (cm): O erro médio do sensor representa a imprecisão de sua detecção. Se um obstáculo que estiver a 15 cm de distância for detectado por um sensor que possua erro médio de 1 cm, ele pode ser reportado como estando a uma distância entre 14 e 16 cm, um exemplo de leitura seria 14,56 cm de distância.
- Direção (grau de ângulo): A direção do sensor representa o seu posicionamento no agente e deve ser informada em graus. Zero graus significa que o sensor fica posicionado para a frente do agente, 180º significa que ele fica posicionado para a traseira do agente. Podem também ser utilizados graus negativos, 270º ou -90º definem o sensor em uma mesma posição.

A leitura dos sensores é realizada através da verificação dos pixels da imagem do ambiente. Para esta verificação são executadas as seguintes etapas.

Primeiramente são traçadas linhas com origem na posição atual do agente e terminando no alcance do sensor, estas linhas preenchem a área do ângulo de abertura do sensor. A quantidade de linhas traçadas depende do ângulo de detecção do sensor e de seu grau de precisão.

Após definir as linhas é feita a verificação dos pixels que as compõe. A definição dos pixels é realizada através do algoritmo de Bresenham (item 2.4, cap. 2). Na verificação do pixel são analisadas suas 3 cores base, vermelho, verde e azul (item 2.5, cap. 2). O pixel que for de cor escura, abaixo da combinação [R 51, G 51, B 151], é considerado um obstáculo.

A verificação de cada linha é iniciada no ponto de origem e segue para a outra extremidade. Ao localizar um pixel que possua obstáculo é calculada a distância do pixel até o obstáculo, este valor é armazenado e é iniciada a leitura da próxima linha. Este procedimento se repete até que todas as linhas tenham sido verificadas.

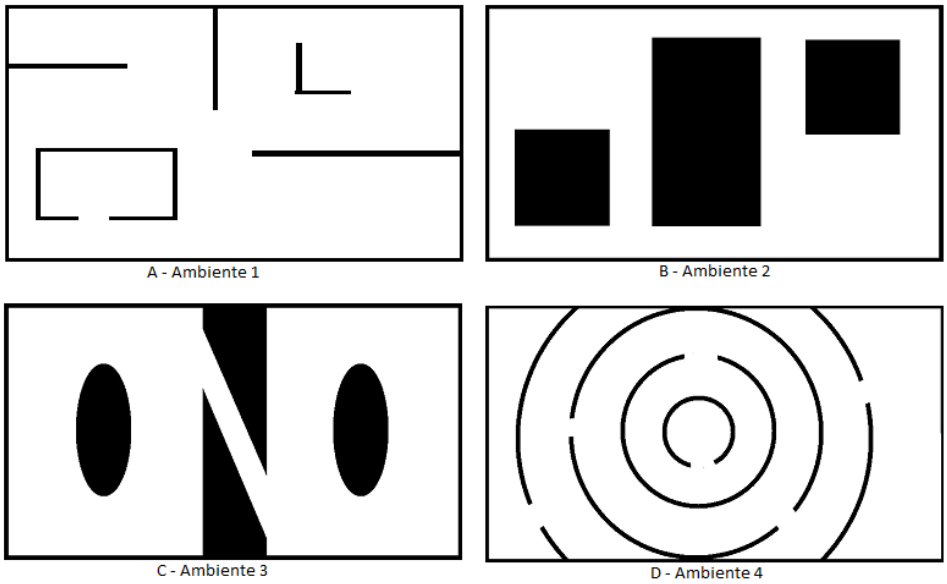
A diferença dos dois tipos de sensores, em relação a Rover Sim, está no retorno de suas leituras. O sensor laser retorna uma lista contendo a leitura exata de cada grau verificado, enquanto o sensor ultrassônico retorna em todas as posições da lista a leitura do obstáculo mais próximo.

O agente pode possuir vários sensores com várias configurações diferentes, onde todos trabalham em conjunto para detectar os obstáculos no em torno.

4.2.3. AMBIENTE SIMULADO

O ambiente de navegação utiliza uma representação com uma visão superior do ambiente, similar a planta baixa de uma construção. O ambiente é gerado a partir de uma imagem fornecida pelo usuário. A Figura 27 demonstra alguns ambientes de navegação.

Figura 27 - Ambientes de navegação.



Fonte: Autor.

Além da imagem do ambiente de navegação é necessário definir o tamanho que o ambiente representa. Para isso deve ser definido quantos centímetros um pixel da imagem representa. Para um mapa de ambiente de 800x450 pixels, se for definido que um pixel corresponde a 10 centímetros, o ambiente de validação terá 80x45 metros.

4.2.4. GERAÇÃO DE RESULTADOS

Para disponibilizar os resultados produzidos a Rover Sim gera um arquivo no formato CSV (*Comma-separated values*). A cada ciclo de iteração do agente, informações atualizadas são adicionadas ao arquivo, mantendo um histórico de tudo que ocorreu durante a simulação. O *layout* do arquivo de resultados pode ser visualizado no Anexo A. As informações armazenadas no arquivo são:

- Iteração: Número da repetição do ciclo de simulação;
- Tempo Iteração (s): Tempo de processamento dos cálculos da iteração atual;
- Tempo Total (s): Tempo de processamento total da simulação;
- X Agente: Posição atual do agente no eixo cartesiano X;

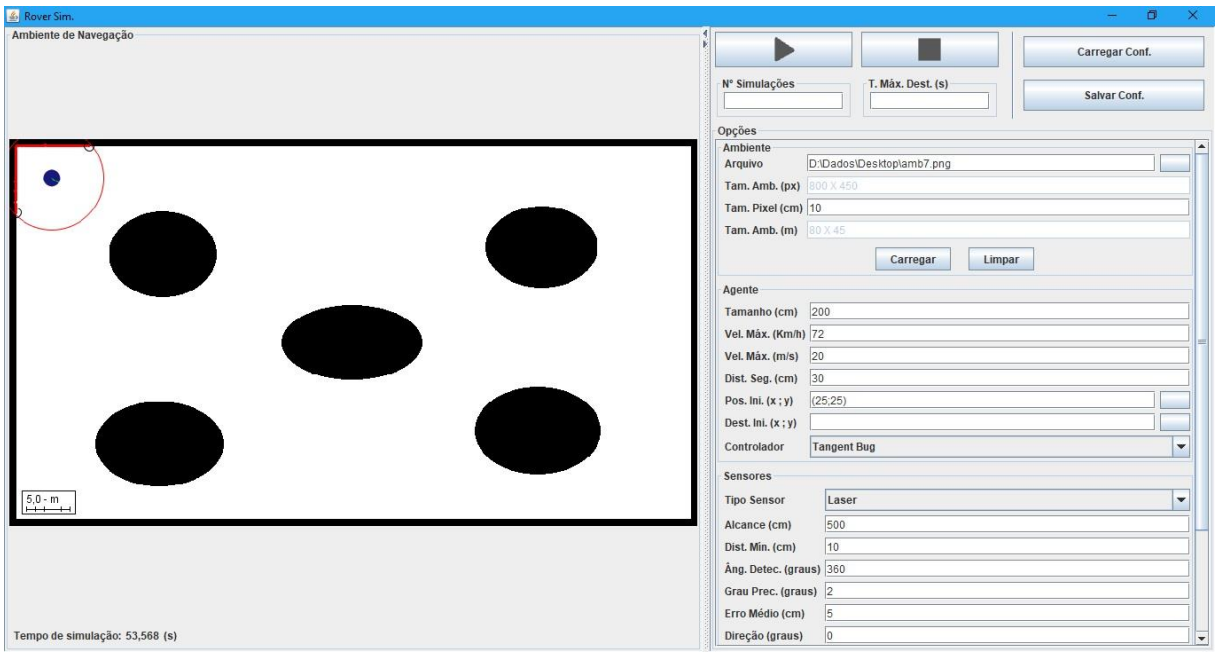
- Y Agente: Posição atual do agente no eixo cartesiano X;
- Distância Destino (m): Distância atual até o destino em linha reta;
- X Destino: Posição no eixo cartesiano X do destino atual;
- Y Destino: Posição no eixo cartesiano Y do destino atual;
- Direção Agente (grau de ângulo): Direção atual do deslocamento do agente;
- Velocidade Atual (m/s): Velocidade atual do agente;
- Velocidade Máxima (m/s): Velocidade máxima do agente;
- Deslocamento (m): Deslocamento do agente na iteração atual;
- Deslocamento Total (m): Deslocamento total do agente durante a simulação;
- Distância O.M.P. (m): Distância atual do obstáculo mais próximo ao agente;
- X Obst. M.P: Posição no eixo cartesiano X do obstáculo mais próximo ao agente;
- Y Obst. M.P: Posição no eixo cartesiano Y do obstáculo mais próximo ao agente;
- Colisão: Informa se o veículo está colidindo na iteração atual (1-Sim, 0-Não);
- Destino Alcançado: Informa se o destino atual foi alcançado (1-Sim, 0-Não);

Os resultados gerados a cada simulação realizada na Rover Sim, contém dados que podem ser confrontados para verificar as configurações que possuem melhor desempenho em um determinado ambiente. Para isso pode ser considerado o menor tempo para alcançar o destino, a menor distância percorrida pelo agente, se houveram ou não colisões durante a validação, o ambiente que o algoritmo melhor atua, entre outros critérios de avaliação.

4.3. UTILIZAÇÃO DA ROVER SIM.

A utilização da Rover Sim é feita através de sua interface gráfica. Inicialmente a interface foi implementada conforme a proposta do Trabalho de Conclusão 1, contudo, durante a utilização do sistema pode-se perceber que o modelo inicial da interface não proporcionava uma boa experiência de uso. Para melhorar e facilitar o uso da plataforma, a interface foi alterada e o resultado final pode ser visto na Figura 28.

Figura 28 - Interface da Rover Sim



Fonte: Autor.

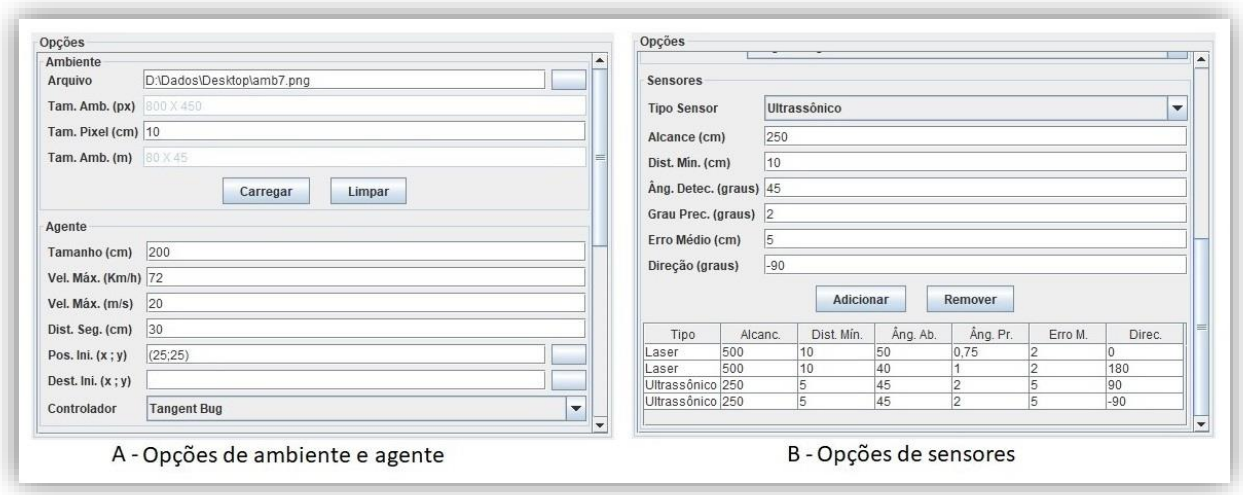
No lado esquerdo da interface fica o ambiente de navegação, onde é visualizado o mapa do ambiente e é acompanhada a movimentação do agente, sendo esta exibida em 2D, com visão superior da cena. No lado direito da interface fica o painel de controle, onde são feitos os ajustes nas configurações antes de inicializar a validação.

Para iniciar a utilização do sistema, o primeiro passo é ajustar suas configurações através do painel de opções.

4.3.1. CONTROLES E CONFIGURAÇÕES

Todas as configurações aplicáveis à plataforma são exibidas no painel de opções, este painel foi subdividido em painéis específicos referentes a ambiente, agente e sensores. Conforme mostra a Figura 29.

Figura 29 – Painel de Opções.



Fonte: Autor.

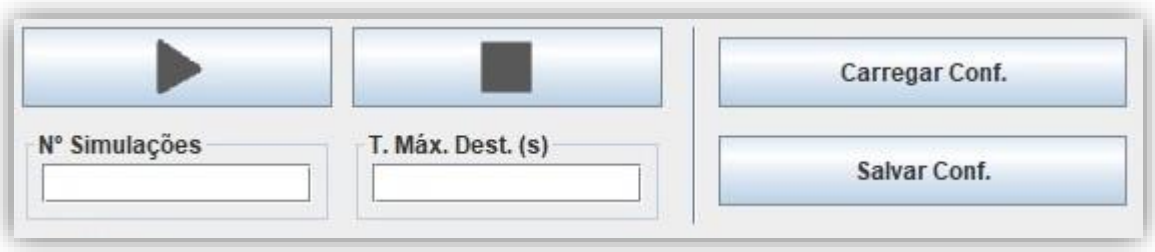
Nas opções de ambiente é definida, através da opção “Arquivo”, o cenário que será utilizado. Após selecionado o arquivo, o botão “Carregar” inicializa o ambiente com o cenário escolhido. Também nestas opções é definido qual o tamanho que um pixel da imagem original do ambiente corresponde em centímetros.

A segunda subdivisão das opções apresenta as configurações do agente. Para este deve ser configurado o tamanho, velocidade máxima, distância segura, posição inicial e o controlador. Nas configurações do agente ainda pode ser definido o destino inicial, caso esta opção seja utilizada, ao iniciar a simulação, o agente automaticamente começará seu deslocamento até o ponto indicado. Nesta opção pode ser definido mais de um ponto de destino inicial, neste caso, o agente irá realizar seu deslocamento tentando alcançar todos os pontos de destino na ordem em que eles foram informados.

A última subdivisão das opções possui as configurações dos sensores. Para cada sensor deve ser definido seu tipo, alcance, distância mínima, ângulo de detecção, grau de precisão, erro médio e direção. Os sensores configurados para o agente são exibidos em uma tabela localizada na parte inferior do painel de opções.

Após terem sido feitas todas as configurações de ambiente, agente e sensores, pode ser iniciada a validação do algoritmo de desvio de obstáculos. Para o controle da simulação é utilizado o Painel de Controle, a Figura 30 apresenta este painel.

Figura 30 – Painel de Controle.



Fonte: Autor.

O botão para iniciar a simulação fica no canto superior esquerdo, com o ícone do "*play*", o botão ao lado, com o ícone do "*stop*", finaliza a simulação. Abaixo do botão iniciar fica o campo "Nº Simulações", caso seja informado o número de simulações, a plataforma irá executar a mesma simulação quantas vezes foi informado. A simulação é reiniciada quando o último ponto de destino inicial for alcançado ou considerado inalcançável.

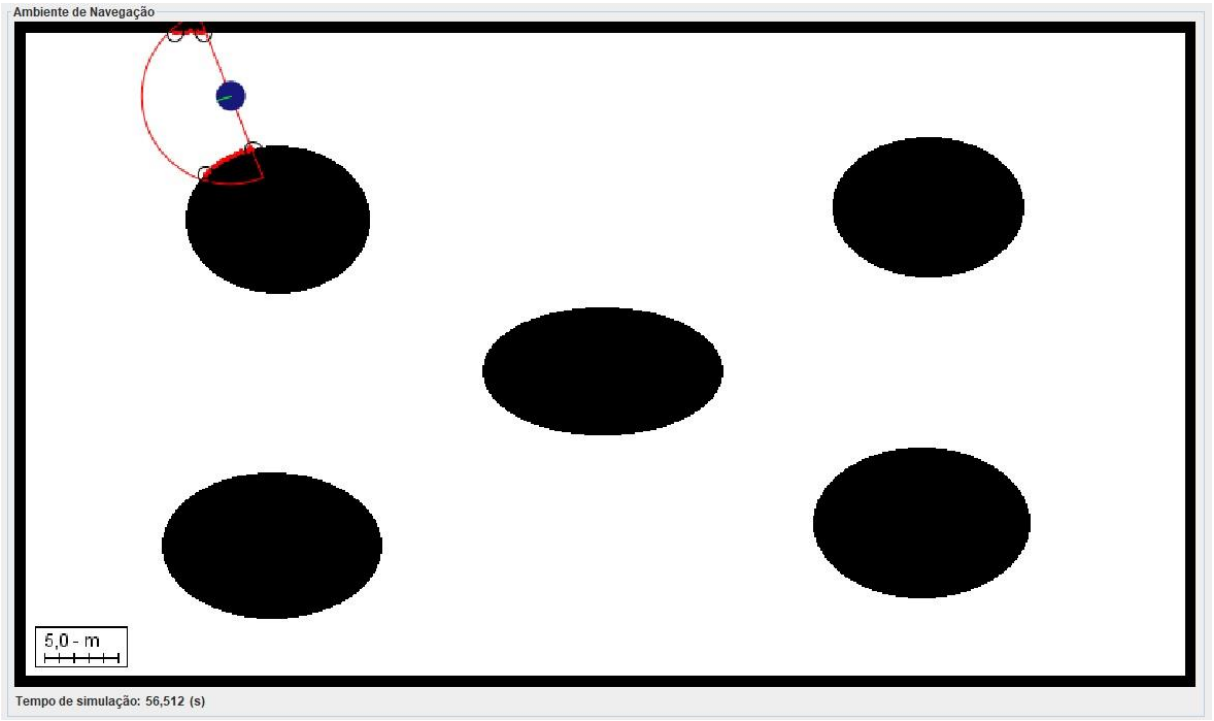
No campo "T. Máx. Dest. (s)" pode ser informado o tempo máximo que o agente terá para alcançar cada destino. Caso este campo seja deixado em branco, o controlador utilizado será responsável por definir o destino como inalcançável.

Os botões no lado direito do painel são utilizados para carregar e salvar as configurações atuais da plataforma, deste modo não é necessário que todas as configurações da plataforma sejam ajustadas a cada utilização.

4.3.2. AMBIENTE DE NAVEGAÇÃO

O ambiente de navegação é onde o resultado do processamento da Rover Sim é exibido. A Figura 31 apresenta uma cena do ambiente de navegação. O ambiente inicial da cena são as cinco elipses de cor preta, mais o retângulo preto que contorna as extremidades do ambiente.

Figura 31 – Ambiente de navegação.



Fonte: Autor.

No canto superior esquerdo da cena é possível visualizar o agente atuando no ambiente, ele é exibido como círculo azul. Pode-se verificar no agente a existência de um traço verde que se estende do centro até uma de suas extremidades, este por sua vez representa a direção que o agente está se deslocando.

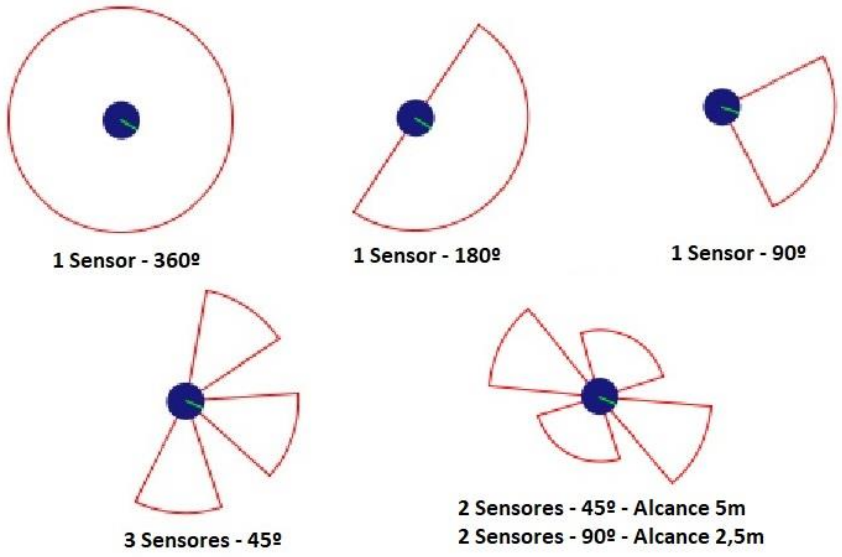
Próximo ao agente é possível visualizar o contorno de um semicírculo vermelho, este representa o alcance do sensor que o agente está utilizando. Nesta cena está senso utilizado um sensor com 180º de detecção. Caso seja utilizado um sensor com um ângulo de detecção menor, o semicírculo se ajusta e fica com formato de cone, se for utilizado um sensor com 360º de detecção é exibido um círculo completo em torno do agente. A Figura 32 irá demonstra a representação de alguns sensores.

Na cena também podem ser vistos os obstáculos detectados pelo sensor do agente, eles são destacados em vermelho e podem ser localizados na intersecção entre o alcance do sensor e os obstáculos do ambiente. Nesta cena um obstáculo foi detectado acima do agente, junto a extremidade superior do ambiente, e outro abaixo do agente, junto a elipse próxima a ele. Além do destaque em vermelho do obstáculo, o ponto inicial e final do obstáculo é contornado com um círculo de cor preta.

O canto inferior esquerdo do ambiente de navegação exibe uma escala do ambiente em relação a metros, a escala é definida sobre o comprimento de 50 pixels da imagem original do ambiente. Antes de ser iniciada esta simulação foi definido que cada pixel do ambiente original corresponde a 10 cm, deste modo, 50 pixels da imagem original correspondem a 5 metros.

Durante a simulação, diversos cálculos são realizados, sendo que estes cálculos levam um determinado tempo para serem executados. Por este motivo o tempo do ambiente simulado demora mais para passar que o tempo do relógio. Esta informação também é apresentada no ambiente de navegação, abaixo da imagem do ambiente, no canto inferior esquerdo, com o nome de “Tempo de simulação”. Este é o tempo decorrido no ambiente simulado.

Figura 32 – Representação de sensores.



Fonte: Autor.

Quanto a exibição do ambiente de navegação deve-se destacar que a cena é ajustada conforme o tamanho da janela da plataforma, sempre mantendo as proporções da cena original. Se a janela for maior que o ambiente original a cena exibida é “aumentada”, deixando a visualização mais próxima, se a janela for menor que o ambiente original, a cena é “diminuída”, deixando a visualização mais afastada.

Através do ambiente de navegação também são definidos os destinos que o agente deve alcançar, para isso basta clicar com o *mouse* sobre a posição do ambiente que se deseja definir como destino.

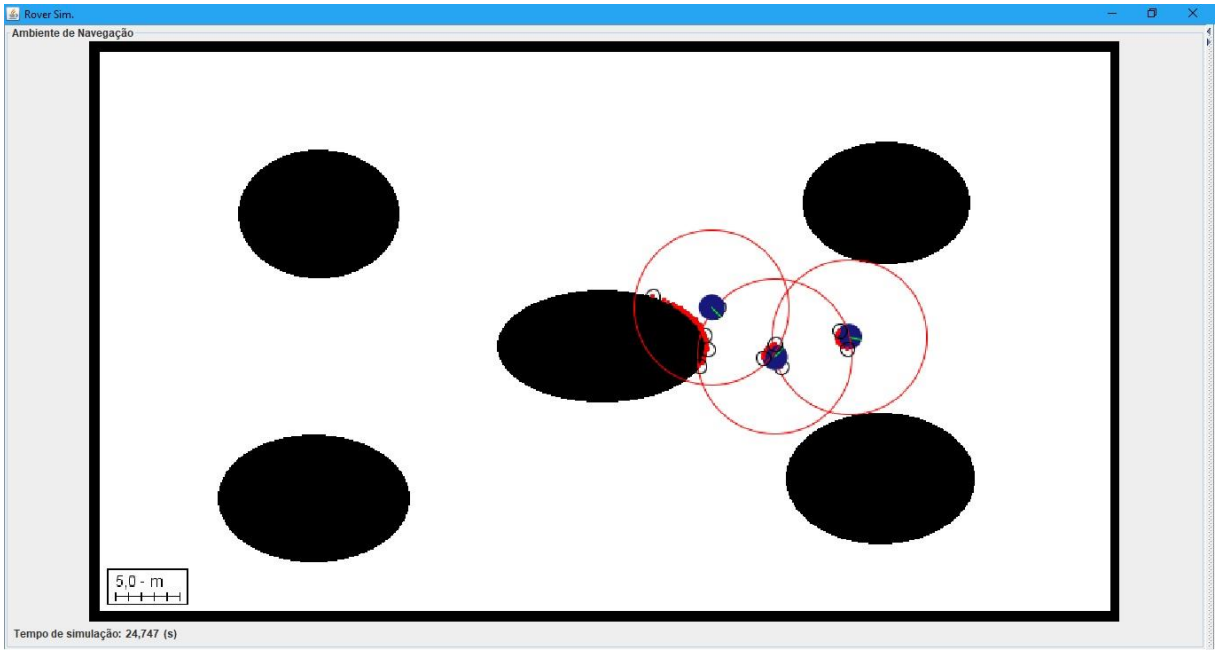
Existem duas opções ao se adicionar um novo destino, clicar com o *mouse* adiciona um destino ao final da fila de destinos do agente, ou seja, será o último destino a ser alcançado. Se

for mantida pressionada a tecla "*Ctrl*" e então realizado o clique do *mouse*, será adicionado um destino prioritário, neste caso o destino atual do agente é retornado para a primeira posição da fila se tornando o próximo destino a ser executado, e o novo destino é definido como ativo.

Além de adicionar novos destinos, ainda é possível definir que o destino que está sendo executado atualmente é inatingível. Para isso basta pressionar a tecla F2.

Destaca-se também que a plataforma permite adicionar mais de um agente ao mesmo ambiente. Para tal, basta clicar novamente no botão de iniciar a simulação enquanto uma simulação sem repetições estiver sendo executada. Durante a simulação os agentes detectam uns aos outros como sendo obstáculos móveis. A Figura 33 exibe uma simulação com múltiplos agentes.

Figura 33 – Simulação com multiplos agentes.

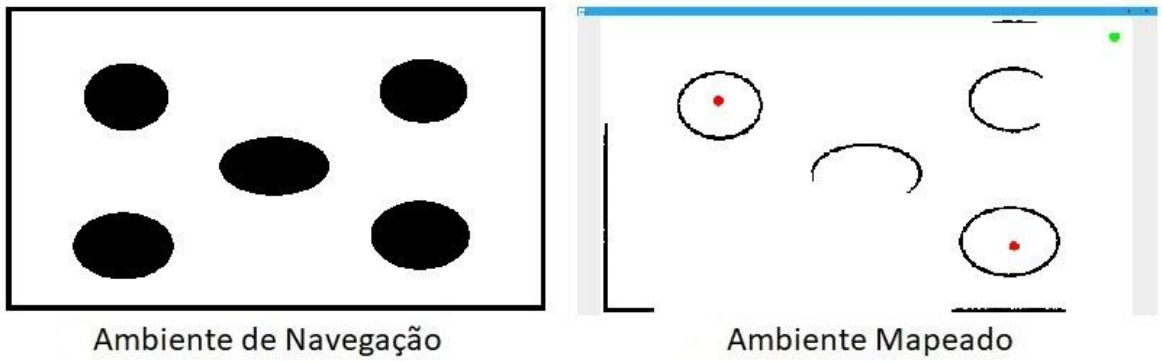


Fonte: Autor.

4.3.3. PAINEL DE MAPEAMENTO

Além da interface da Rover Sim existe mais um painel que é exibido quando o agente é inserido no ambiente. Este painel é inicializado em branco e é "desenhado" conforme o ambiente é explorado, a Figura 34 demonstra o mapeamento já realizado em um ambiente.

Figura 34 – Painel de mapeamento.



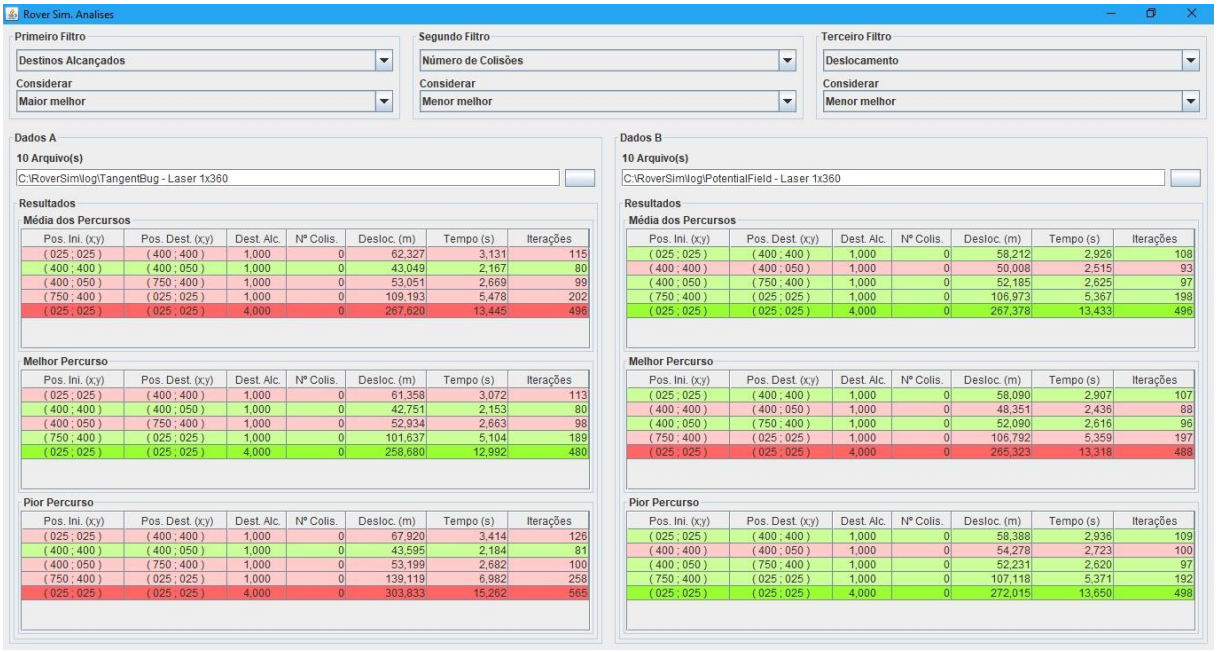
Fonte: Autor.

Além do mapeamento do ambiente este painel exibe o destino atual do agente, representado por um círculo verde, e os destinos que não puderam ser alcançados, representados por círculos vermelhos.

4.4. ROVER SIM ANÁLISES

A Rover Sim Análises é mais uma aplicação desenvolvida. Esta aplicação possui como finalidade auxiliar na análise dos arquivos gerados pela plataforma Rover Sim. Na Rover Sim Análises são carregados arquivos de simulações distintas e, através dos filtros escolhidos, a aplicação retorna qual conjunto de dados apresentou os melhores resultados. A Figura 35 exibe a interface gráfica da Rover Sim Analises.

Figura 35 – Rover Sim Análises.



Fonte: Autor.

A parte superior da aplicação apresenta os filtros disponíveis, as opções de filtro são “Destinos Alcançados”, “Número de Colisões”, “Deslocamento”, “Tempo” e “Iterações”. Além de escolher o tipo de filtro deve ser definido como ele será considerado, se um valor alto é melhor que um valor baixo deve-se escolher a opção “Maior Melhor”, se for o contrário deve-se escolher a opção “Menor Melhor”.

Podem ser aplicados até três filtros diferentes na verificação dos dados, contudo, o “Segundo Filtro” apenas será considerado se para o “Primeiro Filtro” os dois conjuntos de dados apresentarem o mesmo resultado. Do mesmo, o “Terceiro Filtro” apenas será considerado se o “Segundo Filtro” apresentar resultados iguais.

Abaixo dos filtros existem dois painéis, um apresenta as informações do conjunto de dados A e o outro o conjunto de dados B. Em cada um dos painéis existe um campo para selecionar os arquivos que serão considerados para aquele conjunto de dados, durante a seleção dos arquivos que serão considerados, pode ser escolhido um único arquivo ou vários arquivos.

Abaixo do campo de seleção dos arquivos é exibido o painel de resultados. Os resultados são divididos em três categorias, “Média dos Percursos”, “Melhor Percurso” e “Pior Percurso”.

Em cada categoria de resultados os dados são apresentados em forma de tabela. Na tabela é apresentada uma linha para cada ponto de destino do percurso, sendo estes “setores” do percurso completo. A última linha da tabela apresenta o resultado do percurso completo, desde o ponto de origem do agente até o último ponto de destino, através da soma de todos os setores do percurso.

Ao carregar os arquivos, a aplicação analisa todos os arquivos selecionados e exibe no painel de resultados a “Média dos Percursos”, os dados exibidos nesta tabela apenas irão variar quando novos arquivos forem carregados para análise. Além da média, o “Melhor Percurso” e o “Pior Percurso” são gerados.

Para gerar o melhor percurso a plataforma analisa cada setor do percurso separadamente. É analisado o primeiro setor de cada arquivo selecionado e, com base nos filtros definidos pelo usuário, o melhor primeiro setor dentre todos é escolhido, este processo se repete para os demais setores. Deste modo o “Melhor Percurso” pode ser construído a partir de fragmentos de vários arquivos. A construção do “Pior Percurso” dá-se do mesmo modo, contudo, considerando o pior caso de cada setor. Quando os filtros da aplicação são alterados a aplicação processa novamente o melhor e pior percurso.

Além de definir a média, o melhor e o pior percurso de cada conjunto de dados, a aplicação compara os dois conjuntos de dados para definir os melhores percursos de cada conjunto. O primeiro setor da “Média dos Percursos” do conjunto de dados A é comparado com primeiro setor da “Média dos Percursos” do conjunto de dados B, do mesmo modo os demais setores de cada grupo de resultados são comparados. Os percursos completos de cada grupo de resultado também são comparados entre os grupos de dados.

Durante a comparação dos setores e dos percursos, aqueles que apresentam o melhor resultado tem a linha da tabela que os representam pintada de verde e os piores tem a linha pintada de vermelho. Quando os filtros da aplicação são alterados a aplicação processa novamente a comparação entre os conjuntos de dados.

5. RESULTADOS OBTIDOS

Ao finalizar o desenvolvimento da plataforma Rover Sim e da aplicação Rover Sim Análises, iniciou-se o processo de validação dos algoritmos desenvolvidos e verificação dos resultados obtidos.

Incialmente, através de simulações, foram gerados os arquivos a serem analisados. Para as simulações algumas configurações permaneceram fixas e outras variaram. Em todas as simulações utilizaram-se fixas as seguintes configurações:

- Ambiente
  - Tamanho do Pixel: 10cm;
- Agente
  - Quantidade de Agentes: 1;
  - Tamanho: 200cm;
  - Velocidade Máxima: 20m/s;
  - Distância Segura: 30cm;
- Sensores
  - Alcance: 500cm;
  - Distância Mínima: 10cm;
  - Grau de Precisão: 2º;
  - Erro Médio: 5cm;

As configurações que variaram entre as simulações foram: o controlador do agente, a quantidade de sensores, o tipo dos sensores e o ângulo de detecção dos sensores. As combinações de configurações variantes utilizadas podem ser vistas na Tabela 2.

Tabela 2 - Combinações de configurações variantes utilizadas nas simulações.

	Controlador do Agente	Quantidade de Sensores	Tipo dos Sensores	Ângulo de detecção dos Sensores
Configuração 1	*Tangent Bug*	1	*Laser*	360º
Configuração 2	*Tangent Bug*	1	*Laser*	180º
Configuração 3	*Tangent Bug*	3	Ultrassônico	2º
Configuração 4	*Potential Field*	1	*Laser*	360º
Configuração 5	*Potential Field*	1	*Laser*	180º
Configuração 6	*Potential Field*	3	Ultrassônico	2º

Fonte: Autor.

Para as simulações realizadas na Rover Sim foram utilizados dois ambientes. Deve-se destacar que as análises dos dados foram feitas respeitando os diferentes ambientes. Os dados de simulações realizados em um mapa de ambiente apenas foram comparados com outros dados gerados por simulações no mesmo ambiente. Para cada combinação de configurações variantes foram realizadas 10 simulações em cada ambiente, resultando em um total de 120 simulações.

Após terem sido gerados os arquivos de simulação, foram realizadas as análises dos mesmos através da aplicação Rover Sim Análises. Em todas as análises foram considerados os seguintes filtros:

- Primeiro Filtro: Número de Colisões, Menor Melhor;
- Segundo Filtro: Destinos Alcançados, Maior Melhor;
- Terceiro Filtro: Deslocamento, Menor Melhor;

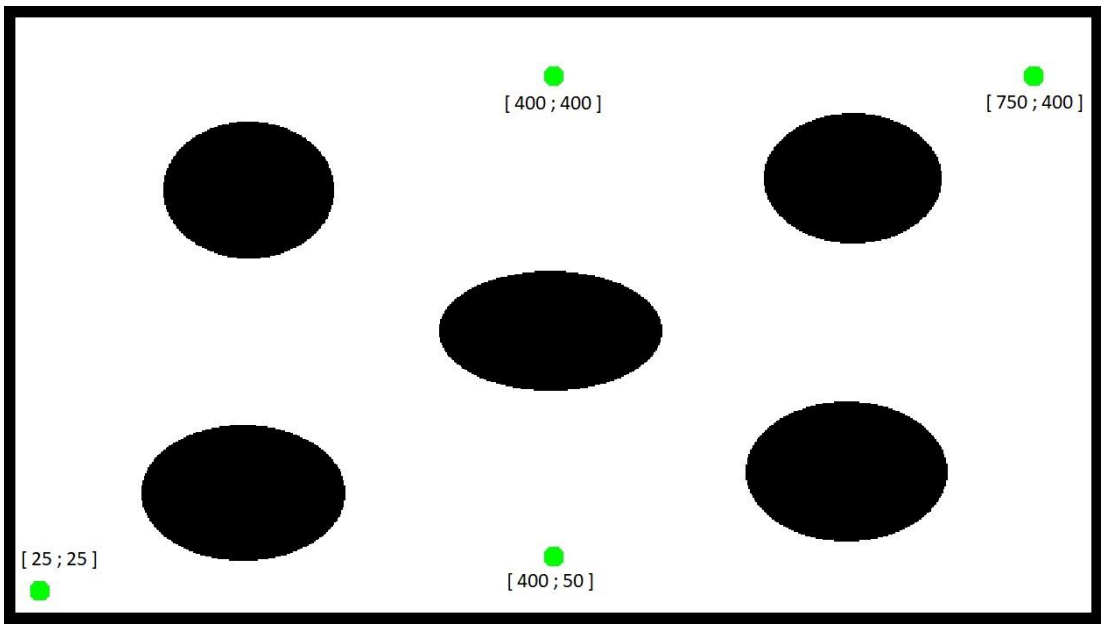
Foram realizadas análises para definir o melhor controlador para cada ambiente, como também para definir o melhor conjunto de sensores para cada controlador.

As análises feitas para a validação dos algoritmos de desvio de obstáculos, compararam as simulações que apresentavam os mesmos conjuntos de sensores. Nestes casos a diferença das simulações estava apenas no controlador do agente. Já as análises feitas para a verificação do melhor conjunto de sensores para cada algoritmo, compararam as simulações que apresentavam os mesmos controladores. Nestes casos a diferença das simulações estava no conjunto de sensores utilizados pelo agente.

5.1. ANÁLISES DO PRIMEIRO AMBIENTE

O primeiro mapa de ambiente utilizado nas simulações pode ser visto na Figura 36. Neste ambiente o agente iniciou as simulações na posição [25;25] e para completar o percurso ele deveria passar pelos pontos [400;400], [400;50], [750;400], retornando ao final para a posição inicial.

Figura 36 – Mapa do primeiro ambiente de simulação.



Fonte: Autor.

5.1.1. ALGORITMOS DE DESVIO DE OBSTÁCULO

A primeira análise comparou as simulações com “Configuração 1” e “Configuração 4”. Nesta análise, em todas simulações, os dois controladores realizaram o percurso sem colidir em nenhum obstáculo e alcançando todos os pontos de destino pré-estabelecidos, contudo, a distância percorrida pelo agente apresentou variação nas simulações. A “Configuração 4” apresentou melhor desempenho na Média dos Percursos e no Pior Percurso, já a “Configuração 5" teve um resultado melhor no Melhor Percurso.

A segunda análise comparou as simulações com “Configuração 2” e “Configuração 5”. Esta análise apresentou comportamento igual a primeira análise: nenhuma colisão, todos os pontos de destino alcançados e variação nas distâncias percorridas pelo agente. A “Configuração 2” apresentou melhor desempenho em todas comparações.

A terceira análise comparou as simulações com “Configuração 3” e “Configuração 6”. Nesta análise foi verificado que a “Configuração 3”, apesar de ter alcançado todos os pontos de destino, fez com que o agente colidisse com obstáculos durante o percurso em todas as simulações. A “Configuração 6” não colidiu nenhuma vez com obstáculos e alcançou todos os pontos de destino, apresentando melhor desempenho em todas as comparações.

A Tabela 3 apresenta de forma resumida os resultados obtidos na análise do desempenho dos algoritmos de desvio de obstáculo. Os resultados completos das análises podem ser encontrados no Anexo B.

Tabela 3 – Análise resumida dos algoritmos de desvio de obstáculo no primeiro ambiente.

	Média dos Percursos	Melhor Percurso	Pior Percurso
Configuração 1	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 267,620	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 258,680	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 303,833
Configuração 4	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 267,378	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 265,323	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 272,015
Configuração 2	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 273,517	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 261,416	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 302,249
Configuração 5	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 307,244	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 300,611	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 318,560
Configuração 3	Colisões: 40 Dest. Alc.: 4 Desloc. (m): 1806,213	Colisões: 14 Dest. Alc.: 4 Desloc. (m): 845,728	Colisões: 98 Dest. Alc.: 4 Desloc. (m): 1619,296
Configuração 6	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 407,375	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 325,298	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 511,488

Fonte: Autor.

5.1.2. CONJUNTO DE SENSORES

Para o algoritmo *Tangent Bug* os três conjuntos de sensores alcançaram todos os pontos de destino, porém, com a “Configuração 3” o agente colidiu em obstáculos em todas as simulações, demonstrando que este é o pior conjunto de sensores para o algoritmo. A “Configuração 1” e “Configuração 2” permitiram que o agente realizasse seu percurso sem colidir nenhuma vez, no entanto, a “Configuração 1” permitiu que o agente realizasse todas as simulações com o deslocamento menor que a “Configuração 2”, mostrando-se o melhor conjunto de sensores.

Com relação aos conjuntos de sensores para o algoritmo *Potential Field*, as três configurações proporcionaram ao agente um deslocamento sem colisões e alcançando todos os pontos de destino em todos os casos. A “Configuração 6” resultou nos maiores deslocamentos para o algoritmo, tornando-se o pior conjunto de sensores, a “Configuração 4” proporcionou os percursos com menor deslocamento, deste modo sendo o melhor conjunto de sensores. Por fim, a “Configuração 5” apresentou o segundo melhor desempenho.

A Tabela 4 apresenta de forma resumida os resultados obtidos nas análises dos conjuntos de sensores utilizados para cada algoritmo.

Tabela 4 – Análise resumida dos conjuntos de sensores no primeiro ambiente.

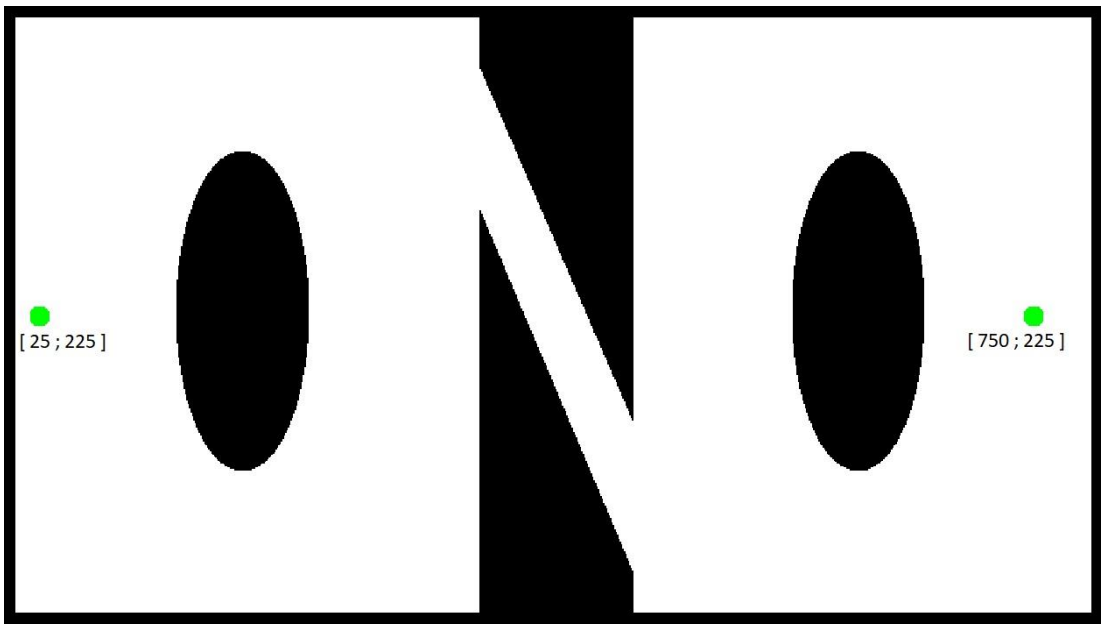
	Média dos Percursos	**Melhor Percurso**	**Pior Percurso**
Configuração 1	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 267,620	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 258,680	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 303,833
Configuração 2	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 273,517	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 261,416	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 302,249
Configuração 3	Colisões: 40 Dest. Alc.: 4 Desloc. (m): 1806,213	Colisões: 14 Dest. Alc.: 4 Desloc. (m): 845,728	Colisões: 98 Dest. Alc.: 4 Desloc. (m): 1619,296
Configuração 4	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 267,378	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 265,323	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 272,015
Configuração 5	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 307,244	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 300,611	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 318,560
Configuração 6	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 407,375	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 325,298	Colisões: 0 Dest. Alc.: 4 Desloc. (m): 511,488

Fonte: Autor.

5.2. ANÁLISES DO SEGUNDO AMBIENTE

O segundo mapa de ambiente utilizado nas simulações pode ser visto na Figura 37. Neste ambiente o agente iniciou as simulações na posição [25;225]. Durante o percurso ele deveria se deslocar até a posição [750;225] e retornar para a posição inicial.

Figura 37 – Mapa do segundo ambiente de simulação.



Fonte: Autor.

5.2.1. ALGORITMOS DE DESVIO DE OBSTÁCULO

A primeira análise comparou as simulações com “Configuração 1” e “Configuração 4”. Nesta análise a “Configuração 1”, em todas simulações, realizou o percurso sem colidir em nenhum obstáculo e alcançando todos os pontos de destino pré-estabelecidos. Já a “Configuração 4”, apesar de não colidir com nenhum obstáculo, não conseguiu realizar o percurso, mantendo o agente próximo de sua posição inicial, em todas simulações. Deste modo, a “Configuração 1” apresentou melhor desempenho em todas comparações.

A segunda análise comparou as simulações com “Configuração 2” e “Configuração 5”. Nesta análise a “Configuração 2”, em todas simulações, realizou o percurso sem colidir em nenhum obstáculo e alcançando todos os pontos de destino pré-estabelecidos. A “Configuração 5” em algumas simulações fez com que o agente colidisse com obstáculos durante sua movimentação, além disso, em todas simulações o agente não conseguiu realizar o percurso, permanecendo próximo de sua posição inicial. Assim sendo, a “Configuração 2” apresentou melhor desempenho em todas comparações.

A terceira análise foi entre as simulações com “Configuração 3” e “Configuração 6”. Nesta análise, em todas simulações, as duas configurações fizeram com que o agente colidisse

durante sua movimentação e nenhuma das configurações conseguiu realizar o percurso proposto. Deste modo, as duas configurações se mostraram igualmente ruins para o ambiente.

A Tabela 5 apresenta de forma resumida os resultados obtidos na análise do desempenho dos algoritmos de desvio de obstáculo. Os resultados completos das análises podem ser encontrados no Anexo C.

Tabela 5 – Análise resumida dos algoritmos de desvio de obstáculo no segundo ambiente.

	Média dos Percursos	**Melhor Percurso**	**Pior Percurso**
Configuração 1	Colisões: 0 Dest. Alc.: 2 Desloc. (m): 419,689	Colisões: 0 Dest. Alc.: 2 Desloc. (m): 376,573	Colisões: 0 Dest. Alc.: 2 Desloc. (m): 491,963
Configuração 4	Colisões: 0 Dest. Alc.: 1 Desloc. (m): 1207,712	Colisões: 0 Dest. Alc.: 1 Desloc. (m): 1207,240	Colisões: 0 Dest. Alc.: 1 Desloc. (m): 1208,480
Configuração 2	Colisões: 0 Dest. Alc.: 2 Desloc. (m): 325,334	Colisões: 0 Dest. Alc.: 2 Desloc. (m): 283,278	Colisões: 0 Dest. Alc.: 2 Desloc. (m): 355,574
Configuração 5	Colisões: 130 Dest. Alc.: 1 Desloc. (m): 1209,204	Colisões: 0 Dest. Alc.: 1 Desloc. (m): 1208,200	Colisões: 337 Dest. Alc.: 1 Desloc. (m): 1209,720
Configuração 3	Colisões: 36 Dest. Alc.: 0,8 Desloc. (m): 1748,049	Colisões: 7 Dest. Alc.: 1 Desloc. (m): 1210,229	Colisões: 128 Dest. Alc.: 0 Desloc. (m): 2400,840
Configuração 6	Colisões: 425 Dest. Alc.: 0,2 Desloc. (m): 2162,349	Colisões: 0 Dest. Alc.: 1 Desloc. (m): 1209,155	Colisões: 1285 Dest. Alc.: 0 Desloc. (m): 2400,340

Fonte: Autor.

5.2.2. CONJUNTO DE SENSORES

Para o algoritmo *Tangent Bug* os três conjuntos de sensores alcançaram todos os pontos de destino, porém, com a “Configuração 3” o agente colidiu em obstáculos em todas as simulações, demonstrando que este é o pior conjunto de sensores para o algoritmo. A “Configuração 1” e “Configuração 2” permitiram que o agente realizasse seu percurso sem colidir nenhuma vez, no entanto, a “Configuração 2” permitiu que o agente realizasse todas as simulações com o deslocamento menor que a “Configuração 1”, mostrando-se o melhor conjunto de sensores.

Com relação aos conjuntos de sensores para o algoritmo *Potential Field*, apenas a "Configuração 3" permitiu que o agente não colidisse com obstáculos em nenhuma simulação, mostrando-se a melhor opção. As configurações 5 e 6 fizeram com que o agente colidisse em todas as simulações, demonstrando-se igualmente ruins.

A Tabela 6 apresenta de forma resumida os resultados obtidos nas análises dos conjuntos de sensores utilizados para cada algoritmo.

Tabela 6 – Análise resumida dos conjuntos de sensores no segundo ambiente.

	Média dos Percursos	**Melhor Percurso**	**Pior Percurso**
Configuração 1	Colisões: 0 Dest. Alc.: 2 Desloc. (m): 419,689	Colisões: 0 Dest. Alc.: 2 Desloc. (m): 376,573	Colisões: 0 Dest. Alc.: 2 Desloc. (m): 491,963
Configuração 2	Colisões: 0 Dest. Alc.: 2 Desloc. (m): 325,334	Colisões: 0 Dest. Alc.: 2 Desloc. (m): 283,278	Colisões: 0 Dest. Alc.: 2 Desloc. (m): 355,574
Configuração 3	Colisões: 36 Dest. Alc.: 0,8 Desloc. (m): 1748,049	Colisões: 7 Dest. Alc.: 1 Desloc. (m): 1210,229	Colisões: 128 Dest. Alc.: 0 Desloc. (m): 2400,840
Configuração 4	Colisões: 0 Dest. Alc.: 1 Desloc. (m): 1207,712	Colisões: 0 Dest. Alc.: 1 Desloc. (m): 1207,240	Colisões: 0 Dest. Alc.: 1 Desloc. (m): 1208,480
Configuração 5	Colisões: 130 Dest. Alc.: 1 Desloc. (m): 1209,204	Colisões: 0 Dest. Alc.: 1 Desloc. (m): 1208,200	Colisões: 337 Dest. Alc.: 1 Desloc. (m): 1209,720
Configuração 6	Colisões: 425 Dest. Alc.: 0,2 Desloc. (m): 2162,349	Colisões: 0 Dest. Alc.: 1 Desloc. (m): 1209,155	Colisões: 1285 Dest. Alc.: 0 Desloc. (m): 2400,340

Fonte: Autor.

6. CONCLUSÃO

O desenvolvimento de um veículo autônomo requer que diversas áreas distintas de conhecimento se complementem, para que a partir desta colaboração possa se desenvolver um projeto. Uma das áreas fundamentais para este tipo de projeto é a computação. É responsabilidade desta área processar as informações disponibilizadas por diversos equipamentos e a partir dos resultados obtidos definir o que cada componente fará.

Dentre as principais informações a serem processadas está a definição do percurso que o veículo deve realizar. Com isto em mente, percebeu-se que caso pretenda ser desenvolvido um projeto de veículo autônomo, possuir uma ferramenta para auxiliar nas definições de estratégia de desvio de obstáculo seria de grande ajuda.

Tendo em vista os pontos citados, este trabalho realizou o desenvolvimento da Rover Sim, uma plataforma de validação de algoritmos aplicados a veículos autônomos para desvio de obstáculos. A Rover Sim foi desenvolvida de forma modular, deste modo, tornasse fácil realizar alterações em seu código, adicionando funcionalidades e opções de configuração. Deste modo pode-se melhorar a simulação realizada, auxiliando ainda mais em projetos de veículos autônomos.

Além da plataforma Rover Sim, foi desenvolvida a aplicação Rover Sim Análises. Esta aplicação não estava planejada no início do projeto, mas seu desenvolvimento mostrou-se bastante relevante para o projeto como um todo. Ela auxilia o usuário na análise dos dados gerados pelas simulações realizadas, complementando e facilitando a utilização da Rover Sim.

Através dos resultados obtidos por meio das ferramentas desenvolvidas, pode-se verificar que o trabalho realizado consegue cumprir a sua função, auxiliando na definição do melhor algoritmo de desvio de obstáculo, e do melhor conjunto de sensores a serem utilizados, em um ambiente específico.

Nas análises realizadas o algoritmo *Tangent Bug* se mostrou mais versátil que o algoritmo *Potential Field*. O *Tangente Bug* conseguiu realizar os percursos definidos nos dois ambientes utilizados, enquanto o *Potential Field* apenas conseguiu realizar o percurso definido em um dos ambientes.

Quanto aos sensores, pode-se verificar que os sensores *laser* apresentaram resultados superiores aos sensores ultrassônicos quando se utilizou o algoritmo *Tangent Bug*. Na utilização do algoritmo *Potential Field,* ambos sensores demonstraram um desempenho semelhante.

Para trabalhos futuros pode-se sugerir:

- Aperfeiçoamento do módulo de mobilidade: tratando o agente como um veículo, restringindo sua movimentação e considerando a física do corpo;
- Ajustar a plataforma de modo que os algoritmos de desvio possam ser carregados a partir de arquivo para execução na plataforma. Deste modo evitando alterações no código da plataforma para validação de novas estratégias de desvio de obstáculo;
- Ajustar a plataforma permitindo que o usuário adicione obstáculos móveis no ambiente. Deste modo melhorando a validação das estratégias de desvio.

REFERÊNCIAS

Bagnato, V. S. Os Fundamentos da Luz Laser, 2001.

Biscotto, Bernardo de Almeida. A simulação de eventos discretos em uma indústria automotiva, 2008.

Caelum. Apostila Java e Orientação a Objetos. Disponível em < https://www.caelum.com.br/apostila-java-orientacao-objetos/>. Acesso em: 16 out. 2016.

Chin Min Wei, D. Método de desvio de obstáculos aplicado em veículo autônomo, 2015.

Gonçalves, L. F. S. Desenvolvimento de sistema de navegação autônoma por GNSS, 2011.

Heinen, F. J. Sistema de Controle Híbrido para Robôs Móveis Autônomos, 2002.

Heinen, M.; Osório, F.; Heinen, F.; Kelber C. Uso de Realidade Virtual no Desenvolvimento de um Sistema de Controle do Estacionamento de Veículos Autônomos, s/d.

HRXL-MaxSonar-WR Datasheet. Disponível em <http://www.maxbotix.com/documents/HRXL-MaxSonar-WR_Datasheet.pdf>. Acesso em 24 set. 2016.

IBM. Introdução à Plataforma Eclipse. Disponível em <https://www.ibm.com/developerworks/br/library/os-eclipse-platform/>. Acesso em: 16 out. 2016.

Jung, C. R.; Osório, F. S.; Kelber, C. R.; Heinen, F. J. Computação Embarcada: Projeto e Implementação de Veículos Autônomos Inteligentes, s/d.

LMS5xx Laser Measurement Technology - Product Information. Disponível em <https://www.sick.com/media/dox/6/66/166/Product_information_LMS5xx_Laser_Measurement_Technology_en_IM0038166.PDF>. Acesso em 24 set. 2016.

Martins, A. S.; Andrade, D. Cinturão de ultra-som para o robô omni, 2005.

Moro, J. Z. Algoritmo de Bresenham: o uso microcontroladores para traçar retas em LCDs. Disponível em <http://www.demic.fee.unicamp.br/~jeff/ArquivosIndex/Bresenham>. Acesso em: 25 maio 2017.

O Globo. Uber inicia testes de rua com carro autônomo. Disponível em <http://oglobo.globo.com/sociedade/tecnologia/uber-inicia-testes-de-rua-com-carro-autonomo-19336659>. Acesso em: 02 ago. 2016.

Observatório Nacional de Segurança Viária - ONSV. 90% dos acidentes são causados por falhas humanas, alerta Observatório. Disponível em <http://www.onsv.org.br/noticias/90-dos-acidentes-sao-causados-por-falhas-humanas-alerta-observatorio/>. Acesso em: 17 jul. 2016.

PC16550D Universal Asynchronous Receiver/Transmitter With FIFOs. Disponível em <http://www.ti.com/lit/ds/symlink/pc16550d.pdf>. Acesso em 26 out. 2016.

Pereira, F. G. Navegação e Desvio de Obstáculos Usando um Robô Móvel Dotado de Sensor de Varredura Laser, 2006.

Pinheiro, Gil. A Interface Serial e o Padrão RS-232, 2011. Disponível em <https://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwjZt9CsyOXPAhWRPpAKHW_NAGAQFggcMAA&url=http%3A%2F%2Fwww.lee.eng.uerj.br%2F~gil%2Ffilas%2FPadrao%2520RS-232.pdf&usg=AFQjCNEvFVQ-uuH5G2ff0iP86-x5usGUxg&sig2=bV62JqndeJR_nANvr-TkAA&bvm=bv.135974163,d.Y2I&cad=rja>. Acesso em: 18 out. 2016.

Pissardini, R. S.; Chin Min Wei, D.; Fonseca Júnior, E. S. Veículos Autônomos: Conceitos, Histórico e Estado-da-Arte, s/d.

Revista Auto Esporte. Google inicia testes de carro em trânsito real nos Estados Unidos. Disponível em <http://revistaautoesporte.globo.com/Noticias/noticia/2015/07/google-inicia-testes-de-carro-em-transito-real-nos-estados-unidos.html>. Acesso em: 02 ago. 2016.

Ribeiro, R. C. Estratégia de desvio de obstáculos e planejamento de trajetória para uma cadeira de rodas autônoma, 2015.

Rocha, J. C. Cor Luz, Cor Pigmento e os sistemas RGB e CMY. Disponível em <http://www.belasartes.br/revistabelasartes/downloads/artigos/3/cor-luz-cor-pigmento-e-os-sistemas-rgb-e-cmy.pdf>. Acesso em: 25 maio 2017.

Souza, Tiago Francioli. A Simulação a Eventos Discretos como ferramenta de apoio à Tomada de Decisão em empresas do ramo de mineração: Aplicação em uma unidade da Yamana Gold, 2009.

Wolf, D. F.; Simões, E. V.; Osório, F. S.; Trindade Junior, O. Robótica Móvel Inteligente: Da Simulação às Aplicações no Mundo Real. 2009.

ANEXO A – *Layout* do arquivo de resultados

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
Iteracao	Tempo Iter. (s)	Tempo Total (s)	X Agente	Y Agente	Dist. Destino (m)	X Destino	Y Destino	Direcao Agente	Vel. Atual (m/s)	Vel. Max. (m/s)	Desloc. (m)	Desloc. Total (m)	Dist. O.M.P (m)	X Obst. M.P	Y Obst. M.P	Colisao	Dest. Alc.
0		0	25	25	0	25	25	0	0	20	0	0	1,672	24,416	8,289	0	0
1	0,008	0,008	25,146	26,593	52,91	400	400	-84,775	20	20	0,16	0,16	1,758	7,574	26,969	0	0
2	0,026	0,034	25,629	31,771	52,512	400	400	-84,672	20	20	0,52	0,68	1,753	8,124	32,789	0	0
3	0,027	0,061	26,155	37,145	52,098	400	400	-84,403	20	20	0,54	1,22	1,862	7,543	37,664	0	0
4	0,026	0,087	26,516	42,333	51,712	400	400	-86,019	20	20	0,52	1,74	1,891	7,711	40,35	0	0
5	0,027	0,114	26,927	47,717	51,311	400	400	-85,637	20	20	0,54	2,28	1,863	8,301	47,835	0	0
6	0,039	0,153	27,516	55,495	50,737	400	400	-85,669	20	20	0,78	3,06	1,984	7,693	56,302	0	0
7	0,025	0,178	27,79	60,487	50,379	400	400	-86,857	20	20	0,5	3,56	1,969	8,101	60,193	0	0
8	0,028	0,206	28,112	66,078	49,98	400	400	-86,703	20	20	0,56	4,12	2,075	7,449	67,994	0	0
9	0,026	0,232	28,292	71,275	49,621	400	400	-88,02	20	20	0,52	4,64	2,059	7,713	70,549	0	0
10	0,026	0,258	28,498	76,471	49,263	400	400	-87,735	20	20	0,52	5,16	2,065	7,861	77,287	0	0
11	0,028	0,286	28,701	82,067	48,882	400	400	-87,914	20	20	0,56	5,72	2,068	8,071	80,656	0	0
12	0,029	0,315	28,941	87,862	48,489	400	400	-87,634	20	20	0,58	6,3	2,056	8,395	88,711	0	0
13	0,025	0,34	29,111	92,859	48,155	400	400	-88,051	20	20	0,5	6,8	2,153	7,6	92,089	0	0
14	0,026	0,366	29,192	98,059	47,819	400	400	-89,113	20	20	0,52	7,32	2,184	7,38	99,159	0	0
15	0,027	0,393	29,292	103,458	47,472	400	400	-88,93	20	20	0,54	7,86	2,192	7,785	107,666	0	0
16	0,028	0,421	29,41	109,056	47,115	400	400	-88,793	20	20	0,56	8,42	2,156	7,857	108,758	0	0
17	0,028	0,449	29,5	114,656	46,764	400	400	-89,085	20	20	0,56	8,98	2,154	7,967	114,248	0	0
18	0,027	0,476	29,595	120,055	46,429	400	400	-88,995	20	20	0,54	9,52	2,161	7,988	120,434	0	0
19	0,026	0,502	29,692	125,254	46,11	400	400	-88,92	20	20	0,52	10,04	2,189	7,888	123,377	0	0
20	0,026	0,528	29,803	130,453	45,793	400	400	-88,78	20	20	0,52	10,56	2,152	8,286	130,16	0	0
21	0,027	0,555	29,886	135,852	45,471	400	400	-89,124	20	20	0,54	11,1	2,183	8,085	134,662	0	0
22	0,029	0,584	29,981	141,651	45,129	400	400	-89,064	20	20	0,58	11,68	2,17	8,366	139,736	0	0

ANEXO B – Resultados completos das análises do primeiro ambiente simulado

1. PRIMEIRA ANÁLISE

Tangent Bug – 1 Sensor Laser – Ângulo Detecção 360º

Resultados

Média dos Percursos

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 025)	(400 ; 400)	1,000	0	62,327	3,131	115
(400 ; 400)	(400 ; 050)	1,000	0	43,049	2,167	80
(400 ; 050)	(750 ; 400)	1,000	0	53,051	2,669	99
(750 ; 400)	(025 ; 025)	1,000	0	109,193	5,478	202
(025 ; 025)	(025 ; 025)	4,000	0	267,620	13,445	496

Melhor Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 025)	(400 ; 400)	1,000	0	61,358	3,072	113
(400 ; 400)	(400 ; 050)	1,000	0	42,751	2,153	80
(400 ; 050)	(750 ; 400)	1,000	0	52,934	2,663	98
(750 ; 400)	(025 ; 025)	1,000	0	101,637	5,104	189
(025 ; 025)	(025 ; 025)	4,000	0	258,680	12,992	480

Pior Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 025)	(400 ; 400)	1,000	0	67,920	3,414	126
(400 ; 400)	(400 ; 050)	1,000	0	43,595	2,184	81
(400 ; 050)	(750 ; 400)	1,000	0	53,199	2,682	100
(750 ; 400)	(025 ; 025)	1,000	0	139,119	6,982	258
(025 ; 025)	(025 ; 025)	4,000	0	303,833	15,262	565

Potential Field – 1 Sensor Laser – Ângulo Detecção 360º

Resultados

Média dos Percursos

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 025)	(400 ; 400)	1,000	0	58,212	2,926	108
(400 ; 400)	(400 ; 050)	1,000	0	50,008	2,515	93
(400 ; 050)	(750 ; 400)	1,000	0	52,185	2,625	97
(750 ; 400)	(025 ; 025)	1,000	0	106,973	5,367	198
(025 ; 025)	(025 ; 025)	4,000	0	267,378	13,433	496

Melhor Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 025)	(400 ; 400)	1,000	0	58,090	2,907	107
(400 ; 400)	(400 ; 050)	1,000	0	48,351	2,436	88
(400 ; 050)	(750 ; 400)	1,000	0	52,090	2,616	96
(750 ; 400)	(025 ; 025)	1,000	0	106,792	5,359	197
(025 ; 025)	(025 ; 025)	4,000	0	265,323	13,318	488

Pior Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 025)	(400 ; 400)	1,000	0	58,388	2,936	109
(400 ; 400)	(400 ; 050)	1,000	0	54,278	2,723	100
(400 ; 050)	(750 ; 400)	1,000	0	52,231	2,620	97
(750 ; 400)	(025 ; 025)	1,000	0	107,118	5,371	192
(025 ; 025)	(025 ; 025)	4,000	0	272,015	13,650	498

2. SEGUNDA ANÁLISE

Tangent Bug – 1 Sensor Laser – Ângulo Detecção 180º

Resultados

Média dos Percursos

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 025)	(400 ; 400)	1,000	0	64,229	3,227	121
(400 ; 400)	(400 ; 050)	1,000	0	43,036	2,173	81
(400 ; 050)	(750 ; 400)	1,000	0	53,598	2,693	101
(750 ; 400)	(025 ; 025)	1,000	0	112,654	5,645	211
(025 ; 025)	(025 ; 025)	4,000	0	273,517	13,738	514

Melhor Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 025)	(400 ; 400)	1,000	0	63,986	3,210	120
(400 ; 400)	(400 ; 050)	1,000	0	42,787	2,161	81
(400 ; 050)	(750 ; 400)	1,000	0	52,905	2,654	99
(750 ; 400)	(025 ; 025)	1,000	0	101,738	5,094	191
(025 ; 025)	(025 ; 025)	4,000	0	261,416	13,119	491

Pior Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 025)	(400 ; 400)	1,000	0	64,461	3,241	122
(400 ; 400)	(400 ; 050)	1,000	0	43,479	2,191	82
(400 ; 050)	(750 ; 400)	1,000	0	58,239	2,926	109
(750 ; 400)	(025 ; 025)	1,000	0	138,070	6,921	258
(025 ; 025)	(025 ; 025)	4,000	0	304,249	15,279	571

Potential Field – 1 Sensor Laser – Ângulo Detecção 180º

Resultados

Média dos Percursos

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 025)	(400 ; 400)	1,000	0	58,724	2,948	110
(400 ; 400)	(400 ; 050)	1,000	0	86,471	4,335	162
(400 ; 050)	(750 ; 400)	1,000	0	52,859	2,651	99
(750 ; 400)	(025 ; 025)	1,000	0	109,190	5,473	204
(025 ; 025)	(025 ; 025)	4,000	0	307,244	15,407	575

Melhor Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 025)	(400 ; 400)	1,000	0	58,550	2,947	110
(400 ; 400)	(400 ; 050)	1,000	0	80,296	4,026	150
(400 ; 050)	(750 ; 400)	1,000	0	52,787	2,640	99
(750 ; 400)	(025 ; 025)	1,000	0	108,978	5,463	204
(025 ; 025)	(025 ; 025)	4,000	0	300,611	15,076	563

Pior Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 025)	(400 ; 400)	1,000	0	59,253	2,967	111
(400 ; 400)	(400 ; 050)	1,000	0	96,980	4,870	181
(400 ; 050)	(750 ; 400)	1,000	0	52,921	2,668	100
(750 ; 400)	(025 ; 025)	1,000	0	109,406	5,482	205
(025 ; 025)	(025 ; 025)	4,000	0	318,560	15,987	597

3. TERCEIRA ANÁLISE

Tangent Bug – 3 Sensores Ultrassônicos – Ângulo Detecção 2º

Resultados

Média dos Percursos

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 025)	(400 ; 400)	1,000	8	170,347	8,527	319
(400 ; 400)	(400 ; 050)	1,000	14	1398,456	69,933	2617
(400 ; 050)	(750 ; 400)	1,000	3	77,716	3,901	146
(750 ; 400)	(025 ; 025)	1,000	15	159,694	8,001	299
(025 ; 025)	(025 ; 025)	4,000	40	1806,213	90,362	3381

Melhor Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 025)	(400 ; 400)	1,000	0	72,275	3,636	136
(400 ; 400)	(400 ; 050)	1,000	7	550,180	27,536	1032
(400 ; 050)	(750 ; 400)	1,000	0	76,624	3,853	144
(750 ; 400)	(025 ; 025)	1,000	7	146,649	7,355	275
(025 ; 025)	(025 ; 025)	4,000	14	845,728	42,380	1587

Pior Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 025)	(400 ; 400)	1,000	30	175,478	8,782	328
(400 ; 400)	(400 ; 050)	1,000	28	1191,686	59,604	2233
(400 ; 050)	(750 ; 400)	1,000	10	87,078	4,378	164
(750 ; 400)	(025 ; 025)	1,000	30	165,054	8,273	310
(025 ; 025)	(025 ; 025)	4,000	98	1619,296	81,037	3035

Potential Field – 3 Sensores Ultrassônicos – Ângulo Detecção 2º

Resultados

Média dos Percursos

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 025)	(400 ; 400)	1,000	0	57,367	2,886	108
(400 ; 400)	(400 ; 050)	1,000	0	184,017	9,216	345
(400 ; 050)	(750 ; 400)	1,000	0	52,578	2,642	99
(750 ; 400)	(025 ; 025)	1,000	0	113,413	5,683	213
(025 ; 025)	(025 ; 025)	4,000	0	407,375	20,427	765

Melhor Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 025)	(400 ; 400)	1,000	0	57,228	2,864	107
(400 ; 400)	(400 ; 050)	1,000	0	106,604	5,335	200
(400 ; 050)	(750 ; 400)	1,000	0	52,460	2,624	98
(750 ; 400)	(025 ; 025)	1,000	0	109,006	5,454	204
(025 ; 025)	(025 ; 025)	4,000	0	325,298	16,277	609

Pior Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 025)	(400 ; 400)	1,000	0	57,526	2,892	108
(400 ; 400)	(400 ; 050)	1,000	0	285,244	14,286	535
(400 ; 050)	(750 ; 400)	1,000	0	52,675	2,648	99
(750 ; 400)	(025 ; 025)	1,000	0	116,043	5,815	217
(025 ; 025)	(025 ; 025)	4,000	0	511,488	25,641	959

ANEXO C – Resultados completos das análises do segundo ambiente simulado

1. PRIMEIRA ANÁLISE

Tangent Bug – 1 Sensor Laser – Ângulo Detecção 360º

Resultados

Média dos Percursos

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 225)	(750 ; 225)	1,000	0	212,727	10,642	423
(750 ; 225)	(025 ; 225)	1,000	0	206,962	10,357	415
(025 ; 225)	(025 ; 225)	2,000	0	419,689	20,999	838

Melhor Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 225)	(750 ; 225)	1,000	0	203,292	10,175	405
(750 ; 225)	(025 ; 225)	1,000	0	173,281	8,668	347
(025 ; 225)	(025 ; 225)	2,000	0	376,573	18,843	752

Pior Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 225)	(750 ; 225)	1,000	0	247,749	12,402	494
(750 ; 225)	(025 ; 225)	1,000	0	244,214	12,216	491
(025 ; 225)	(025 ; 225)	2,000	0	491,963	24,618	985

Potential Field – 1 Sensor Laser – Ângulo Detecção 360º

Resultados

Média dos Percursos

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 225)	(750 ; 225)	0,000	0	1200,226	60,011	2265
(099 ; 225)	(025 ; 225)	1,000	0	7,486	0,386	15
(025 ; 225)	(025 ; 225)	1,000	0	1207,712	60,397	2280

Melhor Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 225)	(750 ; 225)	0,000	0	1200,020	60,001	2309
(097 ; 225)	(025 ; 225)	1,000	0	7,220	0,361	14
(025 ; 225)	(025 ; 225)	1,000	0	1207,240	60,362	2323

Pior Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 225)	(750 ; 225)	0,000	0	1200,540	60,027	2048
(104 ; 225)	(025 ; 225)	1,000	0	7,940	0,416	16
(025 ; 225)	(025 ; 225)	1,000	0	1208,480	60,443	2064

2. SEGUNDA ANÁLISE

Tangent Bug – 1 Sensor Laser – Ângulo Detecção 180º

Resultados

Média dos Percursos

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 225)	(750 ; 225)	1,000	0	167,310	8,370	334
(750 ; 225)	(025 ; 225)	1,000	0	158,024	7,912	316
(025 ; 225)	(025 ; 225)	2,000	0	325,334	16,282	650

Melhor Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 225)	(750 ; 225)	1,000	0	141,964	7,102	286
(750 ; 225)	(025 ; 225)	1,000	0	141,314	7,069	282
(025 ; 225)	(025 ; 225)	2,000	0	283,278	14,171	568

Pior Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 225)	(750 ; 225)	1,000	0	178,735	8,943	361
(750 ; 225)	(025 ; 225)	1,000	0	176,839	8,858	356
(025 ; 225)	(025 ; 225)	2,000	0	355,574	17,801	717

Potential Field – 1 Sensor Laser – Ângulo Detecção 180º

Resultados

Média dos Percursos

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 225)	(750 ; 225)	0,000	130	1200,270	60,014	2310
(119 ; 225)	(025 ; 225)	1,000	0	8,934	0,460	18
(025 ; 225)	(025 ; 225)	1,000	130	1209,204	60,474	2328

Melhor Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 225)	(750 ; 225)	0,000	0	1200,300	60,015	2311
(104 ; 225)	(025 ; 225)	1,000	0	7,900	0,416	16
(025 ; 225)	(025 ; 225)	1,000	0	1208,200	60,431	2327

Pior Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 225)	(750 ; 225)	0,000	337	1200,140	60,007	2315
(121 ; 225)	(025 ; 225)	1,000	0	9,580	0,491	19
(025 ; 225)	(025 ; 225)	1,000	337	1209,720	60,498	2334

3. TERCEIRA ANÁLISE ALGORITMOS

Tangent Bug – 3 Sensores Ultrassônicos – Ângulo Detecção 2º

Resultados

Média dos Percursos

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 225)	(750 ; 225)	0,000	26	1200,362	60,018	2285
(163 ; 265)	(025 ; 225)	0,800	10	547,687	27,395	1047
(025 ; 225)	(025 ; 225)	0,800	36	1748,049	87,413	3332

Melhor Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 225)	(750 ; 225)	0,000	6	1200,340	60,017	2232
(122 ; 242)	(025 ; 225)	1,000	1	9,889	0,512	20
(025 ; 225)	(025 ; 225)	1,000	7	1210,229	60,529	2252

Pior Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 225)	(750 ; 225)	0,000	73	1200,440	60,022	2304
(604 ; 153)	(025 ; 225)	0,000	55	1200,400	60,020	2289
(025 ; 225)	(025 ; 225)	0,000	128	2400,840	120,042	4593

Potential Field – 3 Sensores Ultrassônicos – Ângulo Detecção 2º

Resultados

Média dos Percursos

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 225)	(750 ; 225)	0,000	274	1200,232	60,012	2316
(338 ; 214)	(025 ; 225)	0,200	151	962,117	48,108	1857
(025 ; 225)	(025 ; 225)	0,200	425	2162,349	108,120	4173

Melhor Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 225)	(750 ; 225)	0,000	0	1200,160	60,008	2303
(114 ; 215)	(025 ; 225)	1,000	0	8,995	0,468	18
(025 ; 225)	(025 ; 225)	1,000	0	1209,155	60,476	2321

Pior Percurso

Pos. Ini. (x;y)	Pos. Dest. (x;y)	Dest. Alc.	Nº Colis.	Desloc. (m)	Tempo (s)	Iterações
(025 ; 225)	(750 ; 225)	0,000	871	1200,020	60,001	2302
(328 ; 213)	(025 ; 225)	0,000	414	1200,320	60,016	2316
(025 ; 225)	(025 ; 225)	0,000	1285	2400,340	120,017	4618