

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MANOELA GONÇALVES PIZYBLSKI

**ESTUDO E ANÁLISE COMPARATIVA DE ALGORITMOS DE
PLANEJAMENTO DE CAMINHO PARA VEÍCULOS AUTÔNOMOS**

DISSERTAÇÃO

PONTA GROSSA

2020

MANOELA GONÇALVES PIZYBLSKI

**ESTUDO E ANÁLISE COMPARATIVA DE ALGORITMOS DE
PLANEJAMENTO DE CAMINHO PARA VEÍCULOS AUTÔNOMOS**

**STUDY AND COMPARATIVE ANALYSIS OF PATH PLANNING ALGORITHMS
FOR AUTONOMOUS VEHICLES**

Dissertação apresentada como requisito para
obtenção do título de Mestre em Engenharia Elétrica
da Universidade Tecnológica Federal do Paraná
(UTFPR).

Orientador: Prof. Dr. Max Mauro Dias Santos.

PONTA GROSSA

2020



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite que outros remixem, adaptem e criem a partir do trabalho para fins não comerciais, desde que atribuam o devido crédito e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



**Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Ponta Grossa**



MANOELA GONCALVES PIZYBLSKI

ESTUDO E ANÁLISE COMPARATIVA DE ALGORITMOS DE PLANEJAMENTO DE CAMINHO PARA VEÍCULOS AUTÔNOMOS

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestra Em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Controle E Processamento De Energia.

Data de aprovação: 18 de Dezembro de 2020

Prof Max Mauro Dias Santos, Doutorado - Universidade Tecnológica Federal do Paraná

Prof Evandro Leonardo Silva Teixeira, Doutorado - Universidade de Brasília (Unb)

Prof.a Fernanda Cristina Correa, Doutorado - Universidade Tecnológica Federal do Paraná

Prof Mauricio Aparecido Ribeiro, Doutorado - Universidade Tecnológica Federal do Paraná (Utfpr)

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 18/12/2020.

AGRADECIMENTOS

Primeiramente agradeço à Deus pelas bênçãos recebidas a cada dia nessa passagem aqui na Terra, pela vida e por todos os momentos que posso viver.

Agradeço aos meus pais por todo o empenho exemplar em sempre oferecer o melhor para mim, com muita honestidade, amor, carinho e o melhor de tudo, por sempre acreditarem no meu potencial e me apoiarem.

Agradeço ao irmão que eu tive a sorte de ter ao meu lado nessa vida, muito carinhoso e também sempre com uma piadinha para me fazer sorrir nos momentos difíceis.

Aos meus exemplares avós pela eterna torcida na arquibancada da vida, cobrindo a minha trajetória acadêmica com chuvas de bênçãos. Fazendo com que eu sempre tenha mais empenho e paciência para continuar dando orgulho a eles.

Agradeço ao Denis Takeo Toyoshima, pela paciência e amor dedicados a mim e por sempre estudar comigo. Por estar presente em todos momentos ajudando-me quando necessário.

Ao meu orientador, Professor Doutor Max Mauro Dias Santos, por desde 2016 proporcionar a mim diferentes conhecimentos sobre a engenharia automotiva, propiciando assim o descobrimento de uma paixão por essa área. E claro, por toda a paciência dedicada a mim durante todos esses anos.

Ao Campus Ponta Grossa da Universidade Tecnológica Federal do Paraná por oferecer uma estrutura maravilhosa que promoveu a mim momentos marcantes.

Aos integrantes da banca, Professor Evandro, Professora Fernanda e Professor Mauricio pelas orientações oferecidas para o enriquecimento desse trabalho.

Ao Alberto Shimahara da OppenCadd pelas dicas e apoio durante a elaboração desse trabalho.

E por fim, a todos os meus colegas da vida acadêmica que sempre estiveram dispostos a me ajudar e a torcer por mim. A todos que fazem parte da minha história, o meu muito obrigada!

RESUMO

PIZYBLSKI, Manoela Gonçalves. **Estudo e Análise Comparativa de Algoritmos de Planejamento de Caminho para Veículos Autônomos**. 2020. 86 f. Dissertação (Mestrado em Engenharia Elétrica) – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2020.

O veículo terrestre autônomo pode transportar cargas e pessoas do ponto de origem ao destino por meio de um sistema composto por sua percepção do ambiente em que está inserido, planejamento de trajetória e atuação. A percepção abrange vários sensores combinados capazes de promover a percepção do ambiente, bem como identificar objetos e elementos. Quando se trata de planejamento de trajetórias, existem algoritmos e estratégias utilizadas nas manobras realizadas de forma exata e segura, obedecendo às regras de trânsito. Em relação à ação, estratégias de controle longitudinal e lateral estão presentes para movimentar o veículo. Especificamente, o planejamento do caminho é responsável pelo deslocamento estável e confortável do usuário durante o planejamento, manobra e trajetória do caminho. Neste trabalho dois algoritmos de planejamento de trajetória aplicados a veículos autônomos são apresentados, analisados e colocados lado a lado. Um mapa de um estacionamento foi definido representando um cenário para avaliar e comparar o desempenho de ambos os algoritmos com relação ao planejamento do caminho e manobras de veículos dentro das vagas de estacionamento.

Palavras-chave: Veículos Autônomos. Planejamento de movimento. Modelagem e simulação. Algoritmos.

ABSTRACT

PIZYBLSKI, Manoela Gonçalves. **Study and Comparative Analysis of Path Planning Algorithms for Autonomous Vehicles**. 2020. 86 p. Dissertation (Master Degree in Electric Engineering) - Programa de Pós-Graduação em Engenharia Elétrica, Federal Technology University - Parana. Ponta Grossa, 2020.

The autonomous land vehicle can carry loads and people from an origin point to the destination using a system composed by its perception of the environment in which it is inserted, path planning and action. The perception covers several combined sensors able of promoting the environment perception as well as identifying objects and elements. When it comes to path planning, there are algorithms and strategies used in manoeuvres performing in the exact and safe way, following traffic rules. Regarding action, strategies of longitudinal and lateral control are present in order to move the vehicle. Specifically, the path planning is responsible for stable and comfortable user's displacement during path planning, manoeuvre and trajectory. In this work two algorithms of path planning applied to autonomous vehicles are presented, analysed, and set side by side. A map of a parking was defined representing a scenario to evaluate and compare both algorithms performances with respect to path planning and vehicles manoeuvres within parking slots.

Keywords: Autonomous Vehicles. Path Planning. Modelling and Simulation. Algorithms.

LISTA DE FIGURAS

Figura 1 - Veículo Stanley Vencedor do DARPA 2005.....	16
Figura 2 - Descrição dos Níveis de Automação Segundo SAE J3016	18
Figura 3 - Sistema <i>Park Assist</i>	19
Figura 4 – Estrutura de Veículos Autônomos.....	20
Figura 5 – Principais Sensores Presentes nos Veículos Autônomos	21
Figura 6 - Sistema LiDAR.....	22
Figura 7 – Percepção de Pedestres na Pista	23
Figura 8 - Cálculo da Distância pelo Sensor Ultrassônico.....	24
Figura 9 – Modulação Clássica FMCW (a) Sinal Transmitido e sinal recebido; (b)Frequência Resultante	26
Figura 10 - Trilateração em 2D.....	28
Figura 11 - Sistemas Autônomos	30
Figura 12– Estrutura Hierárquica Utilizada em Sistemas de Controle de VAS	33
Figura 13 – Modelo Bicicleta da Cinemática	38
Figura 14 - Exemplo de Mapa de Grade	42
Figura 15 - Sistema de Tomada de Decisão de um Veículo Autônomo	46
Figura 16 - Gráficos de Dois Caminhos Diferentes	49
Figura 17 - Elipse de Fricção do Veículo.....	50
Figura 18 - Representação do Algoritmo de Busca em Largura.....	54
Figura 19 - Representação do Algoritmo de Busca em Profundidade.....	55
Figura 20 - Processo de Expansão da Árvore Algoritmo RRT	56
Figura 21 - Representação do Algoritmo A*	57
Figura 22 - Representação do Algoritmo de Dijkstra.....	58
Figura 23 - Representação do Ambiente Considerado para Simulação.....	61
Figura 24 - Representação de um Mapa de Custos de Amostra e Estados de Células	61
Figura 25 - Mapa de Custos da Simulação	62
Figura 26 - Dimensões do Veículo	63
Figura 27 - Relação entre o Número de Círculos e Raio de Inflação	64
Figura 28 - Comparação entre Algoritmo A* e Algoritmo Híbrido A*	66
Figura 29 – Posições Conectadas Usando Modelo <i>Reeds-Shepp</i>	67
Figura 30 - Representação dos Pontos Inicial e Objetivo no Mapa de Custos.....	68
Figura 31 - Duas Classes de Caminho Mais Curto para um Veículo Sem Marcha Ré	71
Figura 32 - Representação do Caminho Encontrado pelo Algoritmo Híbrido A*	73
Figura 33 - Representação dos Caminhos Encontrados pelo Algoritmo RRT (1)	74
Figura 34 - Representação dos Caminhos Encontrados pelo Algoritmo RRT (2)	75
Figura 35 – Relação da Posição do Veículo com as Coordenadas Globais	77

Figura 36 - Relação Geométrica Controlador Stanley.....	78
Figura 37 - Gráfico da Relação da Variação dos Tempos de Simulação para cada Algoritmo	81
Figura 38 – Gráfico de Desvio Padrão da Relação da Variação dos Tempos de Simulação para cada Algoritmo.....	82

LISTA DE TABELAS

Tabela 1 - Trabalhos Realizados Utilizando Controladores e Sensores para Veículos Autônomos	32
Tabela 2 - Dimensões do Veículo	64
Tabela 3 - Parâmetros de Simulação Utilizados para o Algoritmo Híbrido A*	69
Tabela 4 - Parâmetros Utilizados na Simulação do Algoritmo RRT	71
Tabela 5 - Valores de Distância Percorrida Obtidos.....	76
Tabela 6 - Variação do Ângulo de Esterçamento em Cada Execução do Algoritmo RRT	79
Tabela 7 - Ângulos de Esterçamento Obtidos nos Pontos do Algoritmo Híbrido A* ..	80
Tabela 8 – Tempos de Simulação Obtidos	80

SUMÁRIO

1 INTRODUÇÃO	12
1.1 OBJETIVOS	13
1.1.1 Objetivo Geral	13
1.1.2 Objetivos Específicos	14
1.2 ORGANIZAÇÃO DO TRABALHO	14
2 ESTADO DA ARTE	15
2.1 HISTÓRIA E DESAFIOS DOS VEÍCULOS AUTÔNOMOS	15
2.2 NÍVEIS DE AUTONOMIA	17
2.3 PRINCIPAIS COMPONENTES DOS VEÍCULOS AUTÔNOMOS	20
2.3.1 Sensores	21
2.3.2 Fusão de Sensores	29
2.3.3 Controladores	31
2.3.4 Atuadores	33
2.4 CONCLUSÃO	34
3 ESTRATÉGIAS DO PLANEJAMENTO DE MOVIMENTO	35
3.1 PLANEJAMENTO DE MOVIMENTO E OS MODELOS DE ESTADOS	37
3.1.1 Modelo de Estado do Veículo	37
3.1.1.1 Modelo bicicleta da cinemática (<i>Kinematic Bicycle Model</i>)	38
3.1.2 Modelos de Estado do Ambiente	41
3.1.2.1 Mapas em Grade	42
3.1.2.2 Mapas Geométricos	43
3.1.2.3 Mapas Sensoriais	44
3.1.2.4 Mapas Topológicos	44
3.2 TÉCNICAS DE PLANEJAMENTO DE MOVIMENTO	44
3.2.1 Planejamento de Caminho	46
3.2.2 Planejamento de Comportamento	47
3.2.3 Planejamento de Movimento	48
3.2.4 Planejamento de Trajetórias e de Manobras	51
3.2.5 Controle Físico do Veículo	52
3.3 ALGORITMOS DE BUSCA DE CAMINHOS	52
3.3.1 Algoritmo de Busca em Largura (<i>Breadth-First Search – BFS</i>)	53
3.3.2 Algoritmo de Busca em Profundidade (<i>Depth-First Search – DFS</i>)	54
3.3.3 Algoritmo de Árvores Aleatórias De Rápida Exploração (<i>Rapidly-exploring Random Trees – RRT</i>)	55
3.3.4 Algoritmo A* (<i>A-Star</i>)	57
3.3.5 Algoritmo de Dijkstra	58
3.4 CONCLUSÃO	59
4 METODOLOGIA	60

4.1	MODELO DE REPRESENTAÇÃO DO AMBIENTE	60
4.2	REPRESENTAÇÃO DO VEÍCULO	63
4.3	APLICAÇÃO DE ALGORITMOS DE BUSCA DE CAMINHOS	65
4.3.1	Simulação Utilizando o Algoritmo A* Híbrido (<i>Hybrid A-Star</i>)	65
4.3.1	Simulação Utilizando o Algoritmo De Árvores Aleatórias De Rápida Exploração (<i>Rapidly-exploring Random Trees – RRT</i>)	69
4.4	CONCLUSÃO	72
5	SIMULAÇÃO E ANÁLISE DE RESULTADOS	73
5.1	CONCLUSÃO	83
6	CONCLUSÕES E PERSPECTIVAS.....	84
	REFERÊNCIAS.....	86
	APÊNDICE A	97
	APÊNDICE B	100

1 INTRODUÇÃO

A utilização dos veículos de transporte acontece pela condução dos motoristas, os quais são responsáveis pela tarefa de deslocamento de forma segura e eficiente. Porém, o número de acidentes de trânsito cresce de forma proporcional a quantidade de veículos circulantes e a principal causa destacada para esses acidentes é a distração. Dessa forma, a tarefa de condução deve ser única para que exista um sistema de transporte seguro para a humanidade. A condução autônoma apresentada na ideia de veículos autônomos, mostra alguns benefícios para a sociedade em geral. Primeiro benefício a ser citado converge com segurança e diminuição do risco de acidentes causados pela distração humana. Há também a capacidade de melhoria da mobilidade e até ocupação de veículos estacionados nas cidades (OLIVEIRA, 2013).

Sabe-se que a pesquisa e os experimentos com carros sem motoristas tiveram início nos anos 70, porém o desenvolvimento e aprimoramento dessa pesquisa acabaram sofrendo pela falta de tecnologia necessária para o sucesso dela. Mas, com o passar do tempo a tecnologia de computação foi se aperfeiçoando, os sistemas de localização evoluíram, e o desenvolvimento da inteligência artificial tem facultado para que a presença de veículos autônomos apresente desenvolvimento e experiências diárias.

Os sistemas empregados em veículos autônomos apresentam características que demonstram o casamento entre a inteligência artificial e a engenharia automotiva, como os algoritmos de grandes avanços e aprendizado de máquina (RAFFO e OTHERS, 2005). No início os veículos que apresentavam certa autonomia eram baseados em *hardware* resultando uma certa limitação, pois não havia maneiras de executar atualizações ou ampliações de tarefas.

Com o surgimento do software, a implantação de recursos fez com que os veículos autônomos se tornassem cada vez melhores, uma vez que o software atua como o cérebro desses veículos (LIN *et al.*, 2018). Entre as diferentes aplicações existentes de veículos autônomos atualmente está o emprego dessa tecnologia na agricultura, grande parte de tratores com alto desempenho e desenvolvimento permitem a coordenação remota de diversos tratores em uma mesma área, com uma precisão na ordem de centímetros (RODRIGUES, 2017).

Podem ser citadas também outras aplicações como da empresa Waymo pertencente a Alphabet e Google, a ideia é oferecer um serviço de táxi-robô, apresentando uma semelhança aos serviços de compartilhamento de viagem como Uber e Lyft, há um motorista humano no volante, porém só intervém em situações e emergência. Entre as mais avançadas tecnologias apresentadas em veículos autônomos está o *Autopilot* da Tesla, os veículos possuem sistemas avançados capacitados para proporcionar uma condução autônoma total. O sistema apresenta oito câmeras provendo uma visão de 360 graus de visibilidade em torno do veículo com até 250 metros de distância, uma rede neural desenvolvida pela empresa permitindo uma capacidade alta de processamento sobre os sensores de radar, sonar e de visão.

Embora a popularização dos veículos autônomos não tenha acontecido, ao ponto de substituírem plenamente veículos comuns, muitas pesquisas demonstram que esse tipo de veículo estará presente no futuro como um auxílio ao condutor. Avaliando o crescimento e desenvolvimento do assunto sobre veículos autônomos, é possível ver a necessidade de pesquisa e aplicações no ramo. Esse trabalho apresenta um estudo sobre esses veículos e simulação de dois algoritmos que auxiliam na tarefa de locomoção e manobras em estacionamento.

1.1 OBJETIVOS

Essa seção apresenta objetivo geral e objetivos específicos que englobam o desenvolvimento desse trabalho.

1.1.1 Objetivo Geral

Este trabalho tem como objetivo geral estudo e avaliação de viabilidade entre dois algoritmos utilizados para planejamento de caminhos em veículos autônomos em ambiente simulado. O primeiro algoritmo utilizado é o algoritmo A* (*A-star*) o segundo trata-se de árvores aleatórias de rápida exploração (*RRT- Rapidly-Exploring Random Trees*), os quais são bastante citados em aplicações de veículos autônomos.

1.1.2 Objetivos Específicos

Entre os objetivos específicos propostos encontram-se os seguintes:

- Apresentar o estado da arte de veículos se motorista;
- Listar os principais algoritmos de planejamento de movimento;
- Avaliar comparativamente o desempenho dos algoritmos de planejamento de caminhos para veículos sem motorista;
- Simulação e comparação entre dois algoritmos utilizados para planejamento de caminhos em veículos autônomos.

1.2 ORGANIZAÇÃO DO TRABALHO

Esse trabalho está dividido em seis partes principais. O Capítulo 2 apresenta uma breve história sobre os veículos autônomos e seus principais componentes.

No Capítulo 3, são discutidas estratégias presentes no planejamento de movimento desses veículos.

O Capítulo 4 aborda a metodologia utilizada para avaliação dos dois algoritmos escolhidos pela autora em ambiente simulado.

No Capítulo 5 são apresentados os resultados obtidos nas simulações e por fim no Capítulo 6 são abordadas as conclusões do estudo realizado.

2 ESTADO DA ARTE

Em dados fornecidos pela ONU (Organização das Nações Unidas), reduzir o índice percentual de acidentes considerados graves no trânsito tem sido um quesito de grande magnitude atualmente. Não apenas na União Europeia ou em países com maior nível de desenvolvimento tecnológico, mas sim mundialmente, pois trata-se de dados que preocupam cada vez mais a sociedade. Prognósticos mostram que se não houver mudanças visando a segurança no trânsito, esses acidentes podem se tornar a sétima predominante causa de morte até 2030 (ONU, 2018).

Com a implementação e utilização de veículos autônomos acredita-se em uma diminuição dos acidentes de trânsito. O objetivo dos veículos autônomos não é o alcance da capacidade humana, mas sim de superá-la, dessa forma tornando-os seguros e livres de falhas (NHTSA, 2012).

Esse capítulo possui como objetivo esclarecer alguns conceitos importantes a respeito de veículos autônomos e está dividido da seguinte maneira: na Seção 2.1 é descrito um breve histórico sobre os veículos em questão. Na Seção 2.2 é apresentada a classificação dos níveis de autonomia dos veículos autônomos, e na Seção 2.3. são apresentados e discutidos principais componentes dos veículos autônomos bem como a interação destes com o ambiente.

2.1 HISTÓRIA E DESAFIOS DOS VEÍCULOS AUTÔNOMOS

Um veículo autônomo ou totalmente automatizado, é aquele em que apresenta uma operação sem interferência direta de um motorista (NHTSA, 2019). O início do desenvolvimento de pesquisas para veículos autônomos aconteceu no ano de 1939, na Feira Mundial de Nova Iorque (EUA), em que o projetista Norman Melancton Bel Geddes com apoio da General Motors (GM) demonstrou a ideia de como seria o sistema rodoviário em vinte anos. A demonstração abrangia um protótipo com sistema de rodovias automatizadas em que as estradas seriam capazes de corrigir erros humanos de condução (WETMORE, 2003).

Mais tarde, em Pittsburgh a Universidade Carnegie Mellon fez uma apresentação de seus veículos Navlab que se moviam em ambientes estruturados e como grande marco destacou-se o projeto PROMETHEUS, desenvolvido na

Universidade de Parma. Este projeto demonstrou a direção autônoma em uma rodovia com três faixas e com velocidades de até 130 km/h, os veículos mudavam de faixa e conseguiam fazer o rastreamento da faixa da pista (DICKMANNNS *et al.*, 1994; THORPE *et al.*, 1988).

Com várias pesquisas visando a melhoria da tecnologia de veículos autônomos, em 2004, o DARPA *Grand Challenge* foi responsável pelo incentivo do desenvolvimento dos carros autônomos. Premiando com 1 milhão de dólares a equipe que apresentasse um veículo robótico que navegasse com sucesso em uma pista com obstáculos. Nesse primeiro evento, os competidores não conseguiram conquistar o prêmio, porém um ano mais tarde cinco veículos conseguiram completar o desafio e o primeiro prêmio foi de 2 milhões de dólares. O vencedor do prêmio foi um veículo desenvolvido pela Universidade de Stanford juntamente com a Volkswagen, com uma tecnologia adaptada do Stanford Cart, o robô denominado como Stanley, Figura 1, conseguiu completar o percurso proposto de 210 Km em menos de 7 horas (THRUN *et al.*, 2006).

Figura 1 - Veículo Stanley Vencedor do DARPA 2005



Fonte: (STRICKLAND, 2005)

Em 2007, surgiu uma nova competição também desenvolvida pela DARPA, desafio urbano, em que os participantes precisaram enfrentar desafios como tráfego, obstáculos e respeitar as leis de trânsito (URMSON *et al.*, 2007). Mais tarde, em 2013 na Itália o BRAiVE, carro autônomo do VisLab dirigiu no centro da cidade de Parma. Esse projeto é considerado pelos autores como o primeiro veículo sem motorista que

precisou interpretar rotatórias, cruzamentos, faixas de pedestres (RUDOLPH e VOELZKE, 2017).

Diferentes competições foram realizadas ao redor do mundo com o objetivo de testar e demonstrar a tecnologia e a capacidade que um veículo autônomo pode apresentar. Embora a concretização de provas e competições envolvendo veículos autônomos tenham acontecido, é fácil observar alguns desafios para que esses veículos estejam nas ruas (BROWN, 2017). Desafios voltados para quesitos governamentais, educacionais e principalmente tecnológicos.

O desafio tecnológico do desenvolvimento de sistemas autônomos seguros juntamente com os desafios sociais e educacionais, são importantes para que exista uma perspectiva voltada para o design juntamente com o ser humano (BISHT, ABBOTT e GAFFAR, 2017; PARKINSON, WARD, *et al.*, 2017). Esses desafios são normais e fazem parte de qualquer tipo de evolução, porém deve-se proporcionar assistência ao condutor de forma a prover uma condução mais simples e segura. Ou seja, o motorista deve ter um carro seguro, que resulte na diminuição do número de acidentes de carro presentes nas estatísticas (BRANCO, 2018). Mas o grau de automatização oferecido ao motorista não deve progredir para uma dependência humano-máquina (HOMANN, 2005).

No momento em que grandes líderes de pesquisa e testes do seguimento autônomo são citadas, como Uber, Volvo e Google, é preciso saber que estas além de apostar no desenvolvimento e soluções de autonomia integral, também estão buscando a empatia do consumidor e do governo por veículos autônomos.


É possível encontrar iniciativas do governo que apoiam a atuação dos veículos autônomos na sociedade. Na Europa, o governo alemão autorizou que a montadora Volkswagen introduzisse testes de carros autônomos no aeroporto de Hamburgo. Há também testes sendo feitos pela Volvo em Estocolmo. E nos Estados Unidos, mais precisamente na Califórnia, é fácil se deparar com carros autônomos em fases de testes circulando pelas ruas.

2.2 NÍVEIS DE AUTONOMIA

Nos últimos anos, veículos autônomos apresentaram grande evolução em se tratando de liberação para o uso do público em geral, pode-se entender isso através

de uma taxonomia de condução. Um veículo autônomo terrestre possui a capacidade de detectar o ambiente em que se encontra e efetivar um deslocamento, por meio de uma variedade de técnicas, tecnologias e sensores. A tarefa de condução autônoma pode ser dividida em níveis seguindo alguns padrões, uma taxonomia de condução bastante abordada é a proposta pela norma SAE J3016. Esse padrão foi lançado pela SAE *International* em janeiro de 2014 e estão em vigência desde então, são propostos seis níveis em que o primeiro, denominado Nível 0 é um nível em que não possui automação e o último nível, Nível 5 compreende veículos completamente automatizados, a classificação pode ser observada na Figura 2.

Figura 2 - Descrição dos Níveis de Automação Segundo SAE J3016



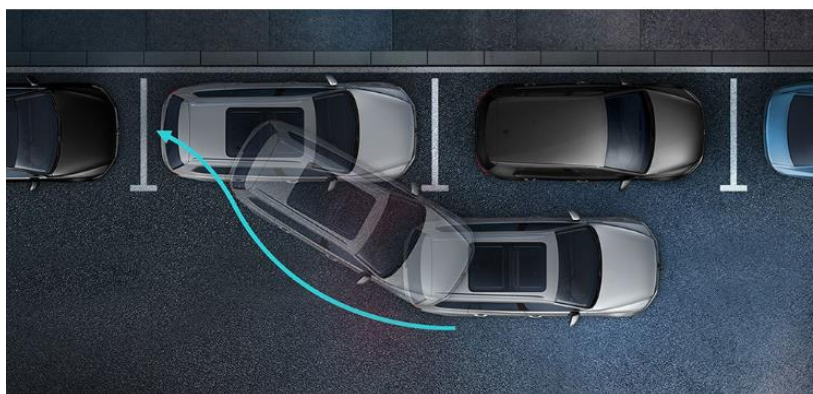
Nível 0	Nível 1	Nível 2	Nível 3	Nível 4	Nível 5
Sem Automação	Assistência ao Motorista	Automação Parcial	Automação Condicional	Alta Automação	Automação Completa
Sem autonomia. O motorista executa todas as tarefas de condução.	Veículo é controlado pelo motorista, porém alguns elementos auxiliares são incluídos no projeto do veículo.	Existe combinação de funções automatizadas. Porém o motorista está envolvido no papel de direção e monitoramento do ambiente.	O motorista se faz necessário, mas não precisa fazer monitoramento do ambiente. Deve assumir controle do veículo em casos de aviso do sistema.	O veículo é capaz de executar todas as funções em determinadas condições. E o motorista pode optar por assumir o controle.	O veículo é capaz de executar todas as funções em todas as condições. E o motorista pode optar por assumir o controle.

Fonte: Adaptado de (TRANSPORTATION, 2019)

O nível 1, denominado Assistência ao motorista, possui como característica informar ao condutor sobre situações de risco. O sistema veicular não intervém totalmente sobre a situação, fornece um aviso ao motorista recorrente de sinais, sendo eles no painel, ou sons de alarme. Nesse nível entre os sistemas mais conhecidos está o controlador adaptativo de velocidade (ACC - *Adaptive Cruise Control*), em que o veículo é capaz de acompanhar a velocidade do trânsito, freando automaticamente caso o carro da frente diminua a velocidade. Nesse nível também encontra-se o assistente de faixa de rodagem (LDWS – *Lane Departure Warning System*), nas aplicações mais simples o sistema é capaz de analisar a estrada por meio de uma câmera e detecta se o veículo sair da faixa de rodagem, emitindo um aviso sonoro ao motorista.

No terceiro nível de automação, nível 2, o veículo é capacitado para avisar o motorista e, ajudar na correção, além dos sistemas apresentados no nível 1, há o sistema de assistência de faixa (LKAS - *Lane Keeping Assist System*) esse sistema também faz a detecção de faixas e faz o ajuste da direção auxiliando o motorista a manter o veículo centralizado nas linhas da faixa de rodagem. Incluso também nesse nível o assistente de estacionamento automático (APA - *Automatic Parking Assist*), Figura 3. Esse sistema é caracterizado por facilitar as tarefas de balizas e manobras, cabe ao motorista fazer a frenagem, aceleração e troca de marchas, sem a necessidade de tocar no volante.

Figura 3 - Sistema Park Assist



Fonte: (CONTINENTAL, 2019)

A intervenção do veículo caracteriza o quarto nível de automação, nível 3, essa ainda apresenta dependência com o grau de interferência, podendo ser semi-automatizada ou altamente automatizada. Encontra-se nesse nível o sistema automático de frenagem de emergência (AEB - *Automatic Emergency Braking*), capacitado por detectar uma colisão frontal eminente podendo atenuar ou evitar a colisão. Outro tipo de intervenção que o veículo pode apresentar nesse nível é o monitoramento do motorista (DM - *Driver Monitoring*), uma câmera monitora a atenção do motorista, alertando-o caso sejam detectados sinais de sonolência ou distração.

A diferença entre um veículo autônomo de nível 3 e um veículo autônomo de nível 4 é que neste o veículo pode assumir o controle do veículo, os sistemas são semelhantes aos presentes nos níveis anteriores, mudando a evolução tecnológica e a presença da fusão de sensores com o emprego de algoritmos que auxiliam a operação de direção. No último nível de automação, nível 5, estão os veículos capazes

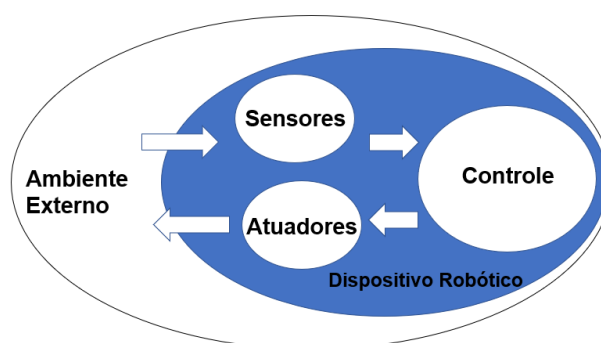
de solucionar alguns problemas, sob circunstâncias ou condições ambientais. Os veículos desse nível de automação não apresentam pedais e volante. Conforme a regulamentação imposta pela SAE J1936 as empresas são responsáveis pelos sistemas de automação e pela definição dos requisitos, bem como pela forma em que o nível de automação será aplicado.

2.3 PRINCIPAIS COMPONENTES DOS VEÍCULOS AUTÔNOMOS

Os veículos autônomos detalhados nesse trabalho podem ser entendidos através de sua estrutura, a qual apresenta três componentes principais que interagem entre si, sensores, controladores e atuadores. Como o veículo deve ser capaz de interagir com o meio em que é inserido, é necessário o fornecimento de informações que promovam a ele sentidos sensoriais. Essas informações podem ser fornecidas por meio de sensores e traduzidas com estratégias lógicas que são capazes de promover uma interação com o ambiente com os atuadores (SOUSA PISSARDINI, WEI e FONSECA JÚNIOR, 2013).

A navegação de um veículo autônomo inclui em seu processo a combinação de diferentes etapas automatizadas que proporcionam a obtenção de dados do ambiente, a determinação da posição do veículo e a capacidade de evitar a colisão do veículo com outros elementos inclusos no ambiente (como pedestres, outros veículos, postes) (PISSARDINI, 2014). Na Figura 4, é possível observar a interação entre os principais componentes que promovem essa navegação.

Figura 4 – Estrutura de Veículos Autônomos



Fonte: Adaptado de (PISSARDINI, 2014)

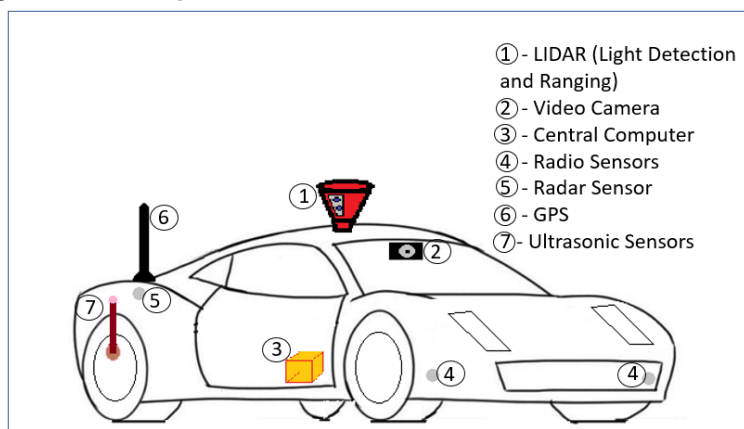
Os veículos autônomos possuem quatro principais perguntas que podem reger o seu funcionamento, a primeira delas diz respeito a localização do veículo.

Uma resposta para essa questão pode ser encontrada com as chamadas tecnologias de detecção e mapeamento, que por sua vez incluem sensores como IMU (*Inertial Measurement Unit*) e GPS (*Global Positioning System*). A segunda pergunta questiona o que há ao redor do veículo, essas informações podem ser processadas e obtidas por meio de sistemas de percepção e uso de sensores como LiDAR (*Light Detection And Ranging*), RADAR e câmeras. A terceira questão está ligada a tarefa de previsão do sistema autônomo, ou seja, o que irá acontecer a seguir, esses dados são processados e avaliados pelos sistemas de previsão (algoritmos e *softwares*). A última questão, envolve a tarefa de decisão do veículo, como ele fará a tarefa de deslocamento após processar as informações sobre localização, obstáculos e previsão. Para a resposta da última questão estão os sistemas de decisão conhecidos como planejamento de movimento e a utilização da inteligência artificial para auxiliar a tomada de decisão (SANTOS, 2020). Os principais componentes envolvidos no processo da direção autônoma serão discutidos nas próximas subseções desse capítulo.

2.3.1 Sensores

Os sensores presentes nos veículos autônomos são responsáveis pela promoção da visibilidade do ambiente (ALAM, 2018). Para que isso aconteça são necessários diferentes dispositivos e cada um possui uma finalidade específica, entre os sensores mais citados estão: LiDAR, radar, sensores ultrassônicos, GPS e câmeras. A Figura 5 mostra um veículo autônomo com seus principais sensores.

Figura 5 – Principais Sensores Presentes nos Veículos Autônomos

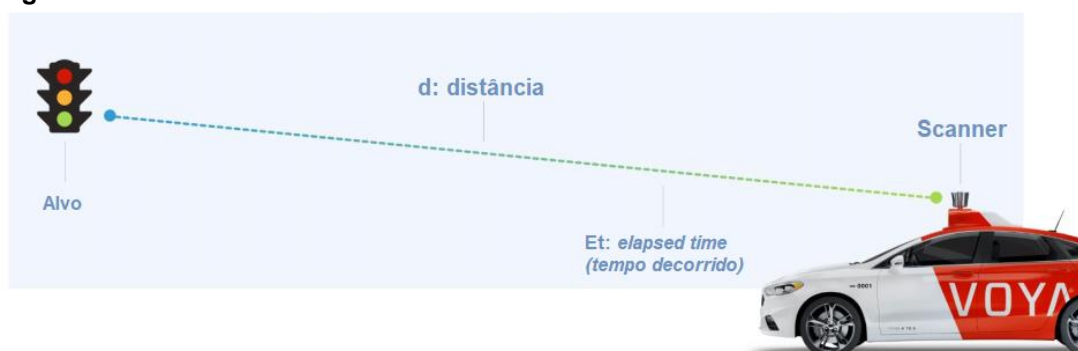


Fonte: Adaptado de (RAJASEKHAR e JASWAL, 2016)

O sensor LiDAR é classificado como um sensor remoto ativo, consegue captar uma vasta quantidade de dados do ambiente e dos objetos ou obstáculos existentes. Possui duas importantes características para um veículo autônomo, como visibilidade contínua em 360° e exatidão em medir distâncias, ou seja, consegue prever a distância de objetos ou obstáculos com uma exatidão de até 2 centímetros (CAMERON, 2017). O princípio de funcionamento é considerado simples, o instrumento faz o disparo de pulsos de luz *laser* muito rápidos em uma superfície (podendo ser até 150.000 pulsos por segundo). Internamente há um sensor que faz a medição de quanto tempo leva cada pulso, como essa ação é feita repetidamente o LiDAR desenha um mapa da superfície que está sendo medida. A

Figura 6 apresenta como é feito o cálculo do sensor interno presente no LiDAR, ou seja, com é obtida a distância final para mapeamento do ambiente.

Figura 6 - Sistema LiDAR



Fonte: Adaptado de (ODUNLADE, 2019)

$$d = \frac{E_t c}{2} \quad (1)$$

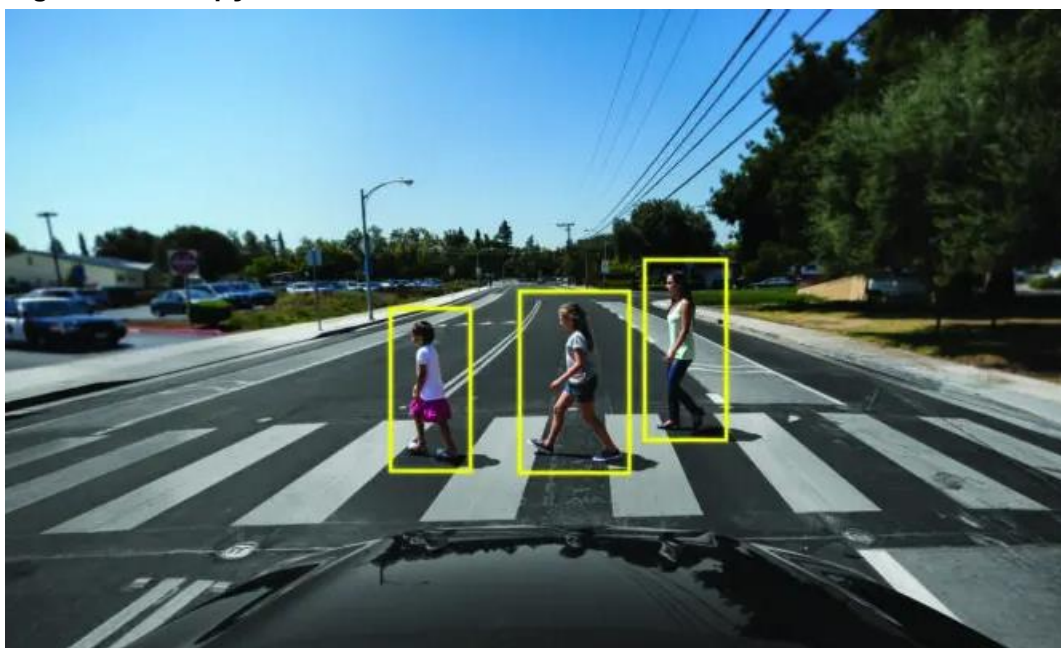
Em que a variável E_t (*elapsed time*) relaciona o tempo percorrido da luz laser e a variável c representa a velocidade da luz. O processo de repetir a emissão dos feixes de *laser* em milhões de vezes por segundo permite a criação de um mapa 3D do ambiente em que o veículo se encontra em tempo real (Velodyne Lidar, 2019).

A localização desse sensor normalmente está na parte superior do automóvel, promovendo a obtenção completa em três dimensões ao redor do veículo. Devido ao formato e a respectiva localização, existe uma discussão voltada para o visual do veículo. Pode-se notar que uma das maiores fabricantes de veículos semiautônomos não faz uso de tal pelo alto custo do sensor e pela estética final que pode apresentar no veículo (LiDAR UK, 2019).

Outro sensor presente na obtenção de dados sobre o ambiente em que o veículo se encontra é a câmera. O emprego das câmeras atualmente na maioria das áreas tecnológicas se dá por apresentar viabilidades quanto aos seus diversos tamanhos e preços acessíveis, se comparado a sensores como o LiDAR. Imagens geradas por câmeras nos veículos possuem a capacidade de melhorar o detalhamento do meio em que ele se encontra, auxiliando também ao fornecer dados como o ângulo do volante. O uso de câmeras tridimensionais está entrando em vigor em carros de luxo e, para que essa realidade exista, são utilizados sinais de entrada provenientes de um conjunto composto de quatro a seis câmeras instaladas ao redor do carro (RUDOLPH e VOELZKE, 2017).

Tecnicamente o uso de câmeras automotivas pode ser dividido em duas aplicações, a primeira aplicação é voltada para câmeras de ré e câmeras 360°, nessa aplicação são geradas imagens ao motorista na mídia presente no painel do carro. A segunda aplicação é chamada de câmeras frontais, essas câmeras utilizam algoritmos para detecção de objetos de forma automática, são capazes de classificá-los e determinar a distância entre eles. A título de exemplo, identificam pedestres como na Figura 7 em que os pedestres são identificados dentro dos retângulos amarelos.

Figura 7 – Percepção de Pedestres na Pista



Fonte: (BURKE, 2019)

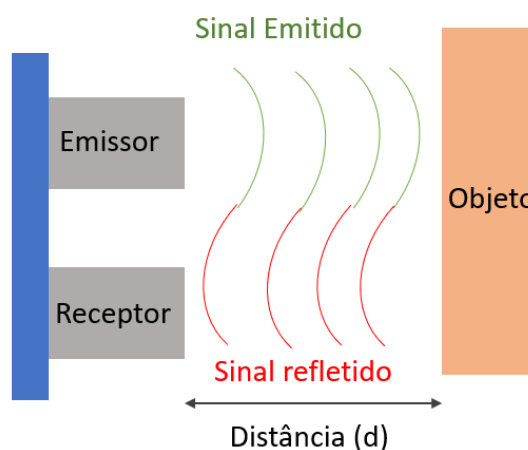
Ainda na segunda classificação existe uma divisão quanto ao alcance característico das câmeras, podendo ser de alcance médio ou de alto alcance. As

primeiras alertam ao motorista sobre presença de carros em um cruzamento, pedestres e até frenagem de emergência (CAMPBELL, OMAHONY, *et al.*, 2018). Já as câmeras de alto alcance possuem o objetivo de fazer o reconhecimento de sinais de trânsito, controle de distância do veículo baseado em imagem e até orientação do veículo no espaço. Nesses sistemas não existe a necessidade de especificação de cor, são utilizados apenas dados brutos de imagem que são captados pelo sensor da câmera.

Ou seja, são utilizados filtros RCCC (*Red Clear Clear Clear*) ou Vermelho Claro Claro Claro, os quais são capazes de fornecer uma intensidade luminosa maior que apresentada pelo filtro RGB (*Red Green Blue*) ou Vermelho Verde Azul. A diferença principal presente entre câmeras de alto alcance e médio alcance é o ângulo de abertura e campo de visão apresentado por elas (MEIRA, 2016). Embora hoje as câmeras promovam grande quantidade de dados para o veículo como identificação de formas, cores e até pessoas em um trabalho conjunto com algoritmos, o uso dela não exclui outros sensores. Ou seja, ainda assim se faz necessária a utilização de outros sensores para complementar as tarefas de um veículo autônomo.

Os sistemas com sensores ultrassônicos também estão presentes nos veículos autônomos, esses sensores fazem a detecção de obstáculos através de ondas de mecânicas de ultrassom, as quais possuem uma frequência de 20kHz, superior a audição humana (KATZWINKEL *et al.*, 2015). A medição dos obstáculos se dá através de pulsos ultrassônicos, ou seja, um sensor responsável pela emissão desses pulsos faz a medição do tempo entre um pulso emitido e um pulso refletido, a Figura 8 mostra a representação desses pulsos (XU, YAN, *et al.*, 2018).

Figura 8 - Cálculo da Distância pelo Sensor Ultrassônico



Fonte: Autoria Própria.

Então, o cálculo da distância medida até o obstáculo adjacente possui dependência do tempo de propagação, conforme a equação (1). Em que t_p e v_s relacionam os tempos de propagação dos pulsos ultrassônicos e a velocidade do som no ar respectivamente (XU *et al.*, 2018).

$$d = 0,5t_p v_s \quad (1)$$

A vantagem da utilização desses sensores está na significativa robustez apresentada, visto que são utilizados como sensores de estacionamento há algum tempo. Porém, podem apresentar dificuldades quanto as condições ambientais, por exemplo temperatura que pode ser responsável por uma perturbação na onda sonora, muitos desses problemas são resolvidos com uso de algoritmos de ajuste de leituras, levando em consideração a temperatura (PARK *et al.*, 2008).

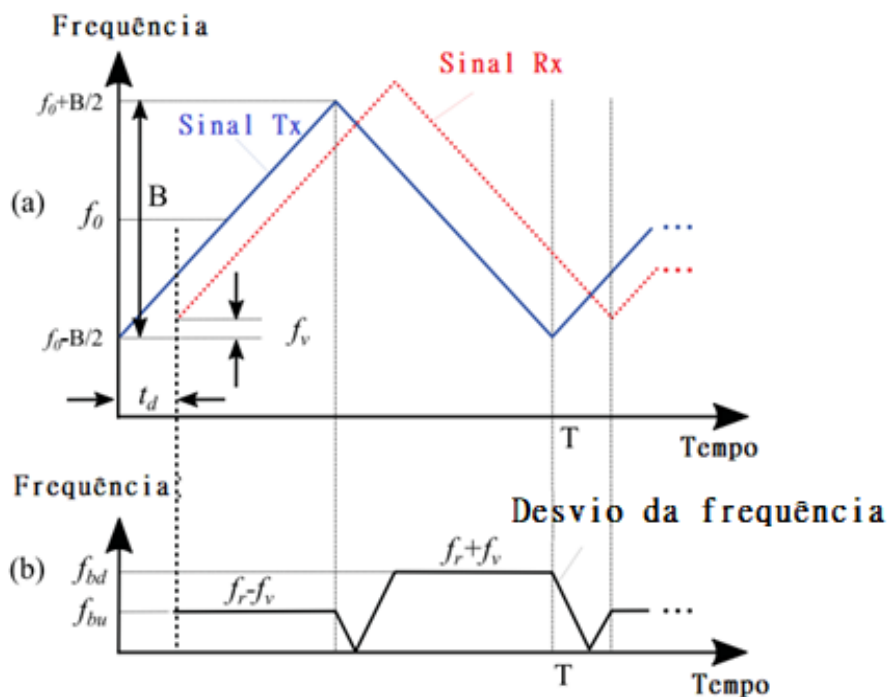
Outra tecnologia presente em VAs é a dos sensores RADAR, que utiliza ondas de rádio para medição de velocidade de objetos, ângulo e distância (DICKMANN, KLAPPSTEIN, *et al.*, 2016). A classificação pode variar conforme as faixas de operação, sendo de curta distância 10 metros a longa distância 100 metros. As aplicações em veículos atualmente abrangem controle de direção adaptativa, aviso de colisão frontal, identificação de pontos cegos e também assistência ao estacionar (AZAM *et al.*, 2019).

A evolução dos sensores aplicados em veículos autônomos tem-se mostrado diariamente. Nos radares esse ponto pode ser verificado a partir da classificação quanto a frequência de alcance apresentada. São comumente usadas duas larguras de banda, sendo 24 a 29 GHz para pequeno ou curto alcance e, 76 a 77 GHz para longo alcance, por exemplo, um sensor radar que apresente 77 GHz permite a detecção de objetos em até 200 metros (DAVIES, 2018).

Como sensor utiliza ondas eletromagnéticas, é importante citar um tipo especial de Radar chamado FMCW (*Frequency Modulation Continues Wave* – Onda Contínua com Frequência Modulada), diferente de um radar de onda contínua simples, esse radar pode fazer uma alteração em sua frequência durante uma medição, o sinal transmitido é modulado em fase (ou em frequência). A Figura 9 demonstra o princípio de funcionamento, os sinais modulados em frequência são enviados pelo transmissor que por sua vez serão refletidos pelo receptor. O desvio de

frequência resultante entre o sinal recebido (Sinal Rx) e o sinal transmitido (Sinal Tx) informa a distância e a velocidade dos objetos (TONG *et al.*, 2015).

Figura 9 – Modulação Clássica FMCW (a) Sinal Transmitido e sinal recebido; (b) Frequência Resultante



Fonte: Adaptado de (TONG, RENTER e FUJIMOTO, 2015)

Em que os símbolos são:

T: Período de modulação;

B: Largura de banda;

f_0 : frequência do sinal transmitido;

f_v : desvio de frequência Doppler;

f_r : mudança da faixa de frequência;

$f_{bu} = f_r - f_v$: frequência refletida para a rampa de subida;

$f_{bd} = f_r + f_v = A$: frequência refletida para a rampa descendente.

A obtenção dos valores f_{bu} e f_{bd} é feita por meio do cálculo das amostras do sinal da recepção. A distância e a velocidade do objeto são obtidos pela equação (2) e pela equação (3).

$$R = \left(\frac{cT}{4B} \right) \left(\frac{f_{bu} + f_{bd}}{2} \right) \quad (2)$$

$$v = \left(\frac{c}{2f_0} \right) \left(\frac{f_{bu} - f_{bd}}{2} \right) \quad (3)$$

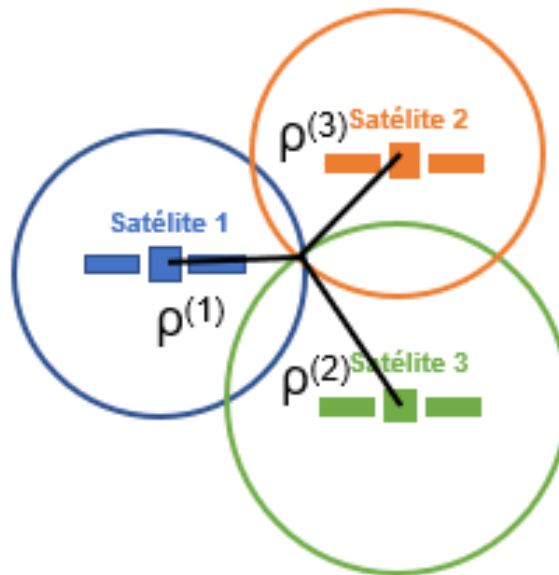
Em que R representa a distância do objeto, v a velocidade relativa do objeto e c a velocidade da luz. Entre as vantagens apresentadas no uso desse sensor para veículos autônomos destacam-se características de robustez, pois não apresenta alteração quanto a luz e indica bom funcionamento em condições de mau tempo.

A navegação confiável de um veículo autônomo exige uma estimativa de sua posição e orientação no mundo. A posição pode ser obtida com a combinação de diferentes sensores como o Sistema de Posicionamento Global ou GPS (*Global Positioning System*), e sistemas de unidades inerciais, como a IMU (*Inertial Measurement Unit*).

Segundo Parakkal e Variyar (2017) o GPS consiste em um sistema de radio-navegação por satélite utilizado para determinar uma localização. A sigla é utilizada como exemplo para várias constelações existentes de satélites, a constelação de GPS possui de 24 a 32 satélites em seis planos orbitais. O projeto da constelação é desenvolvido de forma que pelo menos quatro satélites encontrem-se visíveis em qualquer ponto da superfície da Terra. O princípio de funcionamento envolve a hora de chegada do sinal, um receptor calcula a distância para cada satélite fazendo uma comparação entre o relógio interno e a hora da transmissão, a diferença de tempo obtida é convertida em uma distância utilizando a velocidade da luz.

Para o cálculo de uma posição em três dimensões as equações exigem que existam pelo menos quatro satélites visíveis, nos casos em que apenas uma posição em duas dimensões é necessária, com três satélites visíveis é possível obter o resultado. O cálculo da posição em duas dimensões é chamado de trilateração, demonstrado na Figura 10. Um receptor GPS faz a medição da pseudo-faixa de cada satélite utilizando o modelo de medição da equação (4) em que são considerados os erros no relógio do receptor, atrasos na propagação atmosférica e ruídos de medição. Cada pseudo-faixa promove a definição de um círculo em duas dimensões ou uma esfera em três dimensões (WASLANDER e KELLY, 2018d).

Figura 10 - Trilateração em 2D



Fonte: Adaptado de (WASLANDER e KELLY, 2018d)

$$\rho^{(i)} = c(t_r - t_s) = \sqrt{(\rho^{(i)} - r)^T ((\rho^{(i)} - r) + c\Delta t_r + c\Delta t_a^{(i)} + \eta^{(i)})} \quad (4)$$

Em que:

r posição (3D) do receptor

$\rho^{(i)}$ posição do satélite i

Δt_r erro do relógio receptor

$\Delta t_a^{(i)}$ atraso da propagação atmosférica

η ruído da medição

c velocidade da luz

t_r, t_s hora enviada, hora recebida

O sistema GPS pode manter um alto nível de precisão de posicionamento para um veículo autônomo, exceto que o receptor GPS apresente uma falha ou que não tenha o controle de pelo menos quatro satélites. Para uma estimativa de localização mais exata de um veículo autônomo é importante citar o uso do sistema GPS juntamente com uma IMU (Unidade de Medida Inercial), esses dois sistemas são considerados complementares.

Uma Unidade de Medida Inercial (IMU – *Inertial Measurement Unit*) trata-se de um dispositivo com a capacidade de medir força, velocidade angular e campo

magnético. Normalmente são formados por três acelerômetros e três giroscópios, (algumas vezes incluindo magnetômetros e bússolas), com o objetivo de fazer o rastreamento do movimento de um corpo rígido, contendo um par orientado presente em cada um dos eixos x, y e z ortogonais (WASLANDER e KELLY, 2018a).

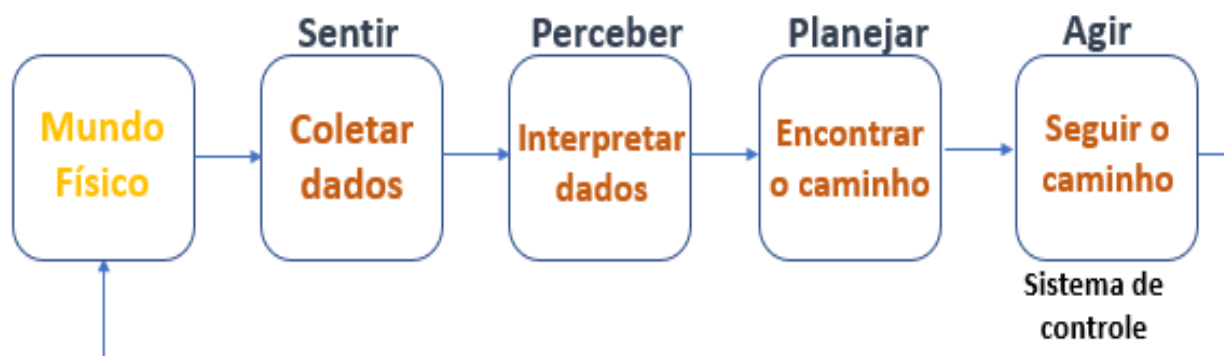
O uso desse dispositivo permite a geração de dados que relacionam aceleração e parâmetros de rotação, ou seja, casos em que o veículo esteja em inclinação, rotação, nos três eixos do veículo. Exigem pouca energia para funcionamento, porém podem acarretar resultados com muito ruído. As IMUs mais robustas fazem uso de componentes com maior complexidade os quais promovem uma melhor precisão dos resultados, pois seus sistemas de calibração fazem a remoção dos ruídos (HORTON, 2018).

2.3.2 Fusão de Sensores

Apresentação dos principais sensores inclusos nos veículos autônomos resulta em uma parte integrante do sistema desses veículos denominada fusão de sensores. A técnica consiste em combinar dois ou mais sensores de forma que sejam gerados resultados para o sistema, como um resultado mais preciso e mais confiável. Douglas (2019) descreve a importância da fusão de sensores sob uma análise das quatro áreas principais em que os veículos autônomos estão divididos: sentir, perceber, planejar e agir.

O sentir está relacionado coleta de dados do local em que o veículo se encontra, que pode ser executada com sensores como RADAR, LiDAR e câmeras. Como a coleta de dados não é suficiente, o sistema precisa interpretá-los e modificá-los para que a autonomia possa ocorrer (NASCIMENTO, SANTOS, *et al.*, 2019). Dessa forma acontece a etapa da percepção, por exemplo os dados obtidos de uma câmera, pixels obtidos pela imagem transformam os objetos como placas, pedestres e faixas da estrada em informações. Então a fase de planejamento inicia, o sistema do veículo encontra o caminho livre de obstáculos para seguir, calcula-o e segue-o. O agir inclui as principais tarefas do sistema de controle, a Figura 11 mostra como essa análise pode ser feita.

Figura 11 - Sistemas Autônomos



Fonte: Adaptado de (DOUGLAS,2019)

O processo da fusão de sensores consiste na realização de medição, combinação das informações obtidas e formulação em modelos matemáticos. Assim promovendo um entendimento melhor do meio em que o sistema autônomo se encontra, podendo ser usado para o planejamento e ação do deslocamento (DOUGLAS,2019).

Existem trabalhos apresentados que mostram técnicas utilizadas na fusão de sensores para localização do veículo em ambientes abertos ou fechados, e algumas metodologias utilizadas para detecção de objetos. Lin *et al.* (2018) apresentam a utilização das matrizes de transformação adquiridas de dados de uma câmera e das nuvens de pontos de um LiDAR para fazer o mapeamento de um ambiente. Os resultados obtidos no trabalho mostram que através de uma câmera de baixo custo e um LiDAR 2D é possível obter de forma eficiente uma modelagem 3D.

De acordo com Jeon *et al.* (2016) o uso do GPS (Sistema de Posicionamento Global), uma IMU (Unidade de Medição Inercial) juntamente com cálculo de posição (DR – *Dead Reckoning*) resultando em um sistema de localização, mostrando através dos testes feitos em diferentes cenários urbanos que não é necessária muita computação e que o erro de posição que pode ser acumulado pode ter seu valor diminuído através do cálculo DR. Com o objetivo de fundir sensores de odometria e sensores magnéticos para um sistema de localização, Surrecio, Nunes e Araujo (2005) mostram um comparativo de resultados utilizando dois diferentes filtros de Kalman.

A fusão de sensores pode ser abordada também como uma forma de melhorar a área de abrangência que o veículo autônomo necessita (DOUGLAS, 2019). Pode-se imaginar o uso de sensores ultrassônicos de baixo alcance para auxiliar um veículo

em uma manobra de estacionamento, esse sensor oferece dados de alguns metros e um pequeno campo de visão. Para os casos em que o sistema veicular necessita da situação do veículo em 360°, outros sensores precisam ser adicionados e as medições obtidas precisam ser fundidas, dessa forma produzindo um campo de visão maior. Diferentes maneiras de fusão de sensores são apresentadas para o uso de veículos autônomos, mesmo que sejam utilizados diferentes algoritmos e sensores para as aplicações o objetivo geral é o mesmo, utilizar diferentes fontes de dados para a melhoria da confiabilidade de medição e estimação de estados.

2.3.3 Controladores

A presença de todos os sensores citados anteriormente no veículo e com a quantidade de dados processados e interpretados, surge a necessidade de controladores para que tal tarefa seja executada. Atualmente com a pesquisa avançada e o interesse de mercado presente em VAs (veículos autônomos) é possível encontrar diferentes tipos de controle apresentados para o gerenciamento das tarefas.

Alguns dos controladores presentes nos sistemas autônomos veiculares podem ser denominados sistemas inteligentes. Capazes de realizar pesquisas e otimizar o conjunto das tarefas com as habilidades advindas do aprendizado. É possível exemplificar esses sistemas com os algoritmos de controle que atuam nos veículos autônomos. Técnicas de controle têm sido apresentadas em pesquisas e artigos científicos mostrando o controle sobre ângulos de direção, aceleração e desaceleração, a Tabela 1 mostra uma breve comparação dos trabalhos apresentados.

A mobilidade do veículo exige um controle que assuma um caminho específico e controle o veículo para segui-lo. Normalmente o caminho é caracterizado por um conjunto de pontos ligados por linhas retas ou por curvas conhecidas como “*splines*”. Um controlador amplamente utilizado nesse quesito é o chamado Controle de Perseguição Pura (*Pure Pursuit*), considerado robusto para terrenos que não apresentam condições favoráveis e também sob certos quesitos ambientais (LATOMBE, 2012).

Tabela 1 - Trabalhos Realizados Utilizando Controladores e Sensores para Veículos Autônomos

Autor	Sensores	Controlador
Bergerman	Gnss e Lasers	Proporcional Integral Derivativo (PID)
Lima	Radar, DGPS, IMU	Modelo de Controle Preditivo (MPCC), Controle de Perseguição Pura (PPC)
Mirisola	Encoders, GNSS	Proporcional Derivativo (PD)
Moreira	GNSS, IMU	Preditivo, <i>Fuzzy</i>
Utsumo	Câmera RGB	Proporcional Integral Derivativo (PID), <i>Non-linear Model Predictive Control</i> (NMPC)

Fonte: Adaptado de (PINTO e PORTO, 2018)

Para a aprendizagem e a capacidade de tomar decisões, algoritmos envolvendo inteligência artificial são utilizados. No quesito localização para veículos autônomos tecnologias como SLAM (Mapeamento e Localização Simultânea – *Simultaneous Localization and Mapping*) e algoritmos A* são fortemente apresentados (VELOSO, 2019; LIKHACHEV, GORDON e THRUN, 2003).

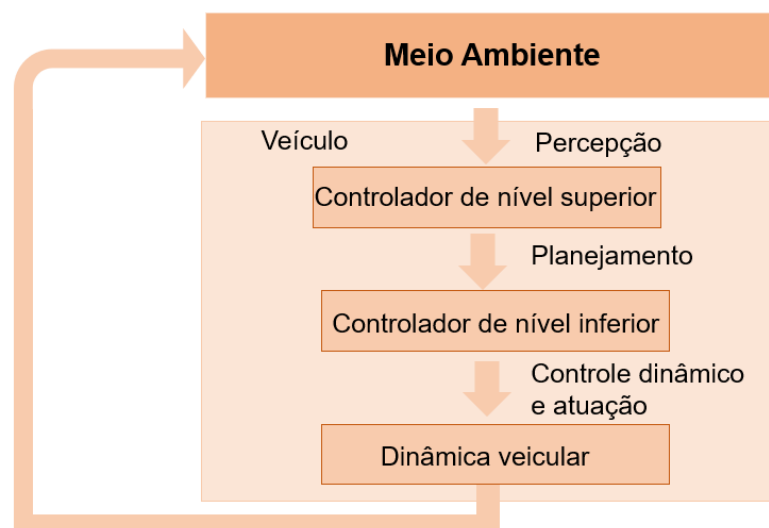
O SLAM trata-se de um conjunto de algoritmos que se concentram em resolver problemas simultâneos de localização e mapeamento. Pode ser apresentado de diversas maneiras, sua tarefa basicamente é a partir dos dados do sensor avaliar de um obstáculo apresentado pode estar lá mesmo o não, por se tratar de um método probabilístico, seu resultado mostra as chances de haver algo no caminho. Em sistemas de localização diferentes técnicas são apresentadas como por exemplo filtros de Kalman. Também sistemas que envolvem SLAM visuais, os quais fazem uso de um conjunto de pontos oferecidos por quadros sucessivos de câmeras e esses pontos fazem uma triangulação 3D da posição do veículo (GUAPACHA, 2017).

O trabalho de LI *et al.* (2016) apresenta uma estrutura hierárquica, apresentada na Figura 12, utilizada nos projetos de sistemas de controle de VAs. Nessa estrutura um controlador externo faz as trajetórias de referência para um controle interno e este gera as entradas de dados para os ângulos de direção, bem como aceleração e frenagem

Outras diferentes abordagens de controle podem ser encontradas, as quais incluem desde árvores de decisão, até processos de utilizando o método de Markov. Com o objetivo de oferecer aprendizagem para o controle longitudinal de um VA, é proposto um modelo de direção que gera entradas de aceleração as quais são utilizadas como referência por um modelo de controle preditivo (LEFEVRE, CARVALHO e BORRELLI, 2015). Os dados utilizados no modelo são reais, quando

os problemas de otimização são resolvidos, o controlador sobreposto no veículo garante que alguns critérios de segurança pré-estabelecidos sejam atendidos.

Figura 12– Estrutura Hierárquica Utilizada em Sistemas de Controle de VAS



Fonte: Adaptado de (LI *et al.*, 2016)

2.3.4 Atuadores

Em uma escala evolutiva para compreensão dos VAs imagina-se como última tarefa a executada pelos atuadores, pois são eles os responsáveis pela visível interação dos VAs com o meio. Entre os três principais atuadores desses veículos está a aceleração ou atuação do acelerador, direção ou atuação da direção e frenagem ou atuação do freio.

A maioria dos veículos contém os atuadores essenciais para o controle da automação, por meio de um controle eletrônico é possível obter a atuação do acelerador. Os sistemas de direção hidráulica tem sido substituídos por sistemas de direção elétrica, promovendo mais segurança. E nos sistemas de frenagem pode ser citado o controle eletrônico de estabilidade (ECS – *Electronic Stability Control*).

Segundo Leiss (2018) os sistemas de atuação que permitem atuação de um veículo autônomo moderno estão em vigor há anos. A inclusão dos sensores juntamente com os sistemas de controle permite que o veículo autônomo obtenha a localização, entenda o ambiente em que se encontra e decida como deve ocorrer atuação para o deslocamento.

2.4 CONCLUSÃO

Embora alguns desafios existam para que a regulamentação dos veículos autônomos entre em vigor, grandes empresas estão investindo em pesquisa e desenvolvimento nesses veículos. Diferentes competições e testes são encontrados no histórico dos veículos autônomos comprovando a funcionalidade dos mesmos. E a integração do sistema como um todo incluindo, sensores, atuadores e controladores, também apresenta grandes atualizações. Sensores estão sendo incrementados e melhorados, e os sistemas de atuação são pesquisados em aplicações específicas.

3 ESTRATÉGIAS DO PLANEJAMENTO DE MOVIMENTO

Como foi possível entender no capítulo anterior, uma estimação da localização e quais as próximas tarefas o veículo autônomo deverá executar para o deslocamento acontece por meio da união de dados de vários sensores que compõe o veículo. Além dos desafios impostos pela fusão de sensores, nesse capítulo será voltado um olhar para o planejamento de movimento. Um VA precisa entender o ambiente em que se encontra e processar informações de localização. Tecnologias desenvolvidas para a estimação da localização tem como objetivo a realização de três funções principais de um VA: percepção, planejamento e controle (LUO *et al.*, 2019).

O processo de navegação de um veículo terrestre pode ser interpretado como o uso de um veículo terrestre motorizado por um ser humano, denominado motorista ou condutor, com o objetivo de sair de um local e atingir outro de seu interesse (PISSARDINI, 2014). Durante essa navegação veicular, vários quesitos devem avaliados pelo condutor, como o caminho mais seguro, ou o caminho mais rápido de forma que o objetivo do deslocamento seja alcançando mantendo a integridade do veículo e do condutor (GOCKLEY, FORLIZZI e SIMMONS, 2007).

Cenários envolvendo pedestres e até motoristas imprudentes, mostram que a tarefa de condução autônoma é bastante desafiadora, pois além da infração das regras impostas, podem gerar confusão quanto as ações de controle que o veículo deve tomar. Por isso autores indicam a divisão da tarefa planejamento de movimento em níveis de problemas de otimização, abrindo possibilidades de adaptação de entradas e saídas para cada problema e o nível de abstração necessário para a realização do planejamento de movimento em tempo real (WASLANDER e KELLY, 2018b).

A atividade do veículo autônomo nesses cenários pode ser imaginada como um arranjo simplificado de ações ou manobras de alto nível (DOLGOV, THRUN, *et al.*, 2010). O ser humano executa a navegação veicular por meio da combinação de um conjunto de condutas mentais e físicas fundamentadas na observação e na aptidão de modelar o ambiente de condução (PISSARDINI, 2014). A navegação robótica veicular pode ser apresentada por um modelo com variáveis de estado em que um estado representa o comportamento das variáveis específicas do veículo ou dos elementos que compõe o ambiente em um intervalo de tempo (DURRANT-WHYTE,

2001). Um conjunto de processos permite organizar a navegação autônoma em subprocessos:

- **Aquisição da missão (estado a ser atingido):** o estado final que o veículo deve alcançar com o movimento. Considerado o primeiro evento do processo de navegação, pode ser sugerido por um agente humano, por um sistema ou pode ser resultado do sistema autônomo a um comportamento do ambiente.
- **Percepção ou Sensoriamento (estado atual do ambiente):** obtenção de características do ambiente por meio de sensores, filtrando-as e modelando-as em representações necessárias com objetivo de oferecer informações aos outros subprocessos da navegação.
- **Posicionamento ou Localização (estado atual do veículo):** definição da habilidade que o veículo possui na obtenção de dados do estado em se encontra e comparação ao ambiente em que está percorrendo.
- **Planejamento de rota (estados intermediários a serem alcançados):** estimativa de estados e ações adequadas e com menor custo operacional por meio de dados e algoritmos, que promovem ao veículo a realização da tarefa de deslocamento evitando colisões e permitindo o planejamento de ideais de locomoção;
- **Mapeamento (estados do ambiente armazenados no veículo):** armazenamento dos dados, obtidos pela percepção, em formato interno adequado que pode ser utilizado em navegações futuras, principalmente pelo processo planejamento de rota. Considerado um processo opcional, pode ser substituído pela aquisição de dados do ambiente sem o mapeamento.
- **Estratégia de movimento (transição entre estados):** objetiva a geração das ações que devem ser realizadas para o movimento do veículo. O planejamento de rotas faz a estimativa do estado que o veículo deve assumir bem como as ações associadas que devem ser realizadas, já a estratégia de movimento gera as ações que serão utilizadas para a locomoção, repassadas a um sistema como capacidade de traduzi-las para que os atuadores operem ações observáveis no mundo real.

Esses processos são caracterizados por um ciclo de processamento que faz o sensoriamento do ambiente, a interpretação dos dados obtidos, a geração das

ações sobre os dados e a execução da ação sobre o ambiente. Para a aplicação em um veículo autônomo, o ciclo do processamento deve ser feito em um limite de tempo aceitável, para que exista a interação adequada com o ambiente da mesma forma que um condutor humano faria.

3.1 PLANEJAMENTO DE MOVIMENTO E OS MODELOS DE ESTADOS

O planejamento de movimento de um veículo autônomo necessita de uma modelagem de estados, que definirá como o veículo realizará o processamento e o desenvolvimento dos processos de navegação e tomada de decisão. A escolha dos modelos de estados possui dependência com os tipos de dados que serão manipulados e os algoritmos que serão utilizados.

Há dois tipos de modelo de estados que podem ser avaliados pelos veículos autônomos terrestres. O primeiro é denominado Modelo de Estados do Veículo, descreve o comportamento do veículo no espaço e no tempo para compreender os processos de aquisição de missão, posicionamento, planejamento de rotas e estratégia de movimento. O segundo é o Modelo de Estados do Ambiente, que descreve o comportamento das informações do ambiente no espaço e no tempo para os processos de percepção, mapeamento, planejamento de rotas e estratégia de movimento (KATRAKAZAS, QUDDUS, *et al.*, 2015).

3.1.1 Modelo de Estado do Veículo

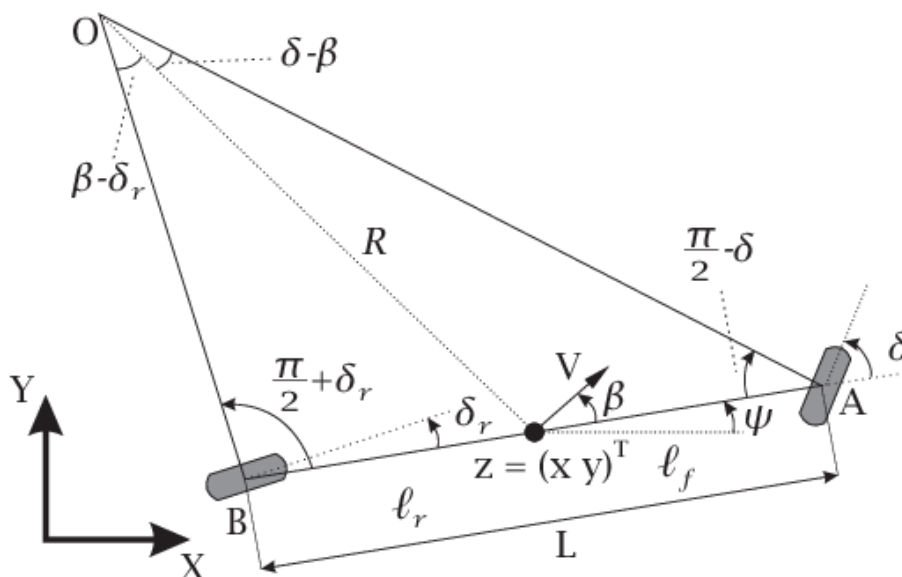
Cada veículo apresenta uma arquitetura específica com tecnologias, conceitos e propostas de acordo com seu desenvolvedor ou fabricante, porém para o processo de navegação o veículo como pode ser caracterizado como um corpo rígido no espaço modelado em função de variáveis que descrevem seu estado no espaço-tempo. A modelagem do veículo pode ser realizada de duas maneiras, a primeira é denominada modelagem cinemática, considerada mais genérica, caracteriza o veículo como um ponto, desconsiderando forças de perturbação que possam intervir em sua estrutura ou em suas ações (PISSARDINI, 2014). A modelagem cinemática do veículo descrita na próxima subseção será o modelo bicicleta da cinemática. A segunda modelagem é a modelagem dinâmica, considerada para aplicações mais

específicas que objetivam entender os componentes do ambiente e os impactos que podem apresentar sobre o veículo, considerando as forças e perturbações do ambiente. (SIEGWART, NOURBAKHS e SCARAMUZZA, 2011).

3.1.1.1 Modelo bicicleta da cinemática (*Kinematic Bicycle Model*)

Esse modelo é muito usado porque bicicletas apresentam uma série de valores angulares de direção de forma aceitável a veículos (WASLANDER e KELLY, 2018c). Para entendimento matemático de tal modelo, considera-se a Figura 13. As rodas dianteiras do automóvel são representadas em apenas uma roda (ponto A), de maneira análoga para as rodas traseiras (ponto B). São adotados ângulos de direção, os quais são representados por δ_r e δ_f , em que o primeiro representa os ângulos da roda traseira (*rear*) e o segundo, os ângulos da roda dianteira (*front*). Como apenas o ângulo dianteiro apresentará ângulos variados de direção, os ângulos de direção traseira são nulos, ou seja, adota-se $\delta_r = 0$.

Figura 13 – Modelo Bicicleta da Cinemática



Fonte: (RAJAMANI, 2011)

No ponto C está localizado o centro de gravidade do veículo, e as distâncias entre esse ponto e os pontos A e B são representadas por l_r (distância da roda traseira até o centro de gravidade do veículo) e l_f (distância da roda dianteira até o

centro de gravidade do veículo), com isso a distância entre eixos do veículo é representada por:

$$L = \ell_r + \ell_f \quad (5)$$

Rajamani (2011) apresenta um estudo detalhado matematicamente a respeito desse modelo. Inicialmente considera-se que o veículo está em movimento plano, e esse movimento é descrito por três coordenadas (X , Y e ψ), ressaltando que (X, Y) são as coordenadas inerciais que referenciam a localização do centro de gravidade do veículo (C), ψ representa a orientação, o ponto V retrata a velocidade do centro de gravidade e β é o ângulo da velocidade atual do centro de gravidade com relação ao eixo longitudinal do veículo.

Aplica-se a regra de senos, para os dois triângulos internos que compõe a figura do modelo, sendo os triângulos OCA e OCB , obtendo as equações (6) e (7).

$$\frac{\text{sen}(\delta_f - \beta)}{\ell_f} = \frac{\text{sen}(\frac{\pi}{2} - \delta_f)}{R} \quad (6)$$

$$\frac{\text{sen}(\beta - \delta_r)}{\ell_r} = \frac{\text{sen}(\frac{\pi}{2} - \delta_r)}{R} \quad (7)$$

Os termos $\text{sen}(\beta - \delta_r)$ e $\text{sen}(\delta_f - \beta)$ podem ser substituídos pela relação de subtração e adição dos arcos, para melhor desenvolvimento de cálculo.

$$\frac{\text{sen}(\delta_f) \cos(\delta_r) - \text{sen}(\beta) \cos(\delta_f)}{\ell_f} = \frac{\text{sen}(\frac{\pi}{2}) \cos(\delta_f) - \text{sen}(\delta_f) \cos(\frac{\pi}{2})}{R} \quad (8)$$

$$\frac{\text{sen}(\beta) \cos(\delta_r) - \text{sen}(\delta_r) \cos(\beta)}{\ell_r} = \frac{\text{sen}(\frac{\pi}{2}) \cos(\delta_r) - \text{sen}(\delta_r) \cos(\frac{\pi}{2})}{R} \quad (9)$$

Simplificando os termos, resulta nas equações (10) e (11).

$$\frac{\text{sen}(\delta_f) \cos(\beta) - \text{sen}(\beta) \cos(\delta_f)}{\ell_f} = \frac{\cos(\delta_f)}{R} \quad (10)$$

$$\frac{\text{sen}(\beta) \cos(\delta_r) - \text{sen}(\delta_r) \cos(\beta)}{\ell_r} = \frac{\cos(\delta_r)}{R} \quad (11)$$

Então na equação (10) multiplica-se ambos os lados por $\frac{\ell_f}{\cos(\delta_f)}$ e na equação (11) por $\frac{\ell_r}{\cos(\delta_r)}$, obtendo as equações (12) e (13).

$$\text{tg}(\delta_f) \cos(\beta) - \text{sen}(\beta) = \frac{\ell_f}{R} \quad (12)$$

$$\text{sen}(\beta) - \text{tg}(\delta_r) \cos(\beta) = \frac{\ell_r}{R} \quad (13)$$

O resultado da soma entre as equações (12) e (13) é:

$$[\text{tg}(\delta_f) - \text{tg}(\delta_r)] \cos(\beta) = \frac{\ell_r + \ell_f}{R} \quad (14)$$

Assume-se que o raio da trajetória (R) pode apresentar mudanças conforme a velocidade que o veículo apresenta, então a taxa de variação da orientação do veículo é igualada a sua velocidade angular. Desde que a velocidade seja uma relação de V por R, é adotada a seguinte relação.

$$\psi = \frac{V}{R} \quad (15)$$

Mediante a equação (15) a equação obtida em (14) é reescrita em (16).

$$\psi = \frac{V \cos(\beta)}{\ell_f + \ell_r} [\text{tg}(\delta_f) - \text{tg}(\delta_r)] \quad (16)$$

Com base nas variáveis que formam o sistema, as equações gerais de tempo contínuo não lineares que compõe um modelo bicicleta, em um sistema inercial são:

$$\dot{X} = v \cos(\psi + \beta) \quad (17)$$

$$\dot{Y} = v \operatorname{sen}(\psi + \beta) \quad (18)$$

$$\psi = \frac{V \cos(\beta)}{\ell_f + \ell_r} (\operatorname{tg}(\delta_f) - \operatorname{tg}(\delta_r)) \quad (19)$$

O uso do modelo cinemático de bicicleta é bastante comum se comparado com outros modelos. A identificação do sistema é dada mais facilmente considerando que os parâmetros que precisam ser identificados são ℓ_r (distância da roda traseira até o centro de gravidade do veículo) e ℓ_f (distância da roda dianteira até o centro de gravidade do veículo), conseqüentemente exige menos entradas no controlador de planejamento de movimento que será utilizado e flexibiliza o uso para diferentes veículos com distâncias entre eixos distintas (KONG, PFEIFFER, *et al.*, 2015).

3.1.2 Modelos de Estado do Ambiente

Por sua vez os Modelos de Estado do Ambiente são importantes porque o ambiente trata-se do espaço de navegação que o veículo ocupa. E o veículo autônomo terrestre necessita da capacidade de extração dos dados do ambiente para que aconteça a interação adequada com a situação de trânsito em que está incluído (DUDEK e JENKIN, 2010).

O deslocamento de um veículo autônomo por um caminho exige a representação do ambiente, isto é, o espaço físico em que o veículo se encontra deve ser modificado em um espaço de estado ou em uma configuração (KATRAKAZAS, QUDDUS, *et al.*, 2015). A integração de dados obtidos por sensores, comunicação com outros elementos do ambiente e informações anexadas no sistema previamente, permite a construção de um modelo de estado de ambiente. O uso de alguns algoritmos permite a geração de mapas computacionais em coordenadas contínuas por meio dos limites da estrada e as posições dos obstáculos. (JEON, COWLAGI, *et al.*, 2013). Mapas computacionais, que podem ser divididos em dois grupos: mapas métricos e mapas topológicos.

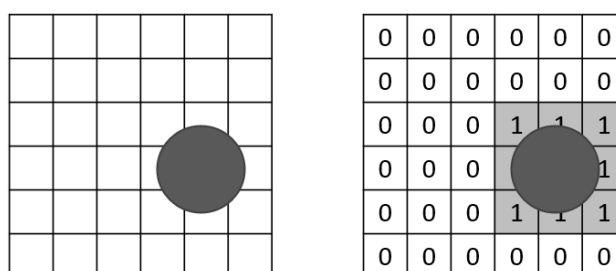
Os mapas métricos oferecem a descrição do ambiente e os elementos que o compõe por meio de propriedades físicas. Já os mapas topológicos fazem uma

representação do ambiente por meio dos elementos generalizados, normalmente resultando em dados na forma de grafos. Segundo Buschka e Saffioti (2004), os mapas métricos apresentam um melhor detalhamento que os mapas topológicos, porém podem necessitar de maior custo computacional e armazenamento para que o ambiente seja descrito e caracterizado de forma exata. A segmentação dos mapas métricos e mapas topológicos pode ser feita em quatro grupos de mapas mais básicos: mapas em grade, mapas geométricos, mapas sensoriais e mapas topológicos (PISSARDINI, 2014).

3.1.2.1 Mapas em Grade

Considerado um modelo e mapa computacional métrico, o mapa em grade caracteriza-se pela divisão do ambiente em uma matriz de duas dimensões com células homogêneas, cada célula possui um valor que determina se está ocupada por algum elemento ou se está disponível para ocupação (FILLIAT e MEYER, 2003). Podem ser chamados de grade de ocupação, mapa e probabilidade, entre outros. A Figura 14 apresenta um exemplo de mapa de grade, em que as células ocupadas pelo círculo estão convencionadas pelo número 1 e as células livres número 0.

Figura 14 - Exemplo de Mapa de Grade



Fonte: Autoria Própria.

O trabalho de Pissardini (2014) detalha como a determinação dos espaços ocupados pode ser feita:

- **Dados de sensores:** com os dados fornecidos pelos sensores, o veículo obtém as posições dos elementos presentes no ambiente vinculando cada posição a uma célula da grade. Células livres não possuem elementos e células vinculadas a algum elemento estão ocupadas. O uso dessa técnica apresenta como vantagem a irrelevância da extração de características dos

elementos presentes no ambiente para a composição do mapa. Porém como a grade é criada com base no alcance dos sensores presentes no veículo torna-se ineficiente o uso para planejamentos a médio e longo alcance, pois o veículo não consegue fazer o reconhecimento das células ocupadas fora do sensoriamento (FILLIAT e MEYER, 2003).

- Dados *a priori*: a construção do mapa em grade acontece antes da navegação com o uso de dados advindos de diferentes fontes, assim o veículo conhece as regiões do espaço que estão livres para locomoção e ocupação. O uso dessa técnica apresenta bom comportamento em ambientes com pequenas escalas e com característica estática, dessa forma sendo de difícil utilização em ambientes grandes e dinâmicos (cidades).
- Híbrida: ocorre a construção da grade antes da navegação com a estimativa do que o veículo encontrará no ambiente. A confirmação dessa estimativa ocorre por meio dos dados advindos dos sensores, dessa forma determinando se as células serão ocupadas ou não. O uso dessa técnica apresenta uma minimização das desvantagens apresentadas a técnica que utiliza apenas os dados provenientes dos sensores, porém mantém as desvantagens apresentadas na técnica dos dados *a priori*.

Os mapas em grade portanto, possuem eficácia nos detalhes para mapeamento do ambiente, porém apresentam maior demanda em custo de armazenamento e processamento computacional para áreas com maior extensão.

3.1.2.2 Mapas Geométricos

Os mapas geométricos são mapas caracterizados em duas ou três dimensões, que fazem uso de formas geométricas como linhas, malhas, pontos, triângulos, poliedros para a segmentação do ambiente (DUDEK e JENKIN, 2010). Esses mapas possuem um tratamento computacional utilizando parâmetros numéricos, facilitando as manipulações geométricas e reduzindo o custo de processamento e armazenamento de dados. A adoção de uma forma geométrica específica é feita considerando a necessidade da operação que será realizada no espaço mapeado (CHOSSET, HUTCHINSON, *et al.*, 2005).

3.1.2.3 Mapas Sensoriais

Caracterizam-se pela obtenção de dados a partir dos sensores, por exemplo dados brutos (nuvens de pontos) provenientes de um sensor *laser*, proporcionando a representação do ambiente em três dimensões. São capazes de fornecer grandes detalhes do ambiente o que resulta em grande custo de processamento para que exista uma manipulação dos detalhes eficiente (DUDEK e JENKIN, 2010).

3.1.2.4 Mapas Topológicos

Os mapas topológicos fazem a descrição da relação geométrica entre os elementos do ambiente que são escolhidos conforme a necessidade de desenvolvimento da aplicação (regiões de interesse, marcações do ambiente). Esses elementos são relacionados e modelados na forma de grafos, em que os vértices caracterizam as regiões de interesse e as arestas relacionam os nós (THRUN, 2002). Apresentam eficiência na modelagem de eventos que não exigem detalhes na construção, como trajetórias e possuem menor armazenamento computacional se comparado com os mapas métricos e também menor custo de processamento.

3.2 TÉCNICAS DE PLANEJAMENTO DE MOVIMENTO

No veículo autônomo a tarefa de encontrar um caminho ou uma trajetória livre de colisões e confortável para o passageiro é de responsabilidade do planejamento de movimento, essa tarefa possui correspondência com a resolução de um problema de movimento. O planejador de movimento muitas vezes pode ser responsável por minimizar algumas funções objetivas da direção autônoma, por exemplo nos casos em que esse planejador absorve informações sobre obstáculos durante o caminho gerando uma trajetória livre de colisões e satisfazendo restrições dinâmicas e cinemáticas do movimento veicular (PADEN, ČÁP, *et al.*, 2016; Realtime Robotics, 2019).

O artigo de Katrakazas *et al* (2015), apresenta conceitos importantes juntamente com sua definição para o entendimento do processo de planejamento de movimento:

- **Caminho:** um traço que o veículo deverá seguir para que atinja o ponto final livre da colisão de obstáculos.
- **Planejamento de caminho:** é a tarefa de encontrar um caminho geométrico de forma que o deslocamento seja viável em conforto, tempo e economia de combustível.
- **Manobra:** operação inclusa entre as tarefas de alto nível de movimento do veículo, envolvendo posição e velocidade. Para exemplificar manobras, podem ser citadas as ações de seguir frente, retornar, ultrapassar.
- **Planejamento de manobra:** objetiva-se a melhor tomada de decisão em alto nível, considerando o trajeto determinado inicialmente no planejamento de caminho.
- **Trajetória:** representação de uma sequência de estados em que o veículo é submetido, com parametrização pelo tempo e pela velocidade.
- **Planejamento de trajetória (ou geração de trajetória):** consiste no planejamento em tempo real da mudança do veículo de um estado para o próximo, de forma que os limites cinemáticos e dinâmicos dele sejam cumpridos bem como as regras de trânsito e os limites de faixa. Parametrizado por tempo, aceleração ou velocidade.

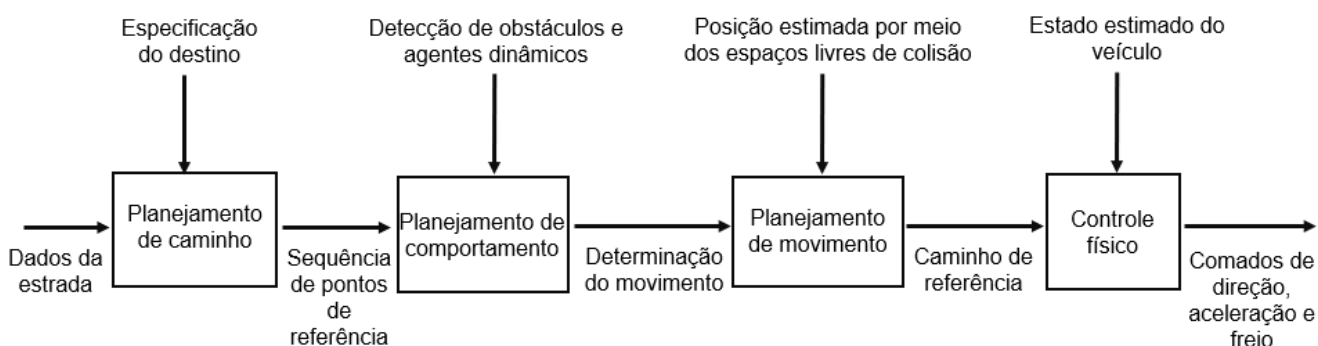
Segundo Paden, Čáp, *et al.* (2016) os veículos autônomos são sistemas com tomada de decisão autônoma, que fazem o processamento de informações com um fluxo de dados advindos dos sensores (como LiDARs, radares, câmeras). Esse processamento juntamente com as informações prévias sobre regras da estrada e dinâmica veicular, selecionam os valores de forma automática para que o movimento do veículo aconteça.

A direção autônoma pode ser dividida em quatro níveis de atividade, ou seja, a percepção do ambiente e a tomada de decisão veicular podem ser organizados em uma estrutura hierárquica que pode ser encontrada em diferentes trabalhos (GONZÁLEZ, PÉREZ, *et al.*, 2015; RODRIGUES, MCGORDON, *et al.*, 2016; ZIEGLER, BENDER, *et al.*, 2014). A Figura 15, foi adaptada de Paden, Čáp, *et al.* (2016) a estrutura é formada por quatro níveis principais: planejamento de caminho, planejamento de comportamento, planejamento de movimento e controle físico.

O planejamento de caminho recebe um destino e determinará uma rota por meio dos dados advindos da rede rodoviária. No planejamento de comportamento

serão determinados os pontos de referência específicos ao longo do caminho planejado, esses pontos possuem dependência com a sinalização da estrada e controle de tráfego por exemplo. Na fase de planejamento de movimento serão efetuados cálculos da posição atual do veículo até o próximo ponto especificado por um planejador de tarefas de direção. Durante esse caminho o veículo deve atingir objetivos específicos como manter-se dentro dos limites da estrada, evitando colisões. A última tarefa envolve o controle físico do veículo, nesse processo serão feitas as conversões de velocidades, orientações, direções e de aceleração do veículo.

Figura 15 - Sistema de Tomada de Decisão de um Veículo Autônomo



Fonte: Adaptado de (PADEN, ČÁP, *et al.*, 2016)

3.2.1 Planejamento de Caminho

O planejamento de caminhos, ou também denominado planejamento de rotas apresenta uma evolução gradativa no âmbito dos transportes individuais e coletivos (TAHA e ABUALI, 2018). A seleção de uma rota pelo sistema de tomada de decisão de um veículo autônomo está localizada no nível mais alto desse sistema, o processamento dos dados que caracteriza a rede rodoviária em que o veículo se encontra permite essa seleção do ponto em que ele se encontra até o destino solicitado (PADEN, ČÁP, *et al.*, 2016).

Problemas envolvendo o planejamento de caminho ideal foram estudados amplamente nas últimas décadas. Esses estudos classificam os veículos autônomos como robôs não holonômicos, que se caracterizam pela igualdade dos graus de liberdade controláveis com os graus de liberdade totais (WASLANDER e KELLY, 2018c). Os veículos autônomos são não holonômicos porque possuem sua descrição em quatro graus de liberdade (duas coordenadas cartesianas, orientação e direção), porém possuem duas restrições cinemáticas, pois movem-se para frente e para trás,

tangencialmente à direção de seu corpo e o raio de direção possui limitação (KATRAKAZAS, QUDDUS, *et al.*, 2015).

O trabalho de Rodrigues, MCGordon, *et al.* (2016) apresenta uma revisão sobre as soluções em forma de requisitos para que um planejamento de caminhos seja bem executado, os requisitos apresentados são:

- Viabilidade: possui referência a um caminho que não possui obstáculos ou áreas impossíveis de ser ultrapassadas.
- Seguro: um caminho que apresenta uma distância segura dos obstáculos apresentados.
- Ideal: apresenta uma distância menor entre o ponto inicial e o ponto final, demanda menor tempo de movimento e até menor consumo de combustível.
- Adequado: ambientes dinâmicos sob algumas perspectivas podem ser classificados como ambientes inadequados. Dessa forma, a geração de caminho necessita de um sistema rápido para que nos casos em que exista um risco dinâmico (como pedestres), seja corrigido em tempo real.
- Adaptável: o planejador de movimento deve apresentar fácil adaptabilidade as mudanças impostas por ambientes dinâmicos, permitindo o movimento contínuo e sem interrupção.
- Eficiente e progressivo: possui um sistema de tomada de decisões eficiente permitindo um movimento progressivo.
- Interativo: deve possuir capacidade de interagir com o ambiente dinâmico de tráfego, dessa forma não ocasionando problemas como congestionamento por exemplo.

Dessa forma, o planejamento de caminhos deve encontrar uma rota que satisfaça condições de segurança viáveis ao usuário e ao ambiente dinâmico em que o veículo se encontra, cumprindo os requisitos exemplificados.

3.2.2 Planejamento de Comportamento

Com a definição de um caminho, o veículo deve apresentar o movimento de forma que exista interação com os elementos presentes no tráfego cumprindo as regras impostas pela estrada e convenções de condução. O nível de planejamento

comportamento possui como função a seleção do comportamento de direção em todo o caminho, baseando-se nos elementos que possuem influência ao seu redor, como tráfego, regras da estrada, e infraestrutura de sinalização (placas e semáforos). O planejamento de comportamento pode ser modelado utilizando o conceito de máquina de estado finito, em que as transições possuem dependência com o estado atual do veículo, considerando obstáculos estáticos, obstáculos dinâmicos e regras de trânsito (PADEN, ČÁP, *et al.*, 2016).

O trabalho de Ziegler, Bender, *et al.* (2014) apresenta uma Mercedes-Benz Classe S autônoma, em que para o planejamento de movimento foi utilizada uma máquina de estados relacionada a um fluxo de eventos. O fato da direção em ambientes urbanos poder ser caracterizada com incerteza a respeito das decisões de outros motoristas, ciclistas e até pedestres, problemas envolvendo previsão de comportamento são estudados. Um método de decisão utilizando Processo de Decisão de Markov (MDP) é apresentado por Brechtel, Gindele e Dillmann (2011), a abordagem considera o comportamento dos motoristas ao longo da trajetória.

3.2.3 Planejamento de Movimento

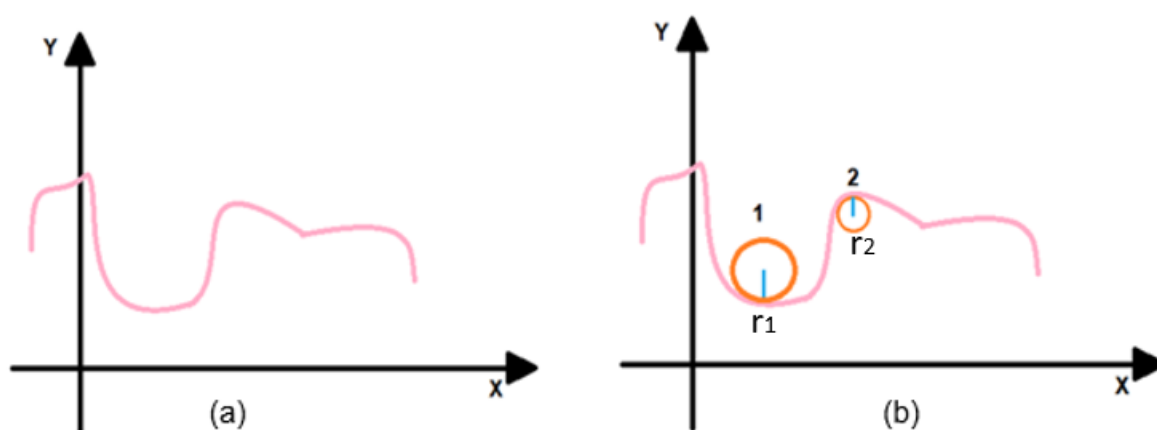
A tarefa de encontrar um caminho viável para o veículo e para o passageiro em um veículo autônomo é responsabilidade do sistema de planejamento de movimento. A parametrização dessa tarefa pode ser feita relacionando tempo, aceleração ou velocidade, uma série de trajetórias é gerada pelo módulo de planejamento de movimento por meio das informações sobre a localização atual do veículo, com uma distância de antecipação o caminho é avaliado com uma função de custo que seja capaz de determinar a trajetória ótima (KATRAKAZAS, QUDDUS, *et al.*, 2015).

Configurações típicas de um planejador de movimento são caracterizadas por apresentar em sua saída a camada de controle veicular, que por sua vez gera um sinal de entrada que regula o veículo para seguir um determinado plano de movimento (PADEN, ČÁP, *et al.*, 2016).

No contexto do planejamento de movimento, são propostas restrições sobre essa tarefa, que dizem respeito a situações dependentes do caminho adotado pelo veículo para que seja possível chegar ao ponto desejado. São responsáveis pela

diminuição do número de direções que um veículo pode adotar em seu respectivo espaço de trabalho, conseqüentemente tornando o planejamento de movimento mais complexo (WASLANDER e KELLY, 2018b). O impacto de uma curva em um caminho durante o planejamento de movimento possui grande importância nas análises do planejamento de movimento. Imaginando uma curva qualquer como a apresentada na Figura 16a, em cada ponto dessa curva podem ser inseridos círculos, Figura 16b, os quais fazem referência a pontos base sobre taxas instantâneas de mudança de curva, com raios diferentes.

Figura 16 - Gráficos de Dois Caminhos Diferentes



Fonte: Adaptado de (WASLANDER e KELLY, 2018b)

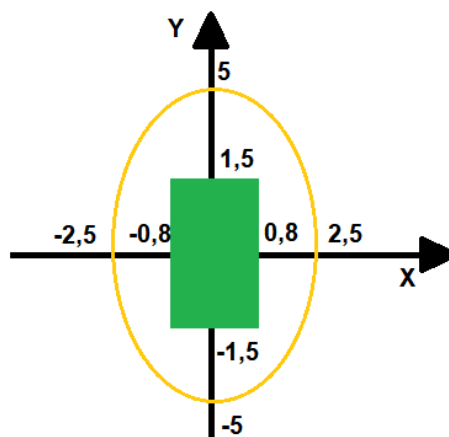
Quando o raio é menor (r_2) a interpretação intuitiva é de que a curva é mais acentuada, conseqüentemente os raios maiores (r_1) um caminho mais suave. Por mais que abordagem dos círculos seja útil como uma interpretação geométrica, uma formulação matemática com maior exatidão pode ser fornecida para essa curva. O cálculo é baseado nas derivadas de primeira e segunda ordem das componentes x e y de um determinado caminho definido com relação ao comprimento de arco. As derivadas do comprimento do arco são dadas por x'' e y'' , conforme a seguinte fórmula:

$$k = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{3/2}} \quad (20)$$

Uma restrição conhecida aborda a elipse de fricção do veículo. Deve-se imaginar uma elipse em torno do veículo, representada na Figura 17, a qual é responsável por apontar a magnitude máxima das forças de atrito que podem existir

entre pneu-solo. Quando as forças justapostas pelo motor do veículo ultrapassam as forças de atrito do pneu ocorrem escorregamentos do pneu. Para que o controle de estabilidade seja mantido, o carro deve preservar-se dentro da elipse de fricção.

Figura 17 - Elipse de Fricção do Veículo



Fonte: Adaptado de (WASLANDER e KELLY, 2018b)

Também devem ser considerados os limites de fricção e restrição de conforto sobre aceleração lateral. Como a velocidade do carro possui uma relação direta com a aceleração lateral veicular e ao raio de giro do caminho (apresentado na primeira restrição), quando é gerado um perfil de velocidade para o veículo autônomo deve ser considerado a curvatura do caminho e aceleração lateral máxima do veículo. Por último, no quesito restrições, devem ser considerados os obstáculos estáticos no processo de planejamento de movimento, englobando veículos estacionados, e postes ou placas que podem ser realocados com o tempo. Basicamente esses obstáculos reprimem locais que o veículo pode estar no decorrer do caminho (WASLANDER e KELLY, 2018b).

O planejamento de movimento é apresentado em diferentes abordagens, o trabalho de Claussmann, Revilloud, *et al.* (2019) descreve o planejamento de movimento contendo cinco aspectos: estimativa de estado, evolução do tempo, planejamento de ações, otimização de critérios e conformidade com as restrições. A forma como essas abordagens são consideradas podem resultar em uma mudança na perspectiva do problema. Gu, Dolan e Lee (2016b) apresentam o desenvolvimento de um planejador de movimento para manobras com a capacidade aprender os estilos de direção de diferentes passageiros.

3.2.4 Planejamento de Trajetórias e de Manobras

O planejamento de trajetórias possui a responsabilidade de planejar uma navegação segura e confortável considerando os limites físicos do veículo e do ambiente (GONZÁLEZ, PÉREZ, *et al.*, 2015). Pode ser abordado de duas maneiras, planejamento baseado em amostragem ou baseado em otimização numérica, a trajetória resultante do primeiro método baseia-se na variedade de candidatos de trajetória com padrões predefinidos, selecionando o custo mínimo envolvendo segurança, conforto e energia (GU, DOLAN e LEE, 2016).

Na segunda abordagem, baseada em otimização numérica, a trajetória encontrada é resultado de uma função de custo diferenciável e restrições, nos casos em que a função de custo e as restrições são funções convexas, a trajetória tende a convergir para um ótimo global (ZIEGLER, BENDER, *et al.*, 2014). Segundo Lim, Lee, *et al.* (2018) a união dos dois métodos é eficiente para o gerenciamento de todos os componentes urbanos, como semáforos e sinais de trânsito que são considerados estados discretos e não podem ser representados por meio de um modelo matemático, mas sim por amostragem. Da mesma forma que a dinâmica veicular e os limites de velocidade permitidos podem ser melhor representados por otimização numérica que por amostragem.

O planejamento comportamental citado anteriormente com o uso das máquinas de estado auxilia a determinação das manobras necessárias em diferentes situações de movimento e direção, entre essas manobras estão inclusas, manutenção de faixa, mudança de faixa, parada, aceleração, desaceleração (LIM, LEE, *et al.*, 2018). Diferentes algoritmos para planejamento de trajetória foram desenvolvidos para aplicações na direção autônoma. Lee e Litkouhi (2012) propõem uma rotina com geração de trajetória em tempo real para mudança de faixa e acompanhamento de local. Uma geração de trajetória utilizando planejamento de rede de estado em espaço de tempo para a direção na estrada é proposto por Mcnaughton, Urmson, *et al.* (2011).

Os diferentes parâmetros adotados para a otimização dos planejadores de movimento e trajetória são ajustados com base nos custos ponderados cumulativos, considerando diferentes características no deslocamento e direção (GU, DOLAN e LEE, 2016). Diferentes algoritmos são utilizados para a implementação do planejamento de trajetória, e a maioria deles faz uso de duas etapas, primeiro ocorre a geração de uma trajetória em um espaço de baixa resolução ou dimensão, depois

acontece a análise de uma dimensão com maior resolução (KATRAKAZAS, QUDDUS, *et al.*, 2015). Mais detalhes desses algoritmos são apresentados na Seção 3.3.

3.2.5 Controle Físico do Veículo

O controle físico do veículo possui como característica o uso de um controlador de *feedback* que faz a seleção das entradas do atuador de forma apropriada para a realização do movimento planejado juntamente com a correção dos erros de rastreamento, por isso o uso de um controlador robusto e estável é considerado importante nesses casos.

O trabalho de Paden, Čáp, *et al.* (2016) demonstra um problema de otimização bem como uma solução para um problema de estabilização de trajetória. Uma análise de estabilidade é considerada com o objetivo de determinar a origem em um sistema que varie no tempo, os autores consideram ainda que na prática as trajetórias de referência são finitas, e devem possuir uma resposta transitória do sistema.

3.3 ALGORITMOS DE BUSCA DE CAMINHOS

Com o desenvolvimento de um modelo do ambiente, a busca pelo melhor caminho entre a posição inicial e final é iniciada, essa tarefa pode acontecer com a utilização dos algoritmos de busca. Pesquisas e aplicações para planejamento de caminho envolvendo algoritmos possuem como principal missão adotar um caminho com ausência de colisões, o algoritmo deve favorecer ao sistema veicular que o deslocamento apresente um custo ótimo, ou quase ótimo (PISSARDINI, 2014).

Técnicas utilizadas para a pesquisa de melhor caminho envolvem configurações e estados de pesquisa incremental, o que possibilita que informações previamente utilizadas promovam maior velocidade a essa pesquisa (KATRAKAZAS, QUDDUS, *et al.*, 2015). Nos mapas gerados previamente são aplicados os algoritmos de planejamento que promovem a definição das posições e ações que o veículo deverá ocupar no ambiente em direção ao ponto objetivo.

O grupo de algoritmos de busca de caminhos compõe a teoria dos grafos em pesquisas operacionais para a resolução de problemas combinatórios sob uma representação gráfica (CLAUSSMANN, REVILLOUD, *et al.*, 2019). Grafos são arranjos formados por nós ou vértices conectados por arcos ou linhas, e esses arcos podem ter pesos atribuídos (SANTOS, 2019). No trabalho de Pissardini (2014) os algoritmos de planejamento são divididos em duas classes, a primeira classe abrange planejamento em mapas topológicos e a segunda planejamento em mapas métricos.

Nos algoritmos aplicados em mapas topológicos, o planejamento de rota é feito por meio da busca de grafos para a geração de caminhos de menor custo. Dessa forma, nesses algoritmos são desenvolvidos cálculos entre o vértice inicial (posição inicial do veículo) e o vértice final (posição final ou ponto objetivo), resultando em um conjunto de vértices e arestas que interligam a posição inicial com o ponto objetivo. Nos algoritmos para mapas métricos as técnicas envolvidas fazem a conversão dos mapas métricos em mapas topológicos e consideram cada célula ou região métrica como um vértice e as adjacências como arestas de um grafo.

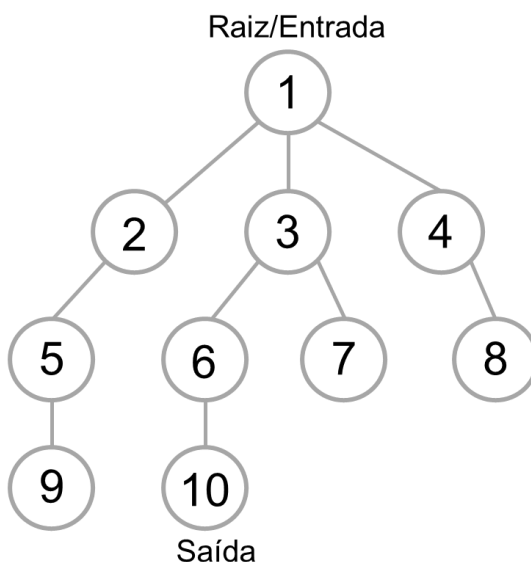
Outra classificação sobre esses algoritmos é apresentada no trabalho de Santos (2019), dividida em duas categorias, a primeira delas considera algoritmos que solucionam problemas de grafos com arcos e pesos iguais, e a segunda categoria abrange algoritmos usados para problemas de busca em grafos com arcos de pesos fixos, distintos e não-negativos. Ambas as classificações consideram a busca por grafos nos mapas, os principais algoritmos presentes na categoria da busca com arcos de pesos iguais são: busca em largura (*Breadth-First Search* – BFS), busca em profundidade (*Depth-First Search* – DFS) e árvores aleatórias de exploração rápida (*Rapidly- Exploring Random Tress* – RRT). Na segunda categoria, algoritmos com arcos de pesos fixos, distintos e não negativos, encontram-se os algoritmos de Dijkstra e A* (*A-Star*) (GIESBRECHT, 2004).

3.3.1 Algoritmo de Busca em Largura (*Breadth-First Search* – BFS)

Algoritmo utilizado para buscar uma estrutura de dados de um grafo com arcos de pesos iguais, ou uma árvore. O início da raiz da árvore e explora os nós vizinhos antes de mover-se em profundidade para os nós (ou vértices) presentes nos próximos níveis. Foi criado por Konrad Zuse em 1945, a ideia do algoritmo consiste em um grafo

e um vértice inicial (raiz) em sua entrada e em sua saída estará a ligação entre os nós descendentes, resultado de um caminho mais curto até a raiz (CORMEN, LEISERSON, *et al.*, 2009). O uso desse algoritmo é adequado para problemas pequenos e com poucas soluções (GIESBRECHT, 2004).

Figura 18 - Representação do Algoritmo de Busca em Largura

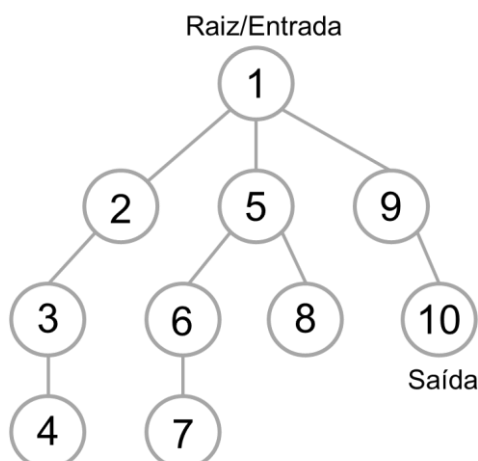


Fonte: Autoria Própria.

3.3.2 Algoritmo de Busca em Profundidade (*Depth-First Search* – DFS)

Um pouco diferente do algoritmo de busca em largura, esse algoritmo faz a busca entre os pontos inicial e final dos grafos de forma rápida. Não faz uso de informações adicionais sobre os nós que já foram explorados, dessa forma seguindo o trajeto até encontrar o ponto objetivo. Se o caminho não leve ao destino, ocorre um recuo da busca e um novo percurso é iniciado, dessa forma esse método não realiza a exploração de caminhos simultaneamente, escolhendo o caminho mais curto até o ponto objetivo (SANTOS, 2019). O uso desse algoritmo é indicado para problemas pequenos, mas com diferentes soluções, para problemas mais complexos a busca em profundidade não apresenta convergência porque o comprimento de um caminho solução tenderá ao infinito (OTTONI, 2000).

Figura 19 - Representação do Algoritmo de Busca em Profundidade



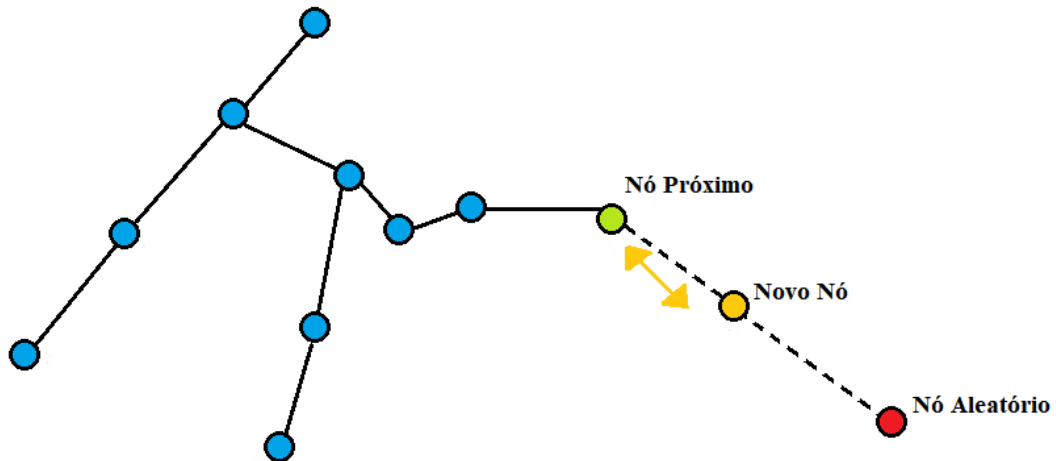
Fonte: Autoria Própria.

3.3.3 Algoritmo de Árvores Aleatórias De Rápida Exploração (*Rapidly-exploring Random Trees – RRT*)

O RRT faz a construção de uma árvore utilizando amostragens aleatórias do espaço, a árvore tem início em um ponto e se propaga de forma a encontrar uma direção favorável ao ponto objetivo, essa expansão ocorre de forma gradativa. No trabalho de Noreen, Khan e Habib (2016) está descrita como a propagação dessa árvore ocorre. O resultado se dá em um conjunto de iterações, e cada iteração um novo estado aleatório é definido.

Por exemplo, o veículo encontra-se em um ponto $Z_{inicial}$ e precisa encontrar o melhor caminho para o ponto $Z_{objetivo}$ acontece uma expansão da árvore e um estado aleatório é selecionado (nó aleatório). Caso a amostra se encontre em uma região com ausência de obstáculos, cria-se um nó que servirá de pesquisa para árvore verificar o nó mais próximo. Se o nó aleatório se encontra acessível, o nó próximo cria uma nova margem para que o caminho seja expandido, caso contrário, retorna para o novo nó utilizando uma função de direção. Existe a possibilidade de classificar o RRT em três partes essenciais, sendo: escolha de um vértice para expansão, expansão e condição de terminação (VOJTECH *et al.*, 2009). A representação desse processo descrito pode ser observada na Figura 20.

Figura 20 - Processo de Expansão da Árvore Algoritmo RRT



Fonte: Adaptado de (NOREEN, KHAN e HABIB, 2016)

Uma variação do algoritmo RRT é o RRT* , nessa variação foram introduzidos dois recursos denominados busca de vizinhos próximos e religação das operações em árvore (KARAMAN e FRAZZOLI, 2011) . O processo envolvendo esses dois passos ocorre dentro de uma área circular, com raio definido na equação (21) em que d é a dimensão do espaço que a árvore fará atuação e γ representa a constante de planejamento com base nesse espaço. A religação faz a reconstrução da árvore dentro desse raio com área (k), à medida que ocorrem as iterações aumenta também a melhoria do custo do caminho, isso acontece pela propriedade assintótica.

$$k = \gamma \left(\frac{\log(n)}{n} \right)^{\frac{1}{d}} \quad (21)$$

Jeon, Choi e Kim (2013) apresentam o emprego do algoritmo RRT* para um modelo dinâmico chamado meio carro, a utilização desse modelo promove trajetórias viáveis em um ambiente com ausência de obstáculos. Por mais que a utilização de um modelo dinâmico possa exigir maior esforço computacional para execução da viabilidade das arestas das árvores, ao combinar esse algoritmo, as soluções apresentadas em um intervalo de entrada de controle foram ideais.

A grande vantagem conhecida nas aplicações desse algoritmo envolve o fato de fornecer de forma rápida as soluções para problemas com complexidade considerável, fazendo o uso de uma amostragem aleatória no espaço (KARAMAN *et al.*, 2011). Na Seção **Erro! Fonte de referência não encontrada.** desse documento foram citadas as restrições não holonômicas que podem aparecer em um ambiente

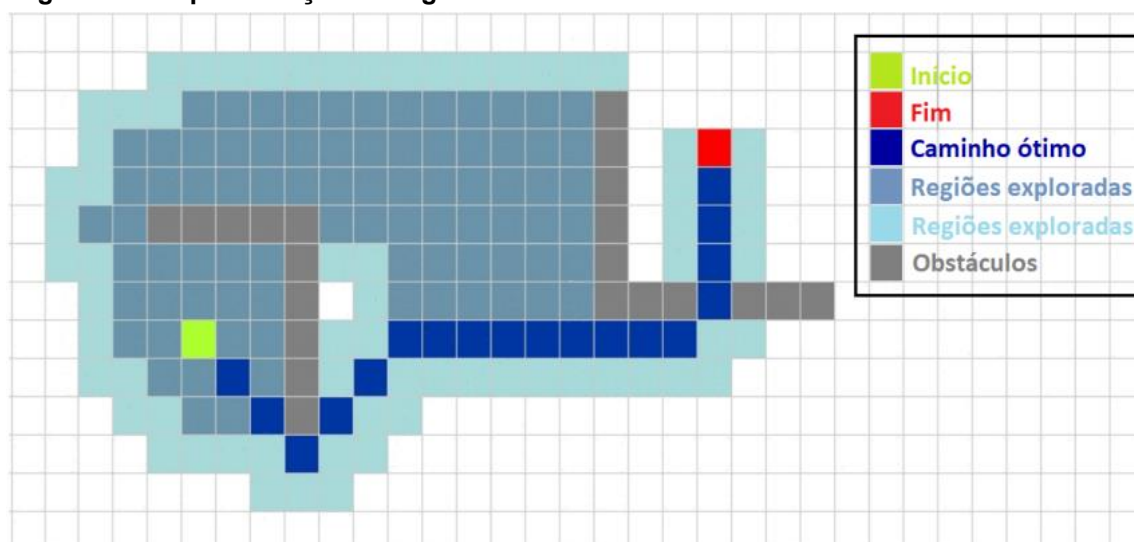
dinâmico, o algoritmo RRT suporta essas restrições de forma positiva. Porém, como desvantagem existe a possibilidade da geração de caminhos bruscos e da submissão associada aos pontos próximos para que exista a expansão da árvore (MURPHY e NEWMAN, 2011).

3.3.4 Algoritmo A* (A-Star)

O princípio desse algoritmo é baseado em uma função de custo definida como $f(n) = g(n) + h(n)$, em que $g(n)$ refere-se ao custo do deslocamento do ponto inicial ao ponto objetivo e $h(n)$ refere-se a heurística de estimação do custo desse deslocamento. Foi inicialmente proposto por Hart et al (1968), os arcos dos grafos são constituídos por pesos fixos associados aos custos calculados. Dessa forma, o algoritmo faz a escolha do sucessor com menor valor.

O que torna o uso desse algoritmo bastante conhecido é o fato do A* subestimar o custo $h(n)$ do caminho ótimo, $h(n) \leq h(n)^*$. Dessa forma o algoritmo explora outros caminhos com custo menor que percurso ótimo, terminando a busca quando o menor caminho é encontrado. Ele também faz uma busca considerada mais rápida em relação a outros algoritmos que fazem o uso da mesma heurística para encontrar o trajeto mais curto (ZANCHIN, 2018). Uma representação do algoritmo é apresentada na Figura 21.

Figura 21 - Representação do Algoritmo A*

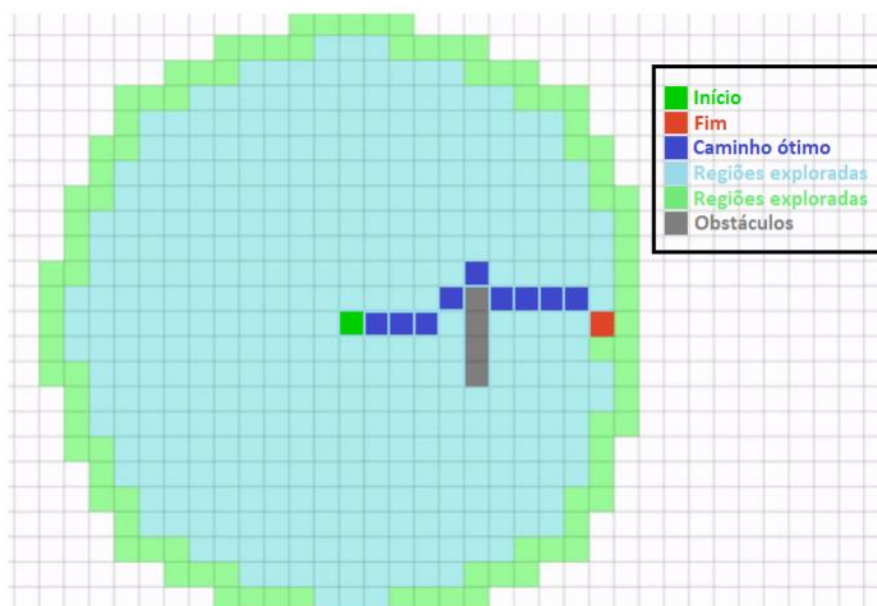


Fonte: (Santos, 2019)

3.3.5 Algoritmo de Dijkstra

Considerado uma variação do algoritmo A^* , esse algoritmo não faz uso de heurísticas na busca pelo percurso ótimo. Em que $h(n) = 0$ e $f(n) = g(n)$, dessa forma, ele executa apenas a expansão do caminho mais curto até o ponto objetivo. Usado para encontrar o caminho de custo mínimo entre vértices de um grafo com arcos de peso não-negativo (SANTOS, 2019). Um vértice é selecionado como raiz da busca e então o algoritmo efetua o cálculo do custo mínimo desse vértice para os demais vértices do grafo (DIJKSTRA e OTHERS, 1959).

Figura 22 - Representação do Algoritmo de Dijkstra



Fonte: (Santos, 2019)

O princípio do algoritmo é partir de uma estimativa inicial para um custo mínimo e sucessivamente ajustar esse valor estimado (DIJKSTRA e OTHERS, 1959). Então ocorre a classificação dos vértices pelo algoritmo, quando o vértice está fechado significa que o algoritmo já obteve um caminho de custo mínimo com o vértice considerado como raiz da busca até esse ponto, de outra forma o vértice é considerado aberto. No momento em que todos os vértices se encontram fechados, os valores resultantes são os custos mínimos dos caminhos que partiram do vértice raiz até os outros vértices do grafo. O caminho do custo mínimo propriamente dito obtém-se por meio dos vértices denominados precedentes.

3.4 CONCLUSÃO

A preocupação com o planejamento de movimento e todas as estratégias envolvidas acontece pelo fato de englobar todo o processo de decisão veicular durante o movimento. Precisam ser considerados os modelos dinâmicos do veículo, os modelos de estado do ambiente e como o ambiente será representado para percepção do veículo. A composição e criação dos mapas para que os algoritmos de planejamento entrem em ação deve ser feita de forma detalhada e com bastante exatidão para que o deslocamento aconteça. Diferentes algoritmos podem ser empregados em deslocamento de movimento, os ajustes necessários dirão qual sistema apresentará mais sucesso conforme a velocidade do sistema em tomar melhores decisões mantendo a segurança do ambiente como um todo.

4 METODOLOGIA

Esse capítulo apresenta a metodologia utilizada para o desenvolvimento do trabalho, que consiste na criação e definição de um modelo ambiente, do modelo veicular e do algoritmo de planejamento de caminho, bem como as métricas utilizadas para avaliar a aplicabilidade dos algoritmos analisados.

Portanto, para esse trabalho foram adotadas as seguintes considerações:

- Utilização de mapas em grade para criação do modelo ambiente;
- Pontos inicial e objetivo são previamente definidos pela autora;
- A velocidade do veículo é considerada constante durante todo o deslocamento do ponto inicial ao ponto final, pois o objetivo é observar a funcionalidade dos algoritmos na geração de caminhos;
- O cenário de simulação representa um estacionamento delimitado com faixas.

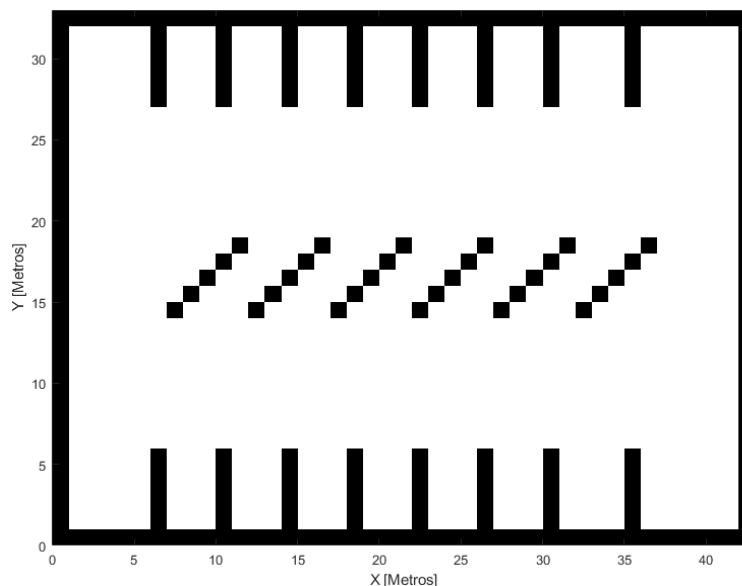
Para avaliação do desenvolvimento e funcionamento de um planejamento de caminhos em ambiente virtual, o *software* Matlab (Mathworks®) foi utilizado juntamente com as funções desenvolvidas para utilização dos algoritmos escolhidos pela autora. Facilitando a análise da empregabilidade dos algoritmos, foram executadas duas simulações, a primeira avaliando o algoritmo de Árvores Aleatórias De Rápida Exploração (*Rapidly-exploring Random Trees* - RRT) e a segunda o algoritmo Híbrido A* (*A-Star*).

4.1 MODELO DE REPRESENTAÇÃO DO AMBIENTE

O ambiente utilizado simula um estacionamento em duas dimensões. Para a representação virtual foi utilizada a técnica de grades de ocupação, apresentada na Seção 3.1.2. A ferramenta Matlab permite, a criação de uma grade de ocupação com valores binários por meio da função *binaryOccupancyMap*. Inicialmente foi desenvolvida uma matriz com as dimensões do estacionamento desejado e as delimitações das vagas que poderiam ser ocupadas pelo veículo, dessa forma a matriz possui tamanho 34x43 (34 linhas e 43 colunas), representando 34 metros no eixo Y e 43 metros no eixo X do gráfico.

As células da matriz que receberam o valor 0 (zero) correspondiam a uma célula vazia (branco) e as células com valor 1 (um) correspondem a uma célula preenchida (preto). O mapa utilizado para a análise dos dois algoritmos, representa um estacionamento formado por vagas verticais e diagonais (em 45°), pode ser observado na Figura 23.

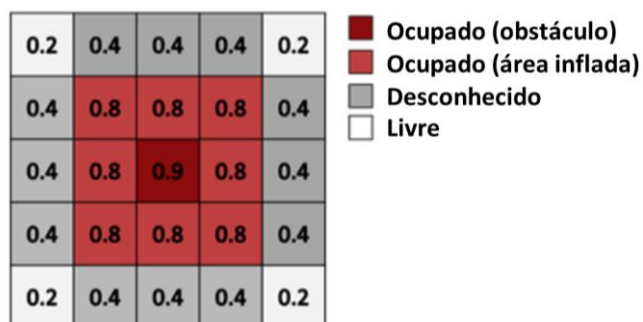
Figura 23 - Representação do Ambiente Considerado para Simulação



Fonte: Autoria Própria.

Posterior a criação da matriz com os valores de células ocupadas e vazias, foi criado um mapa de custos. Definido no *software* Matlab com a função *costmap* responsável pela criação do espaço de busca de planejamento de movimento ao redor do veículo. Algumas informações sobre obstáculos e regiões que o veículo não pode atravessar ou ocupar são representadas por meio dessa função. A Figura 24, demonstra um mapa de custos com diferentes estados de célula.

Figura 24 - Representação de um Mapa de Custos de Amostra e Estados de Células

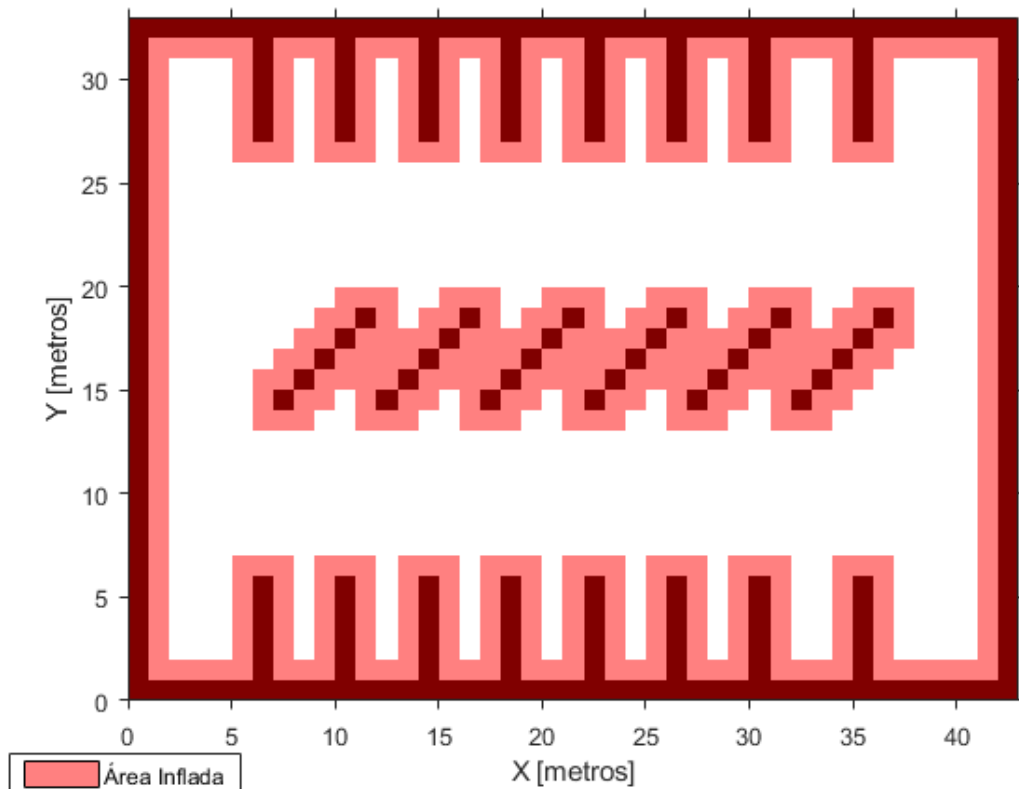


Fonte: Adaptado de (MATWORKS, 2018b)

Os valores em cada célula podem variar entre 0 e 1 e cada valor nesse intervalo representa um estado que a célula pode adotar (livre, ocupado ou desconhecido). As configurações da função *costmap*, apresentam um mapa de custos com uma pequena área que poderá ser ocupada pelo veículo durante o planejamento de caminhos (região em branco). Com isso outras configurações foram ajustadas para que o veículo juntamente com o algoritmo escolhido alcançasse funcionalidade e apresentasse o deslocamento do ponto inicial ao ponto objetivo.

A função denominada *inflationCollisionChecker* responsável pela verificação de colisão no mapa de custos foi utilizada. As propriedades envolvidas nessa função incluem a configuração de verificação de colisão de um mapa de custos criado para o veículo percorrer. Os algoritmos de planejamento de caminho fazem uso dessa configuração de colisão em mapas de custos. Como os obstáculos são inflados uma área de segurança é criada possibilitando um planejamento de caminhos livre de obstáculos (MATHWORKS, 2018a). A Figura 25 apresenta o mapa de custos utilizado para aplicação dos algoritmos, a área em vermelho representa locais ocupados.

Figura 25 - Mapa de Custos da Simulação



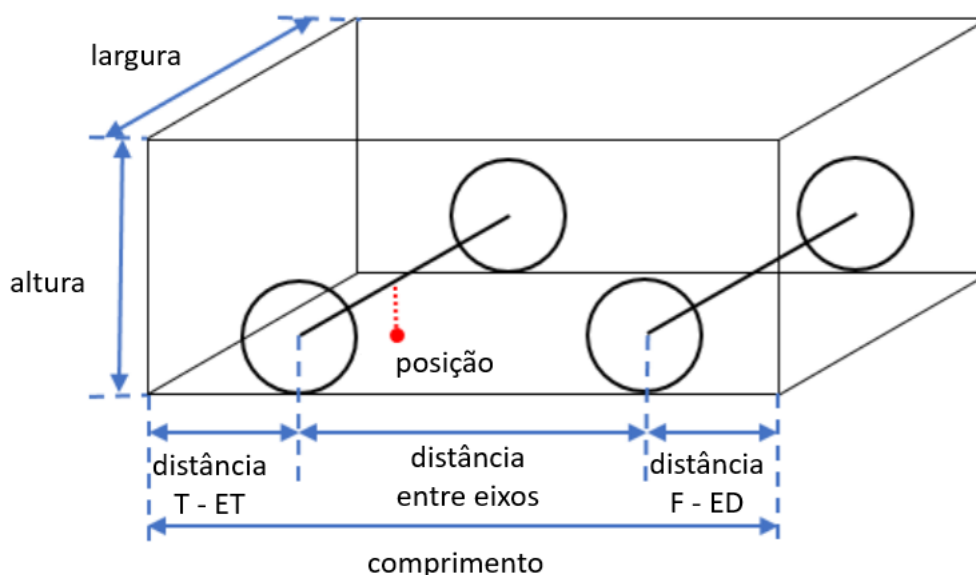
Fonte: Autoria Própria.

O fator de inflação usado durante o planejamento de caminho frequentemente resulta em soluções mais rápidas, pois a além de uma maior proteção do veículo a área de busca torna-se menor, o que é desejável em implementações de tempo real (LIKHACHEV, GORDON e THRUN, 2003).

4.2 REPRESENTAÇÃO DO VEÍCULO

Para que os algoritmos interpretem a presença do veículo no mapa de custos e definam a área de colisão com as funções apresentadas na seção anterior, dimensões veiculares foram definidas. Utilizando a função *vehicleDimensions* um objeto foi criado armazenando as dimensões do veículo. A Figura 26 mostra como as dimensões são incluídas no objeto em questão. A posição do veículo possui como representação um único ponto, localizado no eixo traseiro do veículo, essa posição corresponde ao centro natural de rotação do veículo (MATHWORKS, 2018c).

Figura 26 - Dimensões do Veículo



Fonte: Adaptado de (MATWORKS, 2018c)

Dessa forma são consideradas as medidas de altura, largura e comprimento do veículo. A distância entre eixos, distância da traseira até o eixo traseiro (T-ET), distância da frente até o eixo dianteiro (F-ED), os valores utilizados nas simulações abordam medidas adotadas pela Mathworks® como valores típicos para um veículo sedan, na Tabela 2 são especificados os valores utilizados nesse trabalho.

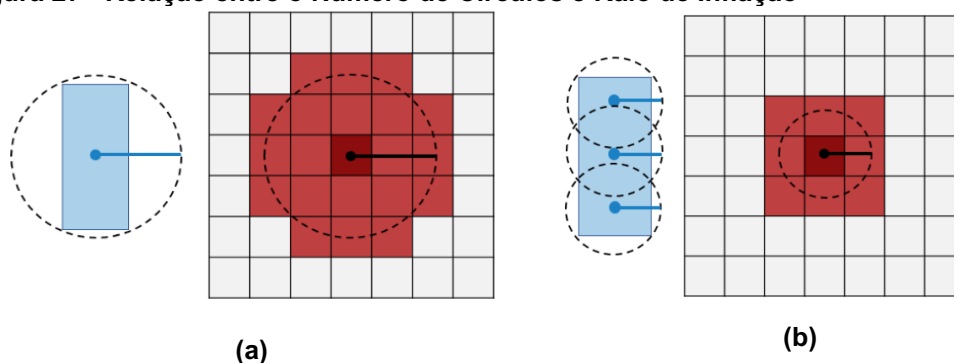
Tabela 2 - Dimensões do Veículo

Parâmetro Matlab	Valor (m)
Comprimento	4,7
Largura	1,8
Altura	1,4
Distância entre eixos	2,8
Distância F-ED	0,9
Distância T-ET	1

Fonte: Autoria Própria.

A relação entre posição do veículo e a área com obstáculos possui algumas etapas a serem seguidas que consideram as dimensões do veículo. O raio de inflação citado na seção anterior é calculado a partir das dimensões do veículo, e refere-se ao menor raio do conjunto de círculos sobrepostos ao veículo virtualmente que o envolvem longitudinalmente. O número de círculos adotado é inversamente proporcional ao raio de inflação, ou seja, quando o número de círculos aumenta, o raio de inflação diminui, dessa forma permitindo uma verificação de colisão mais precisa (MATHWORKS, 2018b).

Na Figura 27a o veículo é representado pelo retângulo em azul com apenas um círculo, ou seja, um centro. Na Figura 27b o veículo possui três centros, dessa forma é possível observar a relação entre o número de círculos e o raio de inflação. A utilização dessa configuração considerando as dimensões do veículo é importante na aplicação dos algoritmos de busca de caminhos pois, ajudará o algoritmo a criar o espaço de busca ao redor do veículo (MATHWORKS, 2018b).

Figura 27 - Relação entre o Número de Círculos e Raio de Inflação

Fonte: Adaptado de (MATWORKS, 2018b)

4.3 APLICAÇÃO DE ALGORITMOS DE BUSCA DE CAMINHOS

De posse do ambiente representado foram escolhidos dois algoritmos para avaliação no comportamento do planejamento de movimento. O primeiro deles, o foi algoritmo Híbrido A* (*Hybrid A-Star*), o segundo foi o algoritmo de Árvores Aleatórias De Rápida Exploração (*Rapidly-exploring Random Trees – RRT*) por apresentar como característica principal o uso adequado para problemas com obstáculos no ambiente e restrições diferenciais do algoritmo escolhido. Ambos algoritmos podem ser classificados como planejadores de caminho global, o que levou a escolha da autora para análise em simulação (MATHWORKS, 2020a). As próximas subseções deste capítulo apresentam detalhadamente como a aplicação dos respectivos algoritmos foi abordada.

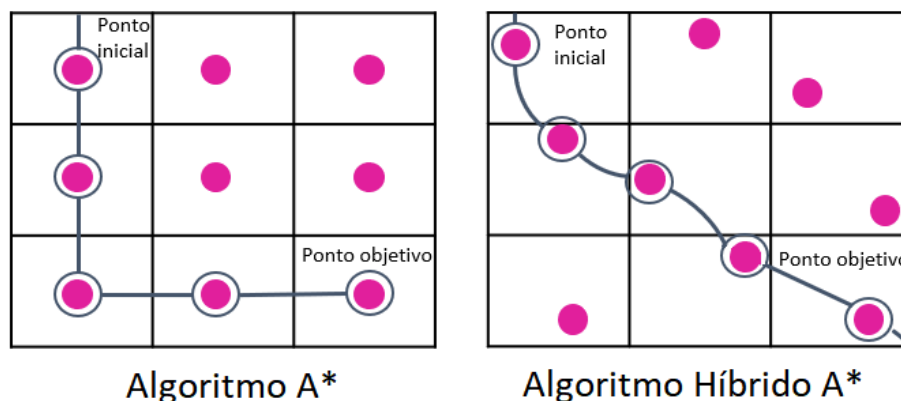
4.3.1 Simulação Utilizando o Algoritmo A* Híbrido (*Hybrid A-Star*)

A escolha e representação de um modelo ambiente, juntamente com áreas com possível ocupação para o veículo foram seguidas de outras configurações de simulação para o deslocamento do veículo. Entre essas configurações encontra-se a função *validatorVehiclecostmap*, responsável por validar os estados e movimentos discretizados com base nos valores de custos do mapa, dessa forma, verificam-se as áreas ocupadas no mapa e interpretam-nas como estados inválidos, ou seja, espaços que não podem ser ocupados. Essa função é adotada para toda a área 2D do mapa, uma distância de interpolação é aplicada para que essa validação entre os seguimentos seja feita (a distância adotada nesse caso foi de 0,1 metros).

A diferença marcante entre o algoritmo A* e o algoritmo Híbrido A* está no fato da suavidade apresentada pelo caminho. Se um mapa de grades dividido em diferentes células possui pequenos centros representados no interior de cada célula, o algoritmo A* irá representar um caminho unindo os pequenos centros com linhas retas. Como os veículos possuem um raio de curvatura mínimo para seguir o caminho, o que pode acontecer são manobras bruscas durante o movimento nos pontos em que as retas se encontram, formando ângulos de 90°. O que o algoritmo A* Híbrido faz é considerar a dinâmica veicular com o objetivo de tornar o caminho mais suave,

livre de manobras bruscas (VIRDI, 2020). A demonstra a diferença entre os dois algoritmos na criação de caminhos.

Figura 28 - Comparação entre Algoritmo A* e Algoritmo Híbrido A*



Fonte: Adaptado de (VIRDI, 2020)

Foram definidos os pontos inicial e final, as validações de caminho foram aplicadas, então a função do algoritmo foi adotada (*plannerHybridAStar*), algumas propriedades precisam ser consideradas para que o caminho suavizado entre o ponto inicial e final seja garantido. A função do algoritmo em questão cria um caminho suave em um espaço de duas dimensões para veículos com restrições não-holonômicas, esse caminho suave é planejado utilizando algoritmo Híbrido A*. Como entrada é necessário especificar o mapa em duas dimensões conforme citado anteriormente e também a função que verificará áreas que podem ser ocupadas pelo veículo (*stateValidator*).

O algoritmo verifica as colisões no mapa fazendo uma interpolação dos chamados primitivos de movimento em uma expansão analítica, com base na propriedade *stateValidator*. Essa propriedade ainda proporciona ao algoritmo uma análise com embasamento no tamanho da célula. A importância de apresentar as áreas infladas antes de aplicar o algoritmo possui relação com o tamanho do veículo (PETEREIT, EMTER, *et al.*, 2012).

Outras propriedades incluindo a função *stateValidator* são abordadas em função do algoritmo Híbrido A* na plataforma virtual. Entre elas estão as duas propriedades que consideram o raio de giro mínimo do veículo, a primeira é a *MinTurningRadius* (raio de giro mínimo do veículo) que como o próprio nome diz deve ser especificada como o raio de curvatura mínimo que o veículo possui. E associada a essa propriedade está o comprimento dos primitivos de movimento que serão

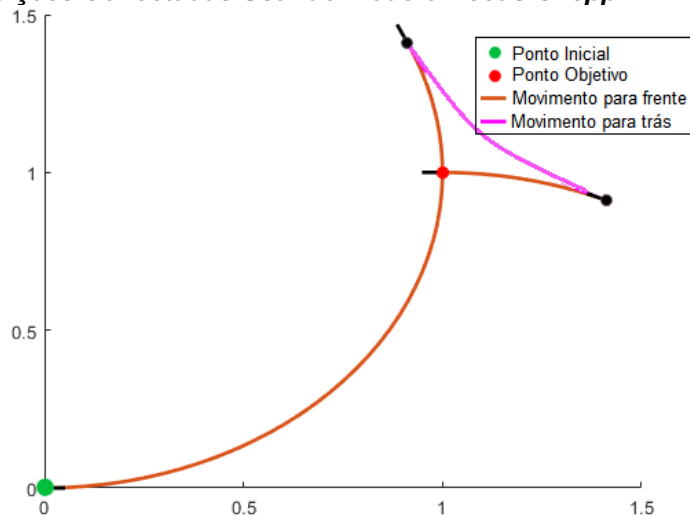
gerados *MotionPrimitiveLength*, esse valor por sua vez não pode ser maior que um quarto do comprimento de circunferência de um círculo baseado na propriedade anterior, essa relação pode ser observada na equação (22).

$$\min \text{TurningRad} = \frac{(2 * \text{Motion PrimitiveLength})}{\pi} \quad (22)$$

Outra propriedade envolvida é sobre o intervalo de expansão analítica aplicado no algoritmo (*AnalyticExpansionInterval*), nesse intervalo uma expansão analítica com menor custo é disponibilizada. O algoritmo faz uma expansão dos nós com menor custo possível, o intervalo assume o valor de um número inteiro positivo. O número de nós que são expandidos está ligado ao número de primitivas que serão geradas na sua direção e em sua validade, dessa forma o ciclo de expansão se repete até que o ponto objetivo seja alcançado (MATHWORKS, 2019b).

A busca pelo caminho com melhor suavidade do ponto inicial ao ponto objetivo acontece por meio de uma árvore. No ambiente virtual utilizado nesse trabalho, foi representada por um modelo de conexões conhecido como *Reeds-Shepp* (MATHWROKS, 2019a). Esse modelo é um tipo de conexão de caminhos entre duas posições em uma sequência de até cinco movimentos. O nome é atribuído aos autores Reeds e Shepp (1990) que provaram que um veículo que se movimenta para frente e para trás (veículo com marcha ré) apresenta um conjunto de nove classes de caminhos mais curtos.

Figura 29 – Posições Conectadas Usando Modelo *Reeds-Shepp*



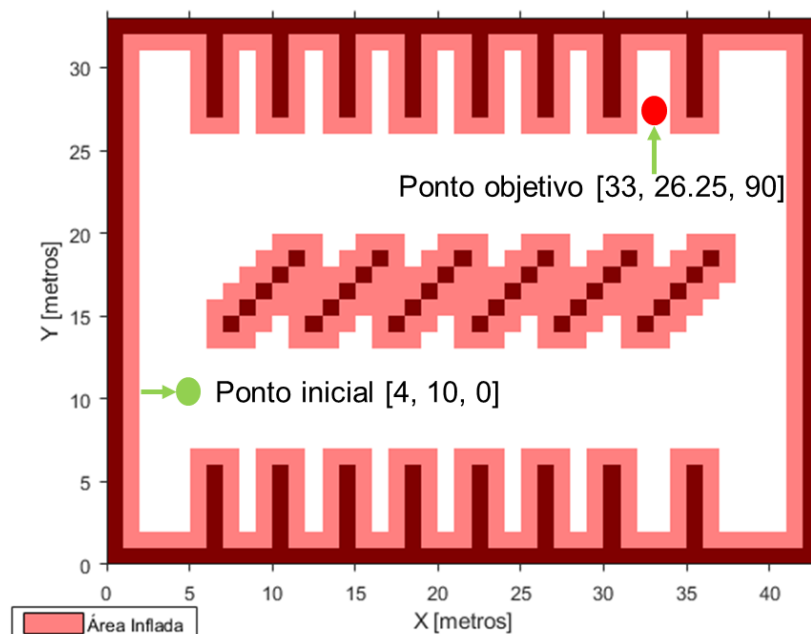
Fonte: Adaptado de (MATHWORKS, 2019a)

Essa análise foi feita utilizando segmentos de reta e arcos com raio de giro mínimo, os arcos com raio de giro mínimo do movimento resumem-se em uma rotação em torno do eixo do veículo fazendo com que o menor caminho proposto possua sempre a mesma estrutura:

- Uma rotação sobre si mesmo com direção ao próximo ponto desejado;
- Um movimento em linha reta; e
- Uma rotação para a orientação final desejada.

Os pontos inicial e objetivo são representados como três coordenadas $[x,y,\theta]$, a primeira coordenada indica a posição x , a segunda indica a posição y e a terceira indica a orientação do veículo. Dessa forma, o ponto inicial foi definido como $[4, 10, 0]$ e o ponto objetivo como $[36, 26.25, 90]$. No ponto objetivo o ângulo de 90° indica a orientação que o veículo deverá adotar ao chegar no destino com relação ao ponto inicial. Como inicialmente o veículo encontra-se paralelo ao eixo x , ao final do trajeto a posição assumida deverá ser paralela ao eixo y , ou seja, perpendicular ao eixo x , assumindo dessa forma o ângulo de 90° . Na Figura 30 é possível observar a relação entre o ponto inicial e o ponto objetivo no mapa de custos adotado para simulação.

Figura 30 - Representação dos Pontos Inicial e Objetivo no Mapa de Custos



Fonte: Autoria Própria.

A criação de um mapa com posições validadas, com a definição do algoritmo e as respectivas propriedades necessárias para o desenvolvimento de um

deslocamento suave do ponto inicial ao ponto objetivo, inicia-se o caminho de referência resultante do algoritmo. O caminho de referência é composto pela união do planejamento criado pelo algoritmo com os pontos (inicial e final) definidos pelo usuário. Essa união resulta em um objeto denominado *navPath* (caminho planejado) que une os pontos criados por planejadores de caminho, nesse caso o algoritmo Híbrido A*, esses pontos são armazenados em estados de espaço.

Os estados de espaço são amostras do espaço no caminho especificado, no caso desse algoritmo foi utilizado o *stateSpaceSE2*, composto por vetores de estado representados por $[x, y, \theta]$, em que x e y de maneira análoga as posições inicial e objetivo representam as coordenadas cartesianas e θ o ângulo de orientação. Uma distância euclidiana é responsável pelo cálculo das distâncias entre os pontos, e para a localização referente a rotação e translação do veículo é feita uma interpolação linear (MATWORKS, 2019c). Os parâmetros utilizados na simulação desse algoritmo estão apresentados na Tabela 3.

Tabela 3 - Parâmetros de Simulação Utilizados para o Algoritmo Híbrido A*

Parâmetro Matlab	Valor	Descrição
<i>Analytic Expansion Interval</i>	5	Intervalo para tentativa de expansão analítica do nó de menor custo disponível
<i>Direction Switching Cost</i>	0	Custo aditivo para mudar a direção do movimento
<i>Forward Cost</i>	1	Multiplicador de custo para deslocamento para frente
<i>Interpolation Distance</i>	1	Distância entre posições interpoladas no caminho de saída
<i>Min Turning Radius</i>	4	Raio de giro mínimo do veículo
<i>Motion Primitive Length</i>	4	Comprimento dos primitivos de movimento a serem gerados
<i>Num Motion Primitives</i>	5	Número de primitivas de movimento a serem geradas
<i>Reverse Cost</i>	3	Multiplicador de custo para deslocamento na direção reversa

Fonte: Autoria Própria.

4.3.1 Simulação Utilizando o Algoritmo De Árvores Aleatórias De Rápida Exploração (*Rapidly-exploring Random Trees – RRT*)

A simulação envolvendo o algoritmo RRT foi executada de maneira bem parecida com apresentada anteriormente pelo algoritmo Híbrido A*, o planejador de caminhos definido pela Mathworks faz a construção das árvores de busca de forma incremental por meio de amostras obtidas aleatoriamente do espaço de estado (mapa). Essa árvore ocupa o espaço de estado conectando o ponto inicial ao ponto

objetivo. O processo executado pela função *plannerRRT* para o início da árvore de busca inicia-se com uma amostra de um estado aleatório denominado *StateRand*. Que por sua vez faz a busca nos nós vizinhos e cria um novo estado denominado *StateNearest*, o algoritmo passa por uma nova expansão e um dessa vez um novo estado é alcançado (*StateNew*) e é adicionado a árvore de pesquisa (MATHWORKS, 2019d).

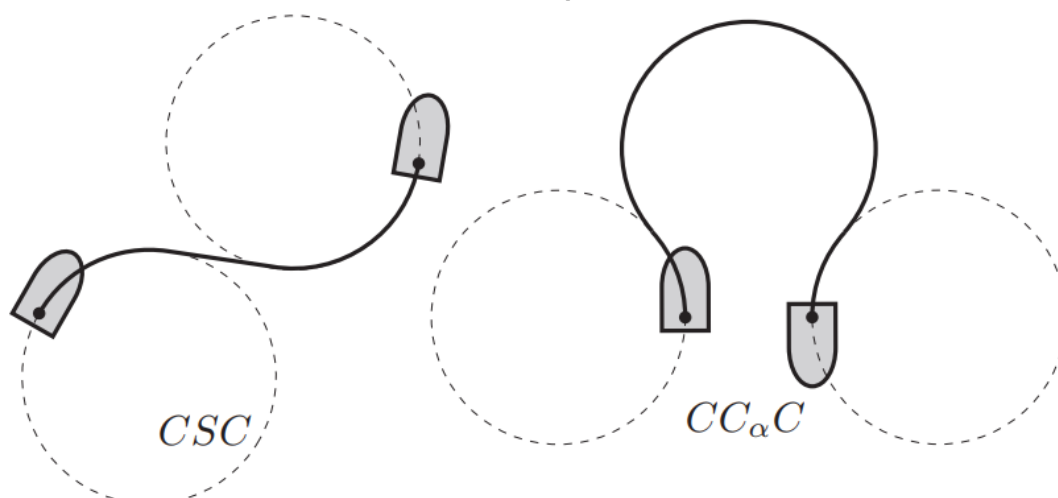
A expansão geométrica e a conexão entre dois pontos podem ser encontradas de maneira analítica sem que a restrição do espaço de estado sofra uma violação. Assim como para o algoritmo anterior a função que representa o algoritmo RRT (*plannerRRT*) apresenta algumas propriedades importantes e que precisam ser ajustadas inicialmente. Entre elas encontra-se o valor da tolerância em torno da posição objetivo (*GoalTolerance*), especificado como um vetor $[x_{Tol}, y_{Tol}, \theta_{Tol}]$, o planejador de caminho terminará o planejamento quando o veículo atingir posição objetivo dentro do limite de tolerância estabelecido. A segunda propriedade inclusa no planejador RRT é denominada *GoalBias*, essa propriedade está relacionada com a probabilidade de o planejador selecionar a posição objetivo no lugar de uma posição aleatória, a especificação é dada como um escalar real em um intervalo entre 0 e 1. Valores maiores aumentam o alcance da meta e apresentam um risco de não conseguir contornar ou evitar obstáculos.

A conexão entre os pontos é armazenada com o espaço de estados denominado Dubin, (*stateSpaceDubins*), esse espaço de estado possui um limite inferior com referência ao raio mínimo de giro do veículo, para que os estados sejam conectados esse modelo de espaço de estado adota a menor curva possível (MATHWORKS, 2019d). O nome Dubin é atribuído em homenagem ao matemático que provou por meio de um teorema que para um carro que se move apenas para frente (sem marcha ré) com um conjunto de controle, apresenta o caminho mais curto com arcos no raio de giro mínimo e segmentos de linha reta (LYNCH e PARK, 2017). Em que o arco circular foi nomeado de C, o segmento de linha reta como S e, o menor caminho entre duas configurações possui a sequência:

- CSC;
- $CC_{\alpha}C$ (em que C_{α} indica um arco com ângulo $\alpha > \pi$).

E qualquer segmento C ou S pode apresentar tamanho nulo. Uma representação desse teorema entre dois caminhos é apresentada na Figura 31.

Figura 31 - Duas Classes de Caminho Mais Curto para um Veículo Sem Marcha Ré



Fonte: (LYNCH E PARK, 2017)

Uma distância máxima entre duas posições é definida como uma propriedade (*ConnectionDistance*), também determinada como um número escalar real e positivo, o planejador fará o cálculo da distância de conexão ao longo do deslocamento entre duas posições incluindo as curvas. É estabelecido também um intervalo de valores de iterações que o planejador utilizará para explorar o mapa de custos (*MinIterations* e *MaxIterations*). Por último de maneira semelhante ao algoritmo Híbrido A* um valor é definido para o raio de giro mínimo do veículo, ou seja, que limitará o ângulo de direção máximo para o planejador, valores menores resultam em curvas mais fechadas. Um valor padrão para essa propriedade é calculado usando uma distância entre eixos de 2,8 metros com ângulo de direção máximo de 35°.

Os pontos inicial e objetivo foram definidos da mesma forma com os mesmos valores do algoritmo Híbrido A*, apresentados na Figura 30. Os parâmetros utilizados para a função do algoritmo RRT podem ser observados na Tabela 4.

Tabela 4 - Parâmetros Utilizados na Simulação do Algoritmo RRT

Parâmetro Matlab	Valor	Descrição
<i>Connection Distance</i>	5	Comprimento do movimento
<i>Goal Bias</i>	0.1	Probabilidade de selecionar a posição objetivo
<i>Goal Tolerance</i>	[0.5 0.5 0.5]	Tolerância em torno da posição objetivo
<i>Max Iterations</i>	10000	Número máximo de iterações do planejador
<i>Min Iterations</i>	100	Número mínimo de iterações do planejador
<i>Min Turning Radius</i>	4	Raio de giro mínimo do veículo

Fonte: Autoria Própria.

4.4 CONCLUSÃO

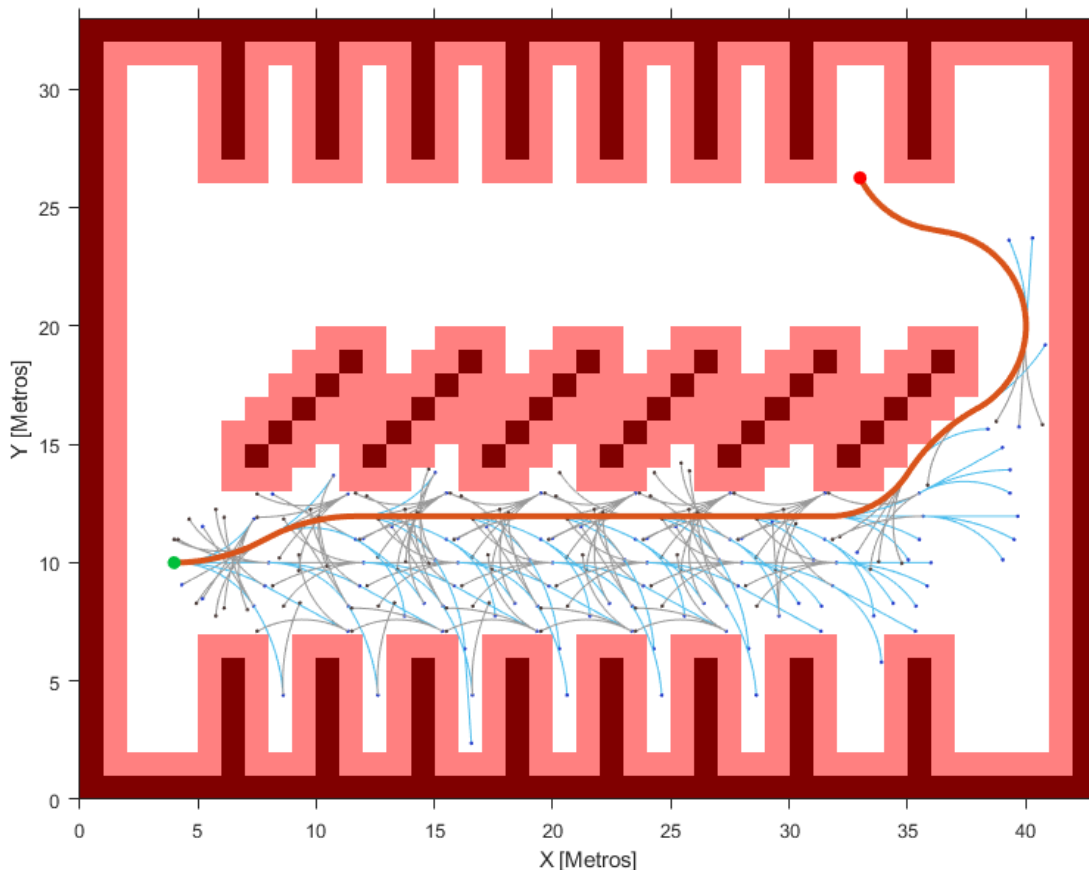
Nesse capítulo foram abordadas as informações metodológicas empregadas na realização desse trabalho. Em que foi apresentado o modelo do estado ambiente utilizado para representar o estacionamento e as métricas utilizadas para criação de um mapa com áreas que não podem ser ocupadas pelo veículo. Foram apresentadas as funções envolvidas para utilização dos dois algoritmos na ferramenta virtual de simulação Matlab® e os conceitos básicos dos ajustes necessários exigidos para o funcionamento da simulação.

5 SIMULAÇÃO E ANÁLISE DE RESULTADOS

Neste capítulo são apresentados os resultados obtidos na simulação utilizando Matlab®. A avaliação do comportamento de cada algoritmo foi feita a partir de dez execuções dos códigos desenvolvidos, que podem ser consultados nos Apêndices A e B. O conjunto de funções apresentadas no capítulo anterior possibilitou a geração dos caminhos do ponto inicial ao ponto desejado.

Inicialmente optou-se pela execução do código do algoritmo Híbrido A*. Além do tempo em cada uma dessas dez execuções, foram analisados os resultados dos caminhos apresentados pelo algoritmo em questão. O caminho apresentado nas dez repetições foi mesmo, sendo possível perceber que os pontos de busca criados também foram os mesmos, conforme Figura 32.

Figura 32 - Representação do Caminho Encontrado pelo Algoritmo Híbrido A*



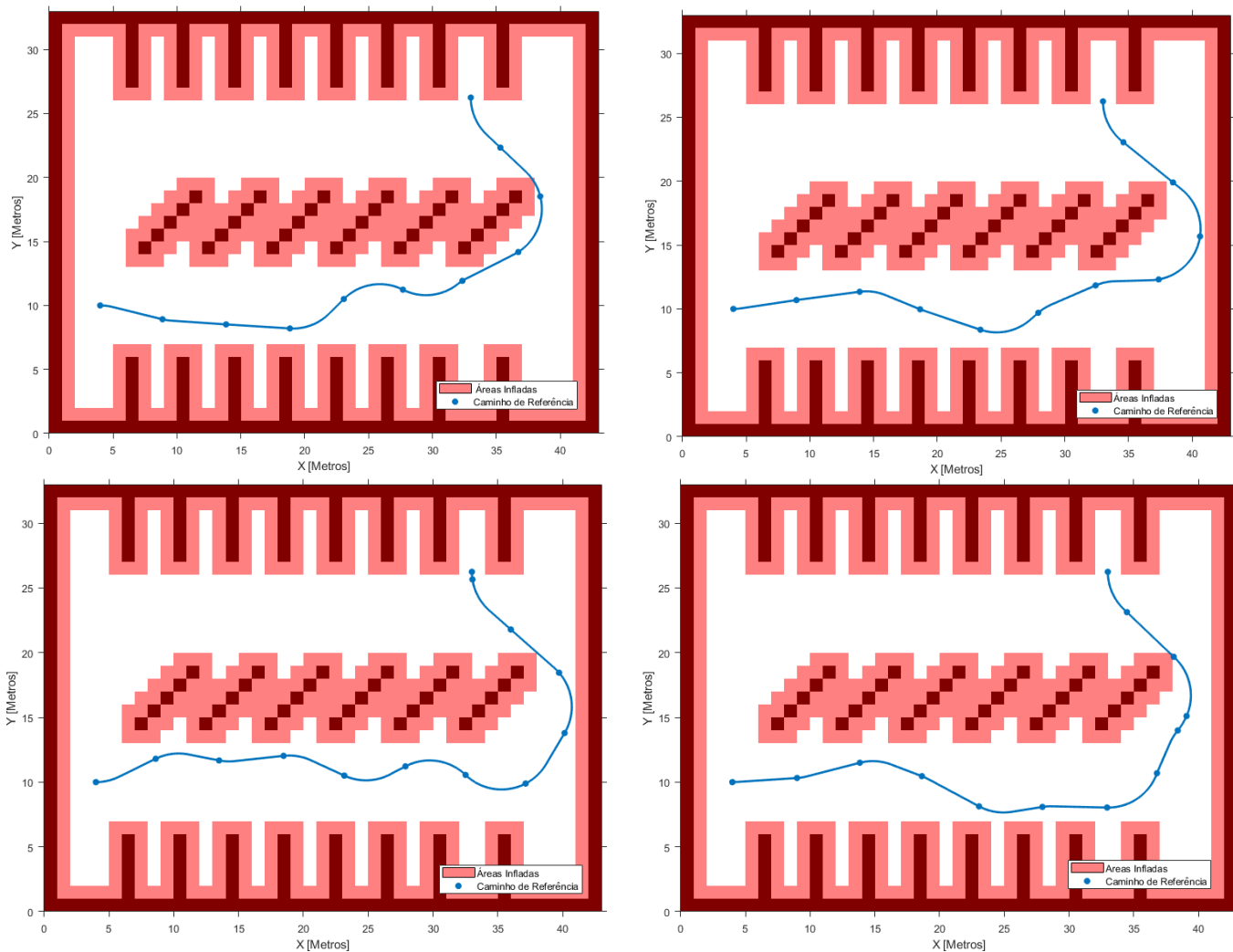
Fonte: Autoria Própria.

O caminho apresentado pelo algoritmo Híbrido A* possui uma certa suavidade, respeitando os limites impostos pelas áreas infladas e apenas fazendo a

curva quando necessário mudar a orientação do veículo para o alcance da posição objetivo em 90°. O mesmo não aconteceu para os caminhos apresentados pelo algoritmo RRT nas 10 simulações realizadas, diferentes pontos foram gerados e consequentemente diferentes caminhos para cada simulação.

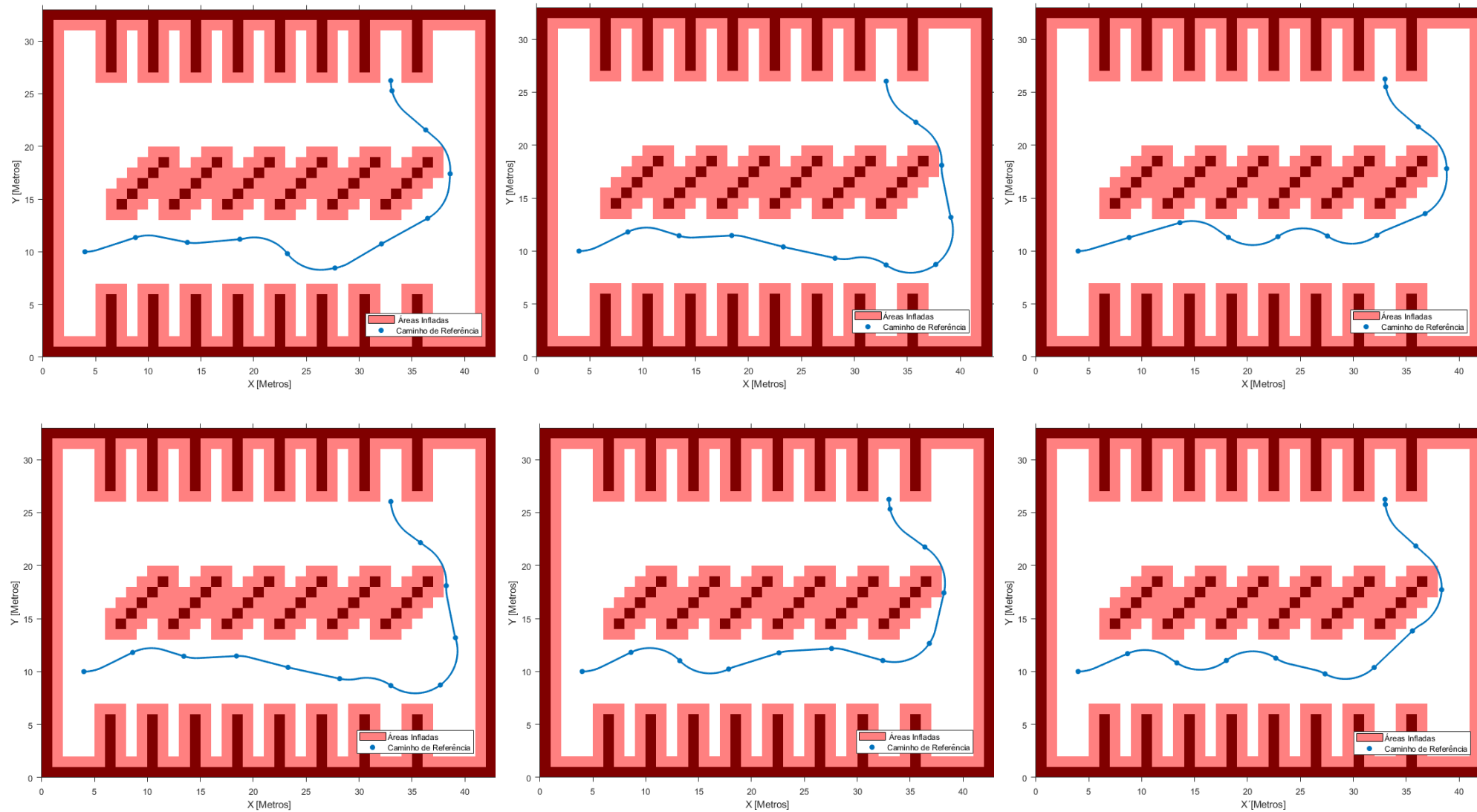
Os caminhos nem sempre apresentaram suavidade durante o deslocamento, se comparado ao caminho apresentado pelo algoritmo Híbrido A*. Considerando as restrições impostas pelo planejamento de movimento apresentados na Seção 3.2.3., a suavidade apresentada no caminho bem como a avaliação dos raios de curvatura ao longo dele, são importantes no deslocamento. Os respectivos caminhos obtidos nas simulações estão representados nas composições da Figura 33 e da Figura 34.

Figura 33 - Representação dos Caminhos Encontrados pelo Algoritmo RRT (1)



Fonte: Autoria Própria.

Figura 34 - Representação dos Caminhos Encontrados pelo Algoritmo RRT (2)



Fonte: Autoria Própria.

Além da avaliação sobre os caminhos apresentados pelos algoritmos, bem como a variação nas diferentes simulações, buscou-se avaliar as distâncias percorridas. Para o algoritmo Híbrido A* essa distância foi obtida utilizando os valores do objeto *navPath* (caminho planejado), como os pontos do caminho são armazenados como um espaço de estado é possível fazer uma interpolação entre esses pontos criados para obter a distância do caminho resultante do algoritmo. Já para o algoritmo RRT os valores obtidos advêm do objeto *refPath* (caminho de referência) em que apresenta as características do caminho criado pelo planejador, entre elas as posições de interpolação e a distância percorrida do ponto inicial ao ponto objetivo, resultados podem ser observados na Tabela 8.

Tabela 5 - Valores de Distância Percorrida Obtidos

Simulação	Algoritmo Híbrido A*	Algoritmo RRT
	Distância percorrida (m)	Distância percorrida (m)
1	50,4463	49,6419
2	50,4463	53,7133
3	50,4463	55,5960
4	50,4463	53,5536
5	50,4463	48,2426
6	50,4463	50,9790
7	50,4463	55,0000
8	50,4463	50,7425
9	50,4463	50,9242
10	50,4463	50,3999

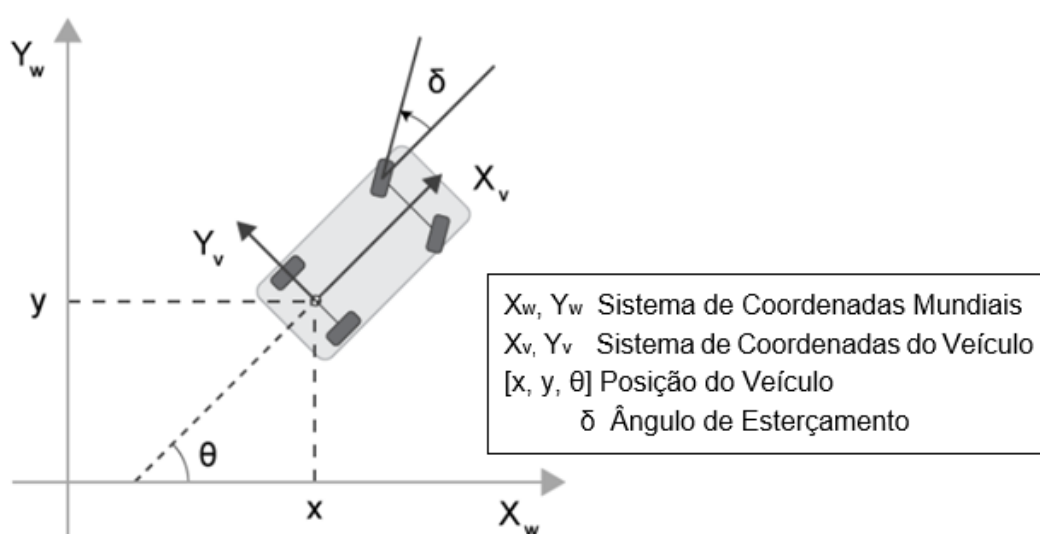
Fonte: Autoria Própria.

As distâncias percorridas nas dez execuções do algoritmo Híbrido A* apresentaram o mesmo valor, visto que o caminho resultante foi o mesmo. Já para o algoritmo RRT as distâncias apresentaram variações, justificadas pelo diferente número de pontos de pesquisa criados pelo algoritmo para o deslocamento do veículo do ponto inicial ao ponto objetivo, em cada simulação.

Além da distância percorrida em cada execução dos códigos foi avaliada a variação do ângulo de esterçamento (*steering angle*). Esse valor foi obtido por meio da função *lateralControllerStanley* que calcula o ângulo de direção no caminho percorrido utilizando o método de Stanley (HOFFMANN, TOMLIN, *et al.*, 2007). Esse método consiste em um controlador que faz o ajuste da posição atual com uma

posição de referência. Considerando a velocidade atual do veículo, assume-se que o veículo está em movimento para frente. A função possui como entrada três valores, posição atual (x,y,θ) , posição final (x,y,θ) e velocidade (m/s) (MATHWORKS, 2018f). A posição inicial adotada para o cálculo desse ângulo considera o eixo traseiro do veículo como referência, na Figura 35 é possível observar a relação da posição inicial do veículo com as coordenadas mundiais, a posição do veículo e o ângulo de esterçamento.

Figura 35 – Relação da Posição do Veículo com as Coordenadas Globais



Fonte: Adaptado de (MATHWORKS, 2018f)

A caixa de ferramentas do Matlab® considera o sistema de coordenadas mundiais cartesiano para destros com definição na ISO8855, em que o eixo z aponta para cima do solo e as unidades são consideradas em metros. O controlador Stanley utiliza o eixo dianteiro como ponto de referência, e analisa o erro de direção e o erro da distância entre o ponto mais próximo do caminho com o eixo dianteiro do veículo, esse erro é denominado *cross-track* (DING, 2020).

Na Figura 36 é possível observar que $\psi(t)$ é o ângulo entre a direção da trajetória e a direção do veículo, o ângulo de direção é definido como δ , esse método possui três leis de orientação:

- Eliminar o erro de direção, adotando $\delta(t) = \psi(t)$;
- Eliminar o erro de *cross-track*, em que acontece encontrando o ponto mais próximo entre a trajetória e o veículo, esse ponto é denotado como $e(t)$.

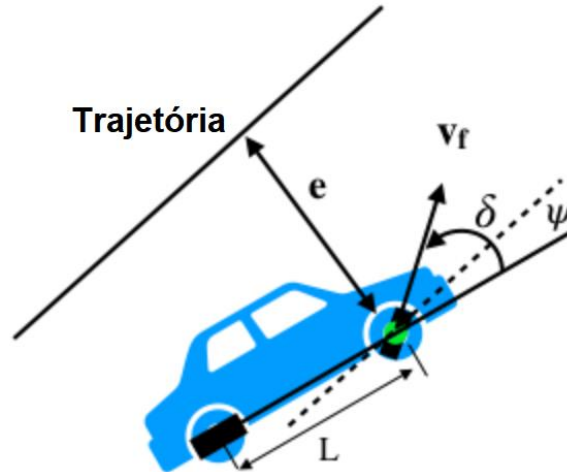
Dessa forma o ângulo de direção é corrigido conforme a seguinte relação:

$$\delta(t) = \tan^{-1} \left(\frac{ke(t)}{v_f(t)} \right) \quad (23)$$

Então desde que os limites de direção máximos sejam considerados, $\delta(t) \in [\delta_{\min}, \delta_{\max}]$, resultando em:

$$\delta(t) = \psi(t) + \tan^{-1} \left(\frac{ke(t)}{v_f(t)} \right), \quad \delta(t) \in [\delta_{\min}, \delta_{\max}] \quad (24)$$

Figura 36 - Relação Geométrica Controlador Stanley



Fonte: Adaptado de (DING, 2020)

Para análise do ângulo de esterçamento neste trabalho a velocidade adotada foi de 4m/s por se tratar de um estacionamento, as posições atual e final foram analisadas por meio dos pontos gerados pelos respectivos algoritmos. Como o algoritmo RRT gerou dez caminhos diferentes com um número desigual de pontos em cada simulação, foram analisados os ângulos ponto a ponto de cada execução.

A Tabela 6 demonstra a variação desse ângulo em cada ponto dos diferentes caminhos gerados pelo algoritmo RRT, na primeira e quinta execuções foram gerados dez pontos de pesquisa pelo algoritmo do ponto inicial até o ponto objetivo, com isso dez ângulos de esterçamento. Na terceira e quarta execuções doze pontos e doze ângulos de esterçamento, nas demais execuções foram gerados onze pontos e onze ângulos de esterçamento.

Tabela 6 - Variação do Ângulo de Esterçamento em Cada Execução do Algoritmo RRT

Simulação	Ângulo de Esterçamento (graus)											
1	35,00	-35,00	-1,83	-35,00	35,00	-35,00	12,0	-35,00	-35,00	35,00		
2	-35,00	6,07	35,00	-20,53	-35,00	35,00	35,00	-35,00	-35,00	0,00	35,00	
3	-35,00	35,00	-35,00	-35,00	-35,00	35,00	-35,00	-35,00	-35,00	-35,00	35,00	24,50
4	-21,22	-35,00	35,00	35,00	-35,00	35,00	-35,00	35,00	-35,00	-35,00	10,00	35,00
5	11,37	-18,80	35,00	35,00	-35,00	-35,00	-35,00	-35,00	-35,00	35,00		
6	-35,00	35,00	-35,00	35,00	-35,00	35,00	8,98	-35,00	-35,00	35,00	35,00	
7	-35,00	35,00	-35,00	35,00	-12,91	35,00	-35,00	-35,00	35,00	-35,00	35,00	
8	-35,00	3,94	35,00	-35,00	35,00	-35,00	35,00	-35,00	-35,00	35,00	30,73	
9	-35,00	35,00	-35,00	35,00	35,00	35,00	-35,00	-35,00	-35,00	35,00	35,00	
10	-35,00	35,00	-35,00	35,00	-35,00	-35,00	-2,86	-35,00	-35,00	35,00	20,14	

Fonte: Autoria Própria.

Observando os valores resultantes para o algoritmo RRT nota-se uma grande variação entre 35° e -35° , esse valor para o algoritmo em questão está diretamente ligado a variável de raio de giro mínimo do veículo (*Minimum Turning Radius*), adotada com valor 4, especificado na Tabela 4. Esse valor é definido como padrão para veículos com a distância entre eixos de 2,8 metros e o ângulo de direção máximo para esses casos é de 35° (MATHWORKS, 2019d). Com isso é possível concluir que o algoritmo RRT explorou pontos ao longo do caminho e respeitou o limite de giro mínimo que o veículo poderia assumir durante o planejamento de caminho e alcançou várias vezes o ângulo máximo de esterçamento entre -35° e 35° .

Para o algoritmo A* a análise foi feita da mesma maneira, considerando os 51 pontos inicial e final gerados para obtenção do caminho. Como os pontos resultantes da criação do caminho foram os mesmos para todas as execuções do código os resultados apresentados na Tabela 7 demonstram a variação do ângulo para os mesmos pontos em todas as simulações. Notou-se uma variação menor no ângulo de esterçamento na aplicação desse algoritmo, visto que os pontos criados pelas primitivas de movimento do algoritmo mantiveram o veículo em caminho com menor oscilação de posição e conseqüentemente menor variação do ângulo de esterçamento.

O valor de raio de giro mínimo definido para esse algoritmo também foi 4, especificado na Tabela 3 (*Minimum Turning Radius*), visto que as dimensões do veículo são as mesmas. O ângulo de esterçamento teve um comportamento melhor nesse algoritmo, visto que a maior variação apresentada foi o valor de -32° , obedecendo os limites impostos para variável de giro mínimo e não alcançando o limite máximo do ângulo de esterçamento, que pode estar entre -35° e 35° .

Tabela 7 - Ângulos de Esterçamento Obtidos nos Pontos do Algoritmo Híbrido A*

Ponto	Ângulo de Esterçamento
1	-2,5767
2	-6,9065
3	-11,0610
4	-14,9530
5	-14,2268
6	-10,3202
7	-6,1489
8	-1,8121
9 ao 28	0
29	-5,1269
30	-13,3885
31	-20,4516
32	-25,9312
33	-26,1105
34	-23,7858
35	-20,9918
36	-17,7993
37	-20,4516
38	-25,9283
39	-29,7320
40	-31,8948
41	-32,4914
42	-31,5417
43	-29,0091
44	-24,8278
45	-18,9726
46	-12,3078
47	-9,5729
48	0
49	-23,9677
50	-29,2287
51	0

Fonte: Autoria Própria.

Foram avaliados também os tempos de simulação obtidos por meio da execução *Run and Time* presente no Matlab®, que ativa um perfil de varredura e calcula o tempo necessário para executar os principais passos presentes no código desenvolvido pelo usuário. Os valores obtidos que correspondem ao tempo total da simulação, ou seja, tempo que o código levou para executar todos os passos e todas as funções presentes, esses valores podem ser observados na Tabela 8.

Tabela 8 – Tempos de Simulação Obtidos

(continua)

Simulação	Algoritmo Híbrido A*	Algoritmo RRT
	Tempo (s)	Tempo (s)
1	9,141	18,876
2	6,881	12,272

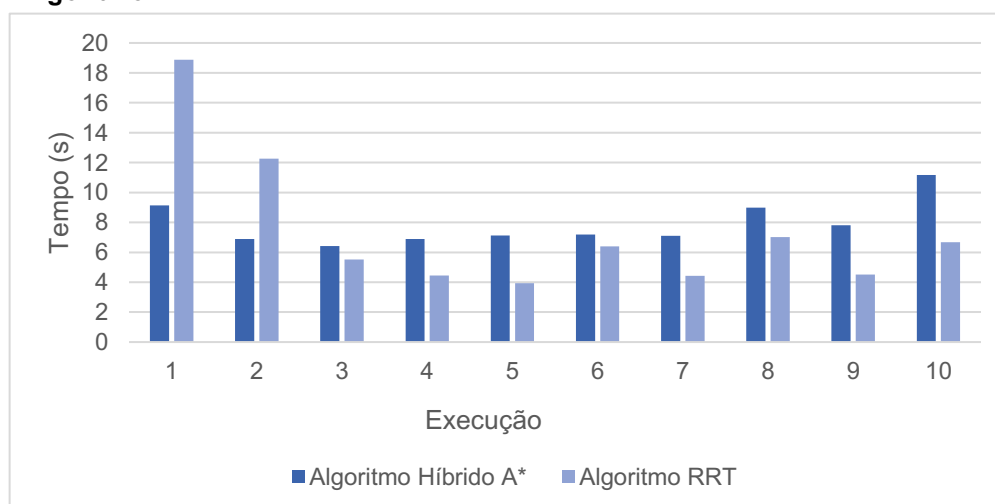
Tabela 8 – Tempos de Simulação Obtidos

Simulação	(conclusão)	
	Algoritmo Híbrido A* Tempo (s)	Algoritmo RRT Tempo (s)
3	6,415	5,519
4	6,884	4,451
5	7,123	3,942
6	7,193	6,407
7	7,147	4,438
8	8,991	7,026
9	7,806	4,513
10	11,168	6,679

Fonte: Autoria Própria.

Foi possível perceber uma variação a respeito do tempo de execução para os algoritmos. O Matlab® possui uma arquitetura aprimorada com relação ao tempo de compilação denominada *Just-in-Time* (JIT), essa arquitetura apresenta um desempenho melhor quando o código é executado mais vezes, fazendo com que exista um esforço para que a simulação ocorra de forma mais rápida (MATHWORKS, 2015a). Mesmo com a utilização do JIT e das técnicas impostas para que o programa apresentasse um desempenho melhor a cada execução, os tempos expressaram grandes oscilações. Uma comparação explicitada em forma gráfica com objetivo de analisar os tempos de forma diferente pode ser vista na Figura 37.

Figura 37 - Gráfico da Relação da Variação dos Tempos de Simulação para cada Algoritmo



Fonte: Autoria Própria.

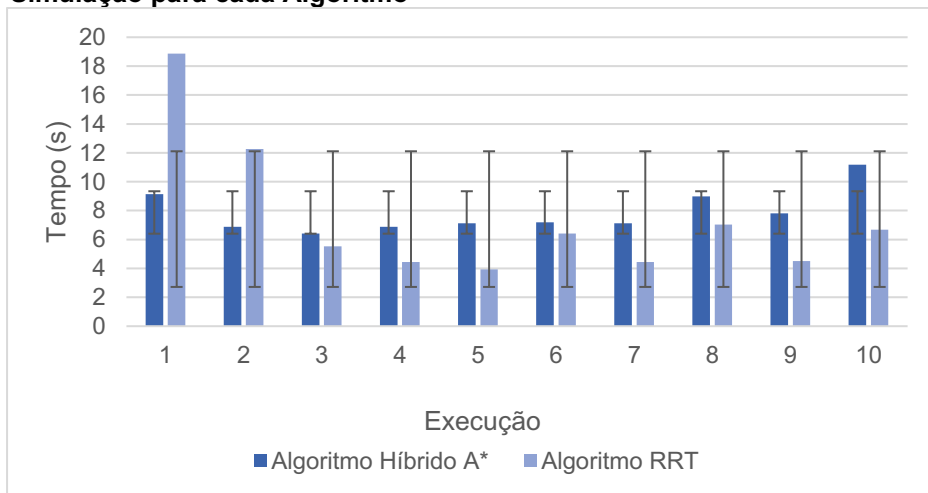
O desvio padrão também foi encontrado, e pode ser visualizado graficamente na Figura 38, considerando as equações (25) e (26).

$$DP = \sqrt{\frac{\sum_{s=1}^m \sum_{i=1}^n (y_{is} - M)^2}{(n_y - 1)}} \quad (25)$$

$$M = \frac{\sum_{s=1}^m \sum_{i=1}^n y_{is}}{n_y} \quad (26)$$

Em que s é o número da série, i representa o número de pontos na série s , m o número de séries para o ponto y no gráfico, n o número de pontos em cada série, y_{is} o valor de dados da série s e do ponto i , n_y o número total de valores de dados em todas as séries e M representa a média aritmética.

Figura 38 – Gráfico de Desvio Padrão da Relação da Variação dos Tempos de Simulação para cada Algoritmo



Fonte: Autoria Própria.

Com isso foi feito o cálculo da variância para avaliar esses tempos. Para um vetor com variáveis aleatórias A composto por N números escalares, a variância é definida na equação 27.

$$V = \frac{1}{N-1} \sum_{i=1}^N |A_i - m|^2 \quad (27)$$

Em que μ é a média de A .

$$m = \frac{1}{N} \sum_{i=1}^N A_i \quad (28)$$

Dessa forma, variância dos tempos obtida foram de 2,1510 (s) para o algoritmo Híbrido A* e de 22,042 (s) para o algoritmo RRT.

5.1 CONCLUSÃO

Este capítulo apresentou os resultados obtidos na simulação dos dois algoritmos utilizados para planejamentos de caminhos em veículos autônomos.

Entre os dois algoritmos utilizados para avaliação de desempenho, foi possível perceber que o algoritmo Híbrido A* apresentou melhor suavidade no deslocamento do ponto inicial ao final e menos incerteza com relação ao número de execuções do código criado pela autora. Essa afirmação se dá pelo fato de o algoritmo apresentar o mesmo caminho e os mesmos pontos nas dez execuções apresentadas. Por sua vez, o algoritmo RRT apresentou diferentes pontos e diferentes caminhos durante as execuções, e menor suavidade durante o caminho criado, visto que a incidência de curvas criadas pelos *Dubins* é maior que a apresentada no caminho do algoritmo Híbrido A*.

Utilizando o controlador Stanley para avaliar o comportamento do ângulo de esterçamento, foi possível perceber que o algoritmo RRT apresentou uma maior variação do ângulo, porém respeitou os limites impostos para o raio de giro mínimo do veículo. O caminho criado pelo algoritmo Híbrido A* mostrou maior suavidade e menor variação no ângulo de esterçamento, também respeitando os limites de raio de giro mínimo.

Os resultados obtidos para o tempo de execução de cada algoritmo foram avaliados sob aspectos estatísticos. Por mais que a ferramenta Matlab® apresente a arquitetura JIT para diminuir a oscilação do tempo em seguidas execuções, ocorreram mudanças consideráveis ao decorrer das simulações executadas. Apresentando um desvio padrão e uma variância para o tempo dos dois algoritmos.

6 CONCLUSÕES E PERSPECTIVAS

A dissertação em questão, apresentou como principais objetivos, o estudo de veículos autônomos, salientando seus principais componentes e a tarefa do planejamento de movimento com a avaliação de dois algoritmos. Essa tarefa é considerada fundamental para esses veículos, pois é responsável pelo início da etapa de deslocamento do ponto inicial ao ponto objetivo sem que ocorram acidentes.

Desta forma, esse trabalho apresentou um referencial teórico sobre os veículos autônomos abordando um breve histórico, seus principais componentes, as tarefas para que o deslocamento autônomo aconteça e também, abordou a utilização de dois algoritmos utilizados na tarefa de planejamento de movimento. O estudo desenvolvido nos capítulos iniciais foi de fundamental relevância para entender a dimensão dos sistemas presentes nos veículos autônomos. O investimento nas tecnologias desenvolvidas tanto nos sensores quanto nos sistemas embarcados é de grande importância para a empregabilidade nesses veículos.

Do estudo realizado no capítulo 3, pôde-se concluir a importância da confiabilidade que o veículo autônomo precisa para estimar a posição do ambiente em que se encontra. E até mesmo que a tarefa de deslocamento autônomo é bastante desafiadora, envolvendo tarefas de diferentes níveis e subprocessos. O controle físico do veículo precisa ser acompanhado e cumprir certos requisitos durante o planejamento de caminhos.

As análises do modelo físico do veículo, juntamente com as diferentes representações do modelo de estado do ambiente e dos algoritmos de busca de caminhos permitiram o desenvolvimento de um código em plataforma virtual. Por meio de métricas e funções específicas foi possível desenvolver um modelo de estado ambiente para representar um estacionamento e aplicar dois algoritmos diferentes para se deslocar de um ponto inicial até um ponto objetivo.

Com a interpretação dos dados obtidos durante as simulações, pode-se constatar a diferença no comportamento dos algoritmos, com relação a apresentação de diferentes pontos ao longo do caminho planejado e dos tempos de execução em cada simulação. O algoritmo Híbrido A* cumpriu com a suavidade do caminho a qual promete, e o algoritmo RRT encontrou um caminho até o ponto objetivo. Ambos cumprem os objetivos inicialmente propostos, encontrar um caminho livre de obstáculos de um ponto inicial a um ponto objetivo. Também foram avaliados distância

percorrida entre os caminhos encontrados pelos algoritmos e o comportamento do ângulo de esterçamento no deslocamento. Ambos algoritmos apresentaram o comportamento previsto, respeitando os limites impostos sobre o raio de giro mínimo, porém os caminhos encontrados pelo algoritmo RRT apresentaram maior variação dos ângulos.

Por fim, como trabalhos futuros, podem ser analisados outros algoritmos de busca de caminhos. Esses algoritmos podem ser combinados com sensores que auxiliem na criação de um modelo ambiente, como câmeras ou também sensores ultrassônicos proporcionando uma melhor estimativa dos obstáculos presentes no caminho. Além das análises sobre os caminhos gerados e sobre o tempo de simulação, pode ser avaliado também o perfil de velocidade para os diferentes caminhos encontrados em tempo real. Melhorando a visibilidade sobre a empregabilidade dos mesmos não apenas em ambiente simulado, mas no mundo real.

REFERÊNCIAS

ALAM, S. **Securing vehicle Electronic Control Unit (ECU) communications and stored data**. [S.l.]. 2018.

AZAM, S. et al. **Data fusion of Lidar and Thermal Camera for Autonomous driving**. Applied Industrial Optics: Spectroscopy, Imaging and Metrology. [S.l.]: [s.n.]. 2019. p. T2A--5.

BISHT, M.; ABBOTT, J.; GAFFAR, A. **Social dilemma of autonomous cars a critical analysis**. 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI). [S.l.]: [s.n.]. 2017. p. 1-3.

BRANCO, C. **2050. O ano de todas as certezas na condução autónoma**, 30 julho 2018. Disponível em: <<https://observador.pt/opiniaao/2050-o-ano-de-todas-as-certezas-na-conducao-autonoma/>>. Acesso em: abril fev. 2019.

BRECHTEL, S.; GINDELE, T.; DILLMANN, R. **Probabilistic MDP-behavior planning for cars**. 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC). [S.l.]: [s.n.]. 2011. p. 1537-1542.

BROWN, B. The social life of autonomous cars. **Computer**, p. 92-96, 2017.

BURKE, K. NVIDIA. **nvidia.com**, 2019. Disponível em: <<https://blogs.nvidia.com/blog/2019/04/15/how-does-a-self-driving-car-see/>>. Acesso em: 22 fevereiro 2020.

BUSCHKA, P.; SAFFIOTTI, A. **Some notes on the use of hybrid maps for mobile robots**. Proc. of the 8th Int. Conf. on Intelligent Autonomous Systems. [S.l.]: [s.n.]. 2004. p. 547-556.

CAMERON, O. **An Introduction to LIDAR: The Key Self-Driving Car Sensor**, 2017. Disponível em: <<https://news.voyage.auto/an-introduction-to-lidar-the-key-self-driving-car-sensor-a7e405590cff>>. Acesso em: 25 março 2019.

CAMPBELL, S. et al. **Sensor Technology in Autonomous Vehicles: A review**. 2018 29th Irish Signals and Systems Conference (ISSC). [S.l.]: IEEE. jun. 2018.

CHOSSET, H. M. et al. **Principles of robot motion: theory, algorithms, and implementation**. [S.l.]: MIT press, 2005.

CLAUSSMANN, L. et al. A review of motion planning for highway autonomous driving. **IEEE Transactions on Intelligent Transportation Systems**, v. 21, p. 1826-1848, 2019.

CONTINENTAL, M. Miles Continental, 2019. Disponível em: <<https://www.milescontinental.co.nz/news/features/volkswagen-park-assist-automatic-parking/>>. Acesso em: 10 março 2020.

CORMEN, T. H. et al. **Introduction to algorithms**. [S.l.]: MIT press, 2009.

DAVIES, A. How Do Self-Driving Cars See? (And How Do They See Me)?, 2018.

DICKMANN, J. et al. **"Automotive radar the key technology for autonomous driving: From detection and ranging to environmental understanding"**. 2016 IEEE Radar Conference (RadarConf). [S.l.]: IEEE. maio 2016.

DICKMANN, E. D. et al. **The seeing passenger car'VaMoRs-P'**. Proceedings of the Intelligent Vehicles' 94 Symposium. [S.l.]: [s.n.]. 1994. p. 68-73.

DIJKSTRA, E. W.; OTHERS. A note on two problems in connexion with graphs. **Numerische mathematik**, v. 1, p. 269-271, 1959.

DING, Y. Three Methods of Vehicle Lateral Control: Pure Pursuit, Stanley and MPC, 2020.

DOLGOV, D. et al. Path planning for autonomous vehicles in unknown semi-structured environments. **The International Journal of Robotics Research**, v. 29, p. 485-501, 2010.

DOUGLAS, B. Understanding Sensor Fusion and Tracking. **mathworks.com**, 2019. Disponível em: <<https://www.mathworks.com/videos/sensor-fusion-part-1-wt-sensor-fusion-1569410785813.html>>. Acesso em: 05 maio 2020.

DUDEK, G.; JENKIN, M. **Computational principles of mobile robotics**. [S.l.]: Cambridge university press, 2010.

DURRANT-WHYTE, H. A critical review of the state-of-the-art in autonomous land vehicle systems and technology. **Albuquerque (NM) and Livermore (CA), USA: Sandia National Laboratories**, v. 41, p. 242, 2001.

FILLIAT, D.; MEYER, J.-A. Map-based navigation in mobile robots: I. a review of localization strategies. **Cognitive Systems Research**, v. 4, p. 243-282, 2003.

GIESBRECHT, J. **Global path planning for unmanned ground vehicles**. DEFENCE RESEARCH AND DEVELOPMENT SUFFIELD (ALBERTA). [S.l.]. 2004.

GOCKLEY, R.; FORLIZZI, J.; SIMMONS, R. **Natural person-following behavior for social robots**. Proceedings of the ACM/IEEE international conference on Human-robot interaction. [S.l.]: [s.n.]. 2007. p. 17-24.

GONZÁLEZ, D. et al. A review of motion planning techniques for automated vehicles. **IEEE Transactions on Intelligent Transportation Systems**, v. 17, p. 1135-1145, 2015.

GU, T.; DOLAN, J. M.; LEE, J.-W. On-road trajectory planning for general autonomous driving with enhanced tunability. In: _____ **Intelligent Autonomous Systems 13**. [S.l.]: Springer, 2016. p. 247-261.

GUAPACHA, J. B. Localização e mapeamento simultâneos (SLAM) visual usando sensor RGB-D para ambientes internos e representação de características, 2017.

HART, P. E.; NILSSON, N. J.; RAPHAEL, B. A formal basis for the heuristic determination of minimum cost paths. **IEEE transactions on Systems Science and Cybernetics**, v. 4, p. 100-107, 1968.

HERGER, M. **The Last Driver License Holder**, 21 outubro 2017. Disponível em: <<https://thelastdriverlicenseholder.com/2017/10/21/overview-of-retrofit-kits-for-autonomous-driving/>>.

HOMANN, K. Wirtschaft und gesellschaftliche Akzeptanz: Fahrerassistenzsysteme auf dem Prüfstand. In: _____ **Fahrerassistenzsysteme mit maschineller Wahrnehmung**. [S.l.]: Springer, 2005. p. 239-244.

HORTON, M. 7 Reasons Your Life Depends on an Accurate IMU (Inertial Measurement Unit) in a Self-Driving Car. **Medium**, 2018. Disponível em: <<https://medium.com/@mikehorton/7-reasons-your-life-depends-on-an-accurate-imu-inertial-measurement-unit-in-a-self-driving-car-75298d5cff9e>>. Acesso em: 22 julho 2019.

JEON, D.; CHOI, H.; KIM, J. **UKF data fusion of odometry and magnetic sensor for a precise indoor localization system of an autonomous vehicle**. 2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI). [S.l.]: IEEE. ago. 2016.

JEON, J. et al. **Optimal motion planning with the half-car dynamical model for autonomous high-speed driving**. 2013 American control conference. [S.l.]: [s.n.]. 2013. p. 188-193.

KARAMAN, S.; FRAZZOLI, E. Sampling-based algorithms for optimal motion planning. **The international journal of robotics research**, v. 30, p. 846-894, 2011.

KATRAKAZAS, C. et al. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. **Transportation Research Part C: Emerging Technologies**, v. 60, p. 416-442, 2015.

KATZWINKEL, R. et al. Einparkassistenz. In: _____ **Handbuch Fahrerassistenzsysteme**. [S.l.]: Springer Fachmedien Wiesbaden, 2015. p. 841-850.

KELLY, J.; WASLANDER, S. **Lesson 2: The Inertial Measurement Unit (IMU)**, 2019. Disponível em: <<https://pt.coursera.org/lecture/state-estimation-localization-self-driving-cars/lesson-2-the-inertial-measurement-unit-imu-TBMU9>>. Acesso em: 07 abril 2019.

KONG, J. et al. **Kinematic and dynamic vehicle models for autonomous driving control design**. 2015 IEEE Intelligent Vehicles Symposium (IV). [S.l.]: [s.n.]. 2015. p. 1094-1099.

LATOMBE, J.-C. **Robot motion planning**. [S.l.]: Springer Science & Business Media, v. 124, 2012.

LEE, J.; LITKOUHI, B. **A unified framework of the automated lane centering/changing control for motion smoothness adaptation**. 2012 15th International IEEE Conference on Intelligent Transportation Systems. [S.l.]: [s.n.]. 2012. p. 282-287.

LEFEVRE, S.; CARVALHO, A.; BORRELLI, F. **Autonomous car following: A learning-based approach**. 2015 IEEE Intelligent Vehicles Symposium (IV). [S.l.]: IEEE. jun. 2015.

LEISS, P. J. **The Functional Components of Autonomous Vehicles**. <https://www.robsonforensic.com>, 2018. Disponível em: <<https://www.robsonforensic.com/articles/autonomous-vehicles-sensors-expert/>>. Acesso em: 05 maio 2020.

LI, N. et al. Game Theoretic Modeling of Driver and Vehicle Interactions for Verification and Validation of Autonomous Vehicle Control Systems. **IEEE Transactions on Control Systems Technology**, v. 26, p. 1782-1797, 2016.

LIDAR UK. **How does LiDAR work? - The science behind the technology**, 2019. Disponível em: <<http://www.lidar-uk.com/how-lidar-works/>>. Acesso em: 12 março 2019.

LIKHACHEV, M.; GORDON, G. J.; THRUN, S. ARA*: Anytime A* with provable bounds on sub-optimality. **Advances in neural information processing systems**, v. 16, p. 767-774, 2003.

LIM, W. et al. Hierarchical trajectory planning of an autonomous car based on the integration of a sampling and an optimization method. **IEEE Transactions on Intelligent Transportation Systems**, v. 19, p. 613-626, 2018.

LIN, S.-C. et al. **The architectural implications of autonomous driving: Constraints and acceleration**. Proceedings of the Twenty-Third International Conference on

Architectural Support for Programming Languages and Operating Systems. [S.l.]: [s.n.]. 2018. p. 751-766.

LUO, Q. et al. Localization and navigation in autonomous driving: Threats and countermeasures. **IEEE Wireless Communications**, v. 26, p. 38-45, 2019.

LYNCH, K. M.; PARK, F. C. **Modern Robotics**. [S.l.]: Cambridge University F 2017.

MATHWORKS. **MATLAB Execution Engine**, 2015a. Disponível em: <<https://www.mathworks.com/products/matlab/matlab-execution-engine.html>>. Acesso em: 22 novembro 2020.

MATHWORKS. **Lateral Controller Stanley**, 2018f. Disponível em: <https://www.mathworks.com/help/driving/ref/lateralcontrollerstanley.html#mw_160e9ca0-d791-4542-8685-77b52dbb6161>. Acesso em: 15 dezembro 2020.

MATHWORKS. **inflationCollisionChecker**, 2018a. Disponível em: <https://www.mathworks.com/help/driving/ref/driving.costmap.inflationcollisionchecker.html?searchHighlight=CollisionChecker&s_tid=srchtitle>. Acesso em: 14 setembro 2020.

MATHWORKS. **vehicleDimensions**, 2018c. Disponível em: <https://www.mathworks.com/help/driving/ref/vehicledimensions.html?searchHighlight=vehicledimensions&s_tid=srchtitle>. Acesso em: 13 dezembro 2020.

MATHWORKS. **vehicleCostmap**, 2018b. Disponível em: <https://www.mathworks.com/help/driving/ref/vehiclecostmap.html?searchHighlight=costmap&s_tid=srchtitle>. Acesso em: 13 setembro 2020.

MATHWORKS. **plannerHybridAStar**, 2019b. Disponível em: <<https://www.mathworks.com/help/nav/ref/plannerhybridastar.html#d122e78449>>. Acesso em: 21 setembro 2020.

MATHWORKS. **plannerRRT**, 2019d. Disponível em: <https://www.mathworks.com/help/nav/ref/plannerrrt.html?searchHighlight=plannerRRT&s_tid=srchtitle>. Acesso em: 22 setembro 2020.

MATHWORKS. **stateSpaceDubins**, 2019d. Disponível em: <<https://www.mathworks.com/help/nav/ref/statespacedubins.html>>. Acesso em: 22 setembro 2020.

MATHWORKS. **Choose Path Planning Algorithms for Navigation**, 2020a. Disponível em: <https://www.mathworks.com/help/nav/ug/choose-path-planning-algorithms-for-navigation.html?s_tid=srchtitle>. Acesso em: 10 outubro 2020.

MATHWROKS. **reedsSheppConnection**, 2019a. Disponível em: <<https://www.mathworks.com/help/nav/ref/reedssheppconnection.html>>. Acesso em: 20 setembro 2020.

MATWORKS. **stateSpaceSE2**, 2019c. Disponível em: <<https://www.mathworks.com/help/nav/ref/statespace2.html>>. Acesso em: 22 setembro 2020.

MCNAUGHTON, M. et al. **Motion planning for autonomous driving with a conformal spatiotemporal lattice**. 2011 IEEE International Conference on Robotics and Automation. [S.l.]: [s.n.]. 2011. p. 4889-4895.

MEIRA, G. T. Stereo Vision-based Autonomous Vehicle Navigation, 2016.

MURPHY, K. N. **Analysis of robotic vehicle steering and controller delay**. Fifth International Symposium on Robotics and Manufacturing (ISRAM). [S.l.]: [s.n.]. 1994. p. 631-636.

MURPHY, L.; NEWMAN, P. **Risky planning**: Path planning over costmaps with a probabilistically bounded speed-accuracy tradeoff. 2011 IEEE International Conference on Robotics and Automation. [S.l.]: [s.n.]. 2011. p. 3727-3732.

NASCIMENTO, L. B. P. et al. Planejamento de Caminho para Sistemas Robóticos Autônomos, 2019.

NHTSA. Preliminary Statement of Policy Concerning Automated Vehicles, 2012. Disponível em: <https://orfe.princeton.edu/~alaink/SmartDrivingCars/CommentOnNHTSA_PrelimStatement.pdf>. Acesso em: 7 março 2020.

NHTSA. Automated Vehicles for Safety, 2019. Disponível em: <<https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety>>. Acesso em: dezembro 2019.

NOREEN, I.; KHAN, A.; HABIB, Z. A comparison of RRT, RRT* and RRT*-smart path planning algorithms. **International Journal of Computer Science and Network Security (IJCSNS)**, v. 16, p. 20, 2016.

ODUNLADE, E. Circuit Digest. **circuitdigest.com**, 2019. Disponível em: <<https://circuitdigest.com/article/what-is-lidar-and-how-does-lidar-works>>. Acesso em: 20 dezembro 2019..

ONU. Organização das Nações Unidas, 17 janeiro 2018. Disponível em: <<https://nacoesunidas.org/acidentes-de-transito-matam-125-milhao-de-pessoas-no-mundo-por-ano/>>.

OTTONI, G. D. L. Planejamento de trajetórias para robôs móveis. **Projeto Graduação em Engenharia de Computação--FURG, Rio Grande**, 2000.

PADEN, B. et al. A survey of motion planning and control techniques for self-driving urban vehicles. **IEEE Transactions on intelligent vehicles**, v. 1, p. 33-55, 2016.

PARAKKAL, P. G.; VARIYAR, V. V. S. **Gps based navigation system for autonomous car**. 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). [S.l.]: IEEE. set. 2017.

PARK, W.-J. et al. **Parking space detection using ultrasonic sensor in parking assistance system**. 2008 IEEE Intelligent Vehicles Symposium. [S.l.]: IEEE. jun. 2008.

PARKINSON, S. et al. Cyber Threats Facing Autonomous and Connected Vehicles: Future Challenges. **IEEE Transactions on Intelligent Transportation Systems**, v. 18, p. 2898-2915, nov. 2017.

GUAPACHA PINTO, R. M.; PORTO, A. J. V. Geração e controle de trajetória ponto a ponto para veículos de grande porte agrícolas, 2018.

PISSARDINI, R. D. S. **Veículos autônomos de transporte terrestre: proposta de arquitetura de tomada de decisão para navegação autônoma**. Universidade de São Paulo. [S.l.]. 2014.

RAFFO, G. V.; OTHERS. Algoritmos de controle preditivo para seguimento de trajetórias de veículos autônomos, 2005.

RAJAMANI, R. **Vehicle dynamics and control**. [S.l.]: Springer Science & Business Media, 2011.

RAJASEKHAR, M. V.; JASWAL, A. K. Autonomous vehicles: The future of automobiles. **2015 IEEE International Transportation Electrification Conference (ITEC)**, 2016.

REALTIME Robotics. **Autonomous Driving and the Need For Motion Planning**, 2019. Disponível em: <<https://rtr.ai/on-going-research/autonomous->

REEDS, J.; SHEPP, L. Optimal paths for a car that goes both forwards and backwards. **Pacific journal of mathematics**, v. 145, p. 367-393, 1990.

RODRIGUES, L. C. **Fundamentos, tecnologias e aplicações de veículos autônomos**. Universidade Tecnológica Federal do Paraná. [S.l.]. 2017.

RODRIGUES, M. et al. **Adaptive tactical behaviour planner for autonomous ground vehicle**. 2016 UKACC 11th International Conference on Control (CONTROL). [S.l.]: [s.n.]. 2016. p. 1-8.

RUDOLPH, G.; VOELZKE, U. Three Sensor Types Drive Autonomous Vehicles, 2017.

SANTOS, A. S. Planejamento de caminhos para robôs móveis em ambientes acidentados., 2019.

SIEGWART, R.; NOURBAKHSH, I. R.; SCARAMUZZA, D. **Introduction to autonomous mobile robots**. [S.l.]: MIT press, 2011.

SOUSA PISSARDINI, R.; WEI, D. C. M.; FONSECA JÚNIOR, E. S. **Veículos Autônomos: conceitos, histórico e estado-da-arte**. Anais do XXVII Congresso de Pesquisa e Ensino em Transportes--ANPET. [S.l.]: [s.n.]. 2013.

STRICKLAND, L. Racing Team's enhanced Touareg makes it to semifinals of robotic car race. **Stanford**, 2005. Disponível em: <<https://news.stanford.edu/news/2005/july13/carup-071305.html>>. Acesso em: 20 maio 2019.

SURRECIO, A.; NUNES, U.; ARAUJO, R. **Fusion of Odometry with Magnetic Sensors Using Kalman Filters and Augmented System Models for Mobile Robot Navigation**. Proceedings of the IEEE International Symposium on Industrial Electronics, 2005. ISIE 2005. [S.l.]: IEEE. 2005.

TAHA, A.-E.; ABUALI, N. Route planning considerations for autonomous vehicles. **IEEE Communications Magazine**, v. 56, p. 78-84, 2018.

THORPE, C. et al. Vision and navigation for the Carnegie-Mellon Navlab. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 10, p. 362-373, 1988.

THRUN, S. Robotic mapping: A survey. **Exploring artificial intelligence in the new millennium**, v. 1, p. 1, 2002.

THRUN, S. et al. Stanley: The robot that won the DARPA Grand Challenge. **Journal of field Robotics**, v. 23, p. 661-692, 2006.

TONG, Z.; RENTER, R.; FUJIMOTO, M. **Fast chirp FMCW radar in automotive applications**. IET International Radar Conference 2015. [S.l.]: [s.n.]. 2015. p. 1-4.

TRANSPORTATION, U. S. D. O. Automated Vehicles for Safety. **NHTSA**, 2019. Disponível em: <<https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety>>. Acesso em: 30 julho 2019.

URMSON, C. et al. Tartan racing: A multi-modal approach to the darpa urban challenge, 2007.

VARAIYA, P. Smart cars on smart roads: problems of control. **IEEE Transactions on automatic control**, v. 38, p. 195-207, 1993.

VELOSO, P. H. O. Desenvolvimento de um protótipo para veículos autônomos com localização e mapeamento simultâneos. 2019.

VIRDI, J. Motion Planning - Finding smooth optimal paths using Hybrid A*/Astar for autonomous vehicles, 2020. Disponível em: <<https://youtube-dl-mp4.blogspot.com/search/label/QAzyG2DB8lo>>. Acesso em: 25 julho 2020.

VOJTECH, V. et al. RRT-Path: a guided Rapidly Exploring Random tree. **Robot motion and control, Poznan, Poland, 2009.**

WASLANDER, S.; KELLY, J. **Lesson 2:** The Inertial Measurement Unit (IMU), 2018a. Disponível em: <<https://pt.coursera.org/lecture/state-estimation-localization-self-driving-cars/lesson-2-the-inertial-measurement-unit-imu-TBMU9>>. Acesso em: 22 agosto 2019.

WASLANDER, S.; KELLY, J. **Lesson 2:** Motion Planning Constraints, 2018b. Disponível em: <<https://pt.coursera.org/lecture/motion-planning-self-driving-cars/lesson-2-motion-planning-constraints-UejAh>>. Acesso em: 18 setembro 20

WASLANDER, S.; KELLY, J. **Lesson 3:** Objective Functions for Autonomous Driving, 2018c. Disponível em: <<https://pt.coursera.org/lecture/motion-planning-self-driving-cars/lesson-3-objective-functions-for-autonomous-driving-mT246>>. Acesso em: 20 setembro 2019.

WASLANDER, S.; KELLY, J. **Lesson 3:** The Global Navigation Satellite Systems (GNSS), 2018d. Disponível em: <<https://pt.coursera.org/lecture/state-estimation-localization-self-driving-cars/lesson-3-the-global-navigation-satellite-systems-gnss-TDInE>>. Acesso em: 20 setembro 2019.

WETMORE, J. Driving the dream. The history and motivations behind 60 years of automated highway systems in America. **Automotive History Review**, v. 7, p. 4-19, 2003.

XU, W. et al. Analyzing and Enhancing the Security of Ultrasonic Sensors for Autonomous Vehicles. **IEEE Internet of Things Journal**, v. 5, p. 5015-5029, dez. 2018.

ZANCHIN, B. C. **Análise do algoritmo A*(a estrela) no planejamento de rotas de veículos autônomos.** Universidade Tecnológica Federal do Paraná. [S.l.]. 2018.

ZIEGLER, J. et al. Making bertha drive—an autonomous journey on a historic route. **IEEE Intelligent transportation systems magazine**, v. 6, p. 8-20, 2014.

APÊNDICE A – CÓDIGO UTILIZADO PARA O ALGORITMO HÍBRIDO A*

```

% Criação de uma matriz com valores binários
p = [...
1  1 1 1 1 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
1
1  0 0 0 0 0  1  0  0  0  1  0  0  0  1  0  0  0  1  0  0
0  1  0  0  0  0  1  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0
1
1  0 0 0 0 0  1  0  0  0  1  0  0  0  1  0  0  0  1  0  0
0  1  0  0  0  0  1  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0
1
1  0 0 0 0 0  1  0  0  0  1  0  0  0  1  0  0  0  1  0  0
0  1  0  0  0  0  1  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0
1
1  0 0 0 0 0  1  0  0  0  1  0  0  0  1  0  0  0  1  0  0
0  1  0  0  0  0  1  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0
1
1  0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
1
1  0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
1
1  0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
1
1  0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
1
1  0 0 0 0 0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0
1  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0
1
1  0 0 0 0 0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  1
0  0  0  0  1  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0
1
1  0 0 0 0 0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  1  0
0  0  0  1  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0
1
1  0 0 0 0 0  0  0  1  0  0  0  0  1  0  0  0  0  1  0  0
0  0  1  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0  0  0
1

```

```

1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0
0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0
1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1
1 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0
0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0
1
1 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0
0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0
1
1 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0
0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0
1
1 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0
0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0
1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1
];
%Transformação da matriz em um mapa com valores lógicos
simpleMap = logical(p);
%Criação de um Mapa de Ocupação Binário
refMap = binaryOccupancyMap(simpleMap,1);
%Apresentação do Mapa de Ocupação Binário
show(refMap);
%Validando o caminho através de um mapa de custos do veículo
map = vehicleCostmap(double(simpleMap));
%Criando uma Região para Checagem de Colisão
map.CollisionChecker = inflationCollisionChecker("InflationRadius",1);
%Plotando o Mapa com as Áreas Infladas que não podem ser ocupadas pelo
%veículo
plot(map)
validator = validatorVehicleCostmap(stateSpaceSE2);
validator.Map = map;

```

```
validator.ValidationDistance = 0.1;
%Utilização do algoritmo para planejamento de movimento
planner =
plannerHybridAStar(validator, 'MinTurningRadius',4, 'MotionPrimitiveLength',4
);
%Definição das posições inicial e objetivo [metros,metros,graus]
startPose = [4 10 0];
goalPose = [33 26.25 90];
%Execução de um caminho de referência
refpath = plan(planner,startPose,goalPose);
figure
%Apresentação do caminho criado pelo Algoritmo entre o ponto inicial e o
ponto objetivo
show(planner)
```

APÊNDICE B – CÓDIGO UTILIZADO PARA O ALGORITMO RRT

```

% Criação de uma matriz com valores binários
p = [...
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1
1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0
0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0
1
1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0
0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0
1
1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0
0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0
1
1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0
0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0
1
1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0
0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0
1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1
1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0
1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0
1
1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1
0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0
1
1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0
0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0
1
1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0
0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0
1

```

```

1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0
0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0
1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1
1 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0
0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0
1
1 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0
0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0
1
1 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0
0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0
1
1 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0
0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0
1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1
];
%Transformação da matriz em um mapa com valores lógicos
simpleMap = logical(p);
%Criação de um Mapa de Ocupação Binário
refMap = binaryOccupancyMap(simpleMap,1);
refFigure = figure('Name','SimpleMap');
figure
%Apresentação do Mapa de Ocupação Binário
show(refMap);
%Validação do caminho através do ambiente utilizando mapa de custos do
veículo
map = vehicleCostmap(double(simpleMap));
%Criando uma Região para Checagem de Colisão
map.CollisionChecker = inflationCollisionChecker("InflationRadius",1);
figure
%Plotando o Mapa com as Áreas Infladas que não podem ser ocupadas pelo

```

```
%veículo
plot(map)
% Definição das posições inicial e final
startPose = [4, 10, 0]; % [meters, meters, degrees]
goalPose = [33,26.25,90];
%Aplicação do algoritmo no mapa de custos (costmap)
planner = pathPlannerRRT(map);
%Execução de um caminho de referência
refPath = plan(planner,startPose,goalPose);
figure
plot(planner)
hold on
%Apresentação do caminho criado pelo Algoritmo entre o ponto inicial e o
ponto objetivo
plot(refPath);
hold off
```