

**Luciano Moura Gonçalves**

***Sistema de Navegação de Veículo Autônomo  
Inteligente***

Itajubá - MG

Abril de 2010

**Luciano Moura Gonçalves**

***Sistema de Navegação de Veículo Autônomo  
Inteligente***

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica como parte dos requisitos para obtenção do Título de Mestre em Ciências em Engenharia Elétrica.

Área de Concentração: Automação e Sistemas Elétricos de Potência

Orientador: Prof. Dr. Leonardo de Mello Honório

Orientador:

Prof. Dr. Leonardo de Mello Honório

UNIVERSIDADE FEDERAL DE ITAJUBÁ  
PROGRAMA DE PÓS-GRADUAÇÃO EM  
ENGENHARIA ELÉTRICA

Itajubá - MG

Abril de 2010

# *Epígrafe*

*”Os problemas significativos com os quais nos deparamos não podem ser resolvidos no mesmo nível de pensamento em que estávamos quando eles foram criados.”*

Albert Einstein

# *Dedicatória e agradecimentos*

Dedico este trabalho a minha família pelo suporte e compreensão que sempre me dedicaram, cada conquista minha é também uma conquista de cada um deles.

Agradeço ao meu orientador, meus colegas e amigos que contribuíram com este trabalho, em cada detalhe existe a ajuda de alguns deles.

A elaboração deste trabalho foi realizada com o uso de *softwares* livres, entre eles:

- GNU-Linux Ubuntu
- Inkscape
- Kile
- $\text{\LaTeX}$
- $\text{\ABNT\TeX}$
- Asymptote

Este trabalho foi realizado com auxílio financeiro da CAPES.

# *Resumo*

Neste trabalho são desenvolvidos alguns componentes de um sistema de navegação para um veículo autônomo inteligente. Foram desenvolvidos alguns recursos necessários para o sistema de navegação cumprir os objetivos propostos nesse trabalho, esses recursos são um modelo cinemático do veículo, uma arquitetura de algoritmo de busca e um sistema de planificação de coordenadas globais.

O modelo cinemático descreve o comportamento da movimentação horizontal do veículo para determinadas variáveis de entrada, baseado nas características do veículo real. O algoritmo de busca é responsável por determinar uma sequência de ações necessária para alcançar algum objetivo estabelecido. O sistema de planificação é necessário para determinar uma nova representação para o posicionamento global do veículo.

Um simulador é desenvolvido utilizando o modelo cinemático e seus parâmetros são otimizados através de algoritmo genético que tem como função de avaliação a diferença de comportamento do veículo simulado em relação ao comportamento do veículo real para as mesmas variáveis de entrada. O modelo desenvolvido é avaliado e validado através dessa diferença de comportamento. Uma vez com o *software* de simulação desenvolvido, ele é utilizado para validar grande parte do desenvolvimento do sistema de navegação autônomo inteligente.

Uma arquitetura híbrida de sistema de controle é desenvolvida, onde controladores são modelados para tratarem problemas específicos da navegação do veículo autônomo inteligente, tendo acesso aos recursos disponibilizados pelo sistema de navegação.

**Palavras-chave:** Veículo Autônomo Inteligente, Sistema de Navegação, Algoritmo de Busca, Planificação de Coordenadas, Modelo Cinemático, Simulador.

# *Abstract*

This work we developed some components of a navigation system for an intelligent autonomous vehicle. We developed some resources to the navigation system to meet the objectives proposed in this work, these resources are a kinematic model of the vehicle, an architecture of search algorithm and a planned system of coordinates.

The dynamic model describes the behavior of the horizontal movement of the vehicle for certain input variables, based on the characteristics of the actual vehicle. The search algorithm is responsible for determining a sequence of actions needed to achieve any goal set. The planning system is needed to determine a new representation for the global positioning of the vehicle.

A simulator is developed using the dynamic model and its parameters are optimized through genetic algorithm that has as its assessment of the difference simulated vehicle behavior in relation to the conduct of the actual vehicle for the same input variables. The model is evaluated and validated through the evaluation of this difference in behavior. Once the simulation software developed, it is used to validate much of the development of autonomous navigation system intelligent.

A hybrid architecture of control system is developed, where drivers are modeled to deal with specific problems of intelligent autonomous vehicle navigation, and access to resources provided by the navigation system.

**Keywords:** Intelligent Autonomous Vehicle, Navigation System, Search Algorithm, Planned System of Coordinates, Kinematic Model, Simulator.

# *Sumário*

**Lista de Figuras**

**Lista de Tabelas**

**Lista de abreviaturas e siglas**

**Lista de símbolos**

<b>1</b>	<b>Introdução</b>	p. 16
1.1	Motivação . . . . .	p. 16
1.2	Estado da Arte . . . . .	p. 17
1.3	Objetivos . . . . .	p. 18
<b>2</b>	<b>Veículo Autônomo Inteligente</b>	p. 20
2.1	Introdução . . . . .	p. 20
2.2	Considerações iniciais . . . . .	p. 20
2.3	Estrutura do trabalho . . . . .	p. 21
2.3.1	Introdução . . . . .	p. 21
2.3.2	Sistema de Navegação . . . . .	p. 22
2.3.3	Veículo Automatizado . . . . .	p. 24
2.3.4	Ambiente . . . . .	p. 25
<b>3</b>	<b>Recursos Compartilhados</b>	p. 27
3.1	Introdução . . . . .	p. 27
3.2	Modelo Cinemático . . . . .	p. 27

3.2.1	Introdução . . . . .	p. 27
3.2.2	Posicionamento e orientação . . . . .	p. 28
3.2.3	Modelo Desenvolvido . . . . .	p. 29
3.3	Planificação de Coordenadas . . . . .	p. 33
3.3.1	Introdução . . . . .	p. 33
3.3.2	Posicionamento Global . . . . .	p. 34
3.3.3	Metodologia . . . . .	p. 36
3.4	Algoritmo de Busca . . . . .	p. 36
3.4.1	Introdução . . . . .	p. 36
3.4.2	Busca em Árvore . . . . .	p. 37
3.4.3	Metodologia . . . . .	p. 38
3.4.4	A* (A Estrela) . . . . .	p. 39
<b>4</b>	<b>Arquitetura do Sistema de Controle</b>	<b>p. 42</b>
4.1	Introdução . . . . .	p. 42
4.2	Camada Coordenadora . . . . .	p. 43
4.3	Camada Deliberativa . . . . .	p. 44
4.4	Camada Reativa . . . . .	p. 44
4.5	Camada Executiva . . . . .	p. 45
<b>5</b>	<b>Simulador</b>	<b>p. 46</b>
5.1	Introdução . . . . .	p. 46
5.2	Estrutura do Simulador . . . . .	p. 46
5.2.1	Introdução . . . . .	p. 46
5.2.2	Algoritmo . . . . .	p. 47
5.3	Otimização . . . . .	p. 50
5.3.1	Introdução . . . . .	p. 50

5.3.2	Metodologia . . . . .	p. 51
5.3.3	Estrutura do Otimizador . . . . .	p. 51
5.3.4	Resultados da Otimização . . . . .	p. 56
5.4	Resultados . . . . .	p. 57
<b>6</b>	<b>Controladores Desenvolvidos</b>	<b>p. 62</b>
6.1	Introdução . . . . .	p. 62
6.2	Navegador Global . . . . .	p. 63
6.2.1	Introdução . . . . .	p. 63
6.2.2	Modelagem . . . . .	p. 63
6.2.3	Operador $\Gamma$ . . . . .	p. 64
6.2.4	Operador $\Omega$ . . . . .	p. 65
6.2.5	Operador $\Lambda$ . . . . .	p. 66
6.2.6	Custo $\hat{g}$ . . . . .	p. 66
6.2.7	Custo $\hat{h}$ . . . . .	p. 66
6.2.8	Resultados . . . . .	p. 67
6.3	Navegador Local . . . . .	p. 70
6.3.1	Introdução . . . . .	p. 70
6.3.2	Modelagem . . . . .	p. 71
6.3.3	Operador $\Gamma$ . . . . .	p. 72
6.3.4	Operador $\Omega$ . . . . .	p. 73
6.3.5	Operador $\Lambda$ . . . . .	p. 73
6.3.6	Custo $\hat{g}$ . . . . .	p. 74
6.3.7	Custo $\hat{h}$ . . . . .	p. 75
6.3.8	Busca . . . . .	p. 76
6.3.9	Resultados . . . . .	p. 77
6.4	Seguidor de Estados . . . . .	p. 80

6.4.1	Introdução . . . . .	p. 80
6.4.2	Metodologia . . . . .	p. 80
6.4.3	Resultados . . . . .	p. 81
<b>7</b>	<b>Conclusões e propostas para trabalhos futuros</b>	<b>p. 85</b>
7.1	Introdução . . . . .	p. 85
7.2	Recursos Compartilhados . . . . .	p. 85
7.2.1	Introdução . . . . .	p. 85
7.2.2	Modelo Cinemático . . . . .	p. 86
7.2.3	Planificação de Coordenadas . . . . .	p. 86
7.2.4	Algoritmo de Busca . . . . .	p. 87
7.3	Sistema de Controle . . . . .	p. 87
7.4	Simulador . . . . .	p. 87
7.5	Controladores . . . . .	p. 88
7.5.1	Introdução . . . . .	p. 88
7.5.2	Navegador Global . . . . .	p. 88
7.5.3	Navegador Local . . . . .	p. 88
7.5.4	Seguidor de Estados . . . . .	p. 89
	<b>Referências Bibliográficas</b>	<b>p. 90</b>

# *Lista de Figuras*

2.1	Estrutura do <i>Veículo Autônomo Inteligente</i> . . . . .	p. 22
2.2	Estrutura do <i>Sistema de Navegação</i> . . . . .	p. 23
2.3	Estrutura do <i>Veículo Automatizado</i> . . . . .	p. 25
3.1	Posicionamento do veículo . . . . .	p. 29
3.2	Modelo de Veículo Bicíclico . . . . .	p. 30
3.3	Modelo do Veículo . . . . .	p. 32
3.4	Sistema de Coordenadas ECEF . . . . .	p. 35
3.5	Árvore de busca . . . . .	p. 37
3.6	Diagrama Simplificado da Classe <i>Nó</i> . . . . .	p. 39
3.7	Algoritmo de busca A* . . . . .	p. 41
4.1	Estrutura do <i>Sistema de Controle</i> . . . . .	p. 42
5.1	Estrutura do <i>Simulador</i> . . . . .	p. 47
5.2	Diagrama de Classes Simplificado . . . . .	p. 48
5.3	Diagrama de Classes simplificado dos sensores . . . . .	p. 49
5.4	Método de Ler Sensores . . . . .	p. 49
5.5	Diagrama de Classes Simplificado dos Atuadores . . . . .	p. 50
5.6	Estrutura da Metodologia de Otimização . . . . .	p. 51
5.7	Estrutura do Otimizador . . . . .	p. 52
5.8	Estrutura da Função de Avaliação . . . . .	p. 52
5.9	Estrutura do Algoritmo Genético . . . . .	p. 54
5.10	Processo de Cruzamento . . . . .	p. 55
5.11	Processo de Mutação . . . . .	p. 55

5.12	Base de Dados para Otimização . . . . .	p. 57
5.13	Resultado da Otimização . . . . .	p. 58
5.14	Circuito . . . . .	p. 59
5.15	Comportamento do Veículo . . . . .	p. 60
5.16	Comportamento real do Veículo . . . . .	p. 61
6.1	Diagrama de Classes simplificado . . . . .	p. 64
6.2	Ligação entre <i>Sub Circuitos</i> . . . . .	p. 65
6.3	Área mapeada para o <i>Navegador Global</i> . . . . .	p. 67
6.4	<i>Circuito</i> resultante da área mapeada para o <i>Navegador Global</i> . . . . .	p. 68
6.5	Busca de rota sendo executada pelo <i>Navegador Global</i> . . . . .	p. 69
6.6	Rota resultante da busca executada pelo <i>Navegador Global</i> . . . . .	p. 69
6.7	Rota replanejada pelo <i>Navegador Global</i> . . . . .	p. 70
6.8	Raio de curvatura e <i>Célula</i> . . . . .	p. 72
6.9	<i>Células</i> . . . . .	p. 72
6.10	Custo $\hat{h}$ para o <i>Navegador Local</i> . . . . .	p. 75
6.11	Busca de rota sendo executada pelo <i>Navegador Local</i> . . . . .	p. 76
6.12	Busca de rota executada pelo <i>Navegador Local</i> . . . . .	p. 77
6.13	Resultado do primeiro exemplo do <i>Navegador Local</i> . . . . .	p. 78
6.14	Resultado do segundo exemplo do <i>Navegador Local</i> . . . . .	p. 79
6.15	Resultado do terceiro exemplo do <i>Navegador Local</i> . . . . .	p. 79
6.16	Resultado do primeiro exemplo do <i>Seguidor de Estados</i> . . . . .	p. 81
6.17	Resultado do segundo exemplo do <i>Seguidor de Estados</i> . . . . .	p. 83
6.18	Resultado do terceiro exemplo do <i>Seguidor de Estados</i> . . . . .	p. 84

## *Lista de Tabelas*

- 6.1 Tabela de Resultados do primeiro exemplo do *Seguidor de Estados* . . . . . p. 82
- 6.2 Tabela de Resultados do segundo exemplo do *Seguidor de Estados* . . . . . p. 82
- 6.3 Tabela de Resultados do terceiro exemplo do *Seguidor de Estados* . . . . . p. 83

## *Lista de abreviaturas e siglas*

ABS	Sistema de anti bloqueio de frenagem (Anti-lock brake systems),	p. 17
ASCS	Sistemas de controle de estabilidade ativa (Active Stability Control Systems),	p. 17
UNIFEI	Universidade Federal de Itajubá,	p. 29
CRTi	Centro de Referência em Tecnologia da Informação,	p. 29
GPS	Sistema de Posicionamento Global (Global Position System),	p. 34
DGPS	GPS Diferencial (Differential GPS),	p. 34
ECEF	Centrado na Terra, fixado na Terra (Earth-centered, Earth-fixed),	p. 34
MGS84	World Geodetic System of 1984,	p. 35

## *Lista de símbolos*

$X$	Eixo horizontal de coordenadas globais,	p. 28
$Y$	Eixo vertical de coordenadas globais,	p. 28
$x_i$	Eixo longitudinal do veículo,	p. 28
$y_j$	Eixo lateral do veículo,	p. 28
$\vec{P}$	Vetor posição do Veículo,	p. 28
$P_x$	Posição no eixo $X$ ,	p. 28
$P_y$	Posição no eixo $Y$ ,	p. 28
$\psi$	Ângulo formado entre os eixos $X$ e $x_i$ ,	p. 28
$\delta$	Ângulo da roda dianteira do veículo,	p. 29
$\vec{V}$	Vetor de velocidade do veículo,	p. 30
$\beta$	Ângulo formado entre $\vec{V}$ e $x_i$ ,	p. 30
$\dot{\psi}$	Variação do ângulo $\psi$ em relação ao tempo,	p. 31
$\dot{X}$	Variação da posição do veículo com relação ao tempo no eixo $X$ ,	p. 31
$\dot{Y}$	Variação da posição do veículo com relação ao tempo no eixo $Y$ ,	p. 31
$\phi$	Ângulo de Latitude,	p. 34
$L$	Ângulo de Longitude,	p. 34
$a$	Semi-eixo maior da elipsóide,	p. 34
$b$	Semi-eixo menor da elipsóide,	p. 34
$f$	Fator de achatamento da elipsóide,	p. 35
$D_{xy}(\phi, L)$	Operador de planificação das coordenadas globais,	p. 36
$C_{\phi, L}(P_x, P_y)$	Operador de planificação das coordenadas globais,	p. 36
$\Lambda$	Operador responsável por determinar quando um nó será considerado igual ao nó atual,	p. 38
$\Omega$	Operador que indica se o nó atual é um dos nós de destino,	p. 38
$\Gamma$	Operador que determina os nós sucessores ao nó atual,	p. 38
$A^*$	Algoritmo de busca A Estrela,	p. 39
$f(n)$	Custo de um caminho ótimo indo do nó inicial a um nó objetivo passando pelo nó $n$ ,	p. 40

$\hat{f}(n)$	Custo estimado de um caminho ótimo indo do nó inicial a um nó objetivo passando pelo nó $n$ ,	p. 40
$g(n)$	Custo atual do caminho ótimo do nó inicial ao nó $n$ ,	p. 40
$h(n)$	Custo atual do caminho ótimo do nó $n$ a um nó objetivo,	p. 40
$\hat{g}(n)$	Custo do caminho ótimo encontrado pelo algoritmo do nó inicial ao nó $n$ ,	p. 40
$\hat{h}(n)$	Custo estimado do nó $n$ a um nó objetivo,	p. 40
$C$	Circuito,	p. 64
$S$	Sub Circuito,	p. 64
$W$	Waypoint,	p. 64

# *1 Introdução*

## **1.1 Motivação**

É indiscutível a importância dos veículos na sociedade atual, seu uso passou em poucos anos de uma questão de conforto para uma questão de necessidade para grande parte da sociedade. A locomoção de pessoas e de produtos de um lugar para outro é uma tarefa necessária para toda a sociedade, função exercida por veículos que são quase sempre conduzidos por pessoas.

O grande aumento no número de veículos tem causado grandes problemas, e talvez os mais importantes sejam os relacionados à poluição e à segurança das pessoas que precisam manipular ou são expostas a esses veículos, esses problemas são agravados pela grande quantidade de veículos em circulação e pelo seu mal uso.

A primeira publicação a respeito desse grande aumento na quantidade e na importância dos veículos e de suas consequências foi feita por *Norman Bel Geddes* em seu livro *Magic Motorways* em 1939 [Geddes 1939]. Portanto não é um tema tão atual quanto possa parecer, no livro é tratado uma série de fatores consequentes do grande aumento no volume de veículos e feito um estudo sobre soluções para os problemas resultantes.

”O motorista pressiona um botão no painel de instrumentos que o levará para a rota desejada na primeira oportunidade. Todo o caminho será feito sem nenhum perigo de mudança de rotas ou interseções. Tudo é mantido por um controle automático do carro.”[Geddes 1939]

Esse trecho retirado do livro *Magic Motorways* mostra uma previsão muito otimista para um futuro vinte anos a frente da publicação do livro, portanto para o início da década de 1960. Essa visão otimista pode ter sido causada pela grande revolução tecnológica iniciada na época em que o livro foi escrito, que resultou em um otimismo exacerbado não só da sociedade em geral mas também de toda a sociedade científica.

Mas mesmo parecendo muito otimista hoje, a visão do autor foi extremamente visionária. Ele previu que com o ritmo dos avanços tecnológicos, seria possível através de suas implementações possibilitar entre grandes melhorias a existência de veículos autônomos programáveis.

Prevendo dessa forma que os avanços tecnológicos seriam a chave na solução dos problemas causados pela grande quantidade de veículos e por seu mal uso.

Atualmente o desenvolvimento de sistemas para navegação de veículos autônomos inteligentes tem sido de grande interesse tanto na área acadêmica quanto na indústria. Esses sistemas permitem que veículos tomem decisões sem a influência de operadores e determinem mudanças de comportamento para alcançarem algum objetivo estabelecido.

Este trabalho trata justamente do desenvolvimento de alguns subsistemas necessários para permitir que um veículo tenha a capacidade de tomar decisões de forma inteligente e autônoma para realizar funções como navegar de uma posição inicial para uma posição objetivo.

## 1.2 Estado da Arte

Como previsto por *Norman Bel Geddes* em 1939, hoje grande parte dos problemas causados pelos veículos tem sido amenizada com o avanço da tecnologia, que se deve em grande parte ao uso de computadores e sistemas de controle embarcados.

Esse avanço tem possibilitado o desenvolvimento de Sistemas de Transporte Inteligentes que são desenvolvidos utilizando não apenas sistemas de controle mas também sistemas de telecomunicações tanto no nível de veículos quanto no de infra-estrutura [Vlacic 2001].

Sistemas de auxílio a direção já têm sido largamente desenvolvidos pela indústria automotiva. Podendo destacar freios equipados com sistemas anti bloqueio de frenagem conhecidos como freios ABS (Anti-lock brake systems) que possibilitam um controle quando o veículo é freado pelo condutor, de forma que as rodas não são travadas resultando em uma maior eficiência para interromper o movimento de veículo. E também sistemas de controle de estabilidade ativa que previnem que o veículo rode ou derrape, conhecidos como ASCS (Active Stability Control Systems) [Rajamani 2006].

Um fator para um novo grande interesse no desenvolvimento de veículos autônomos inteligentes é a competição *DARPA Grand Challenge*, uma competição mantida pelo departamento de defesa do governo dos Estados Unidos destinada ao desenvolvimento de veículos autônomos que precisam completar uma série de objetivos preestabelecidos.

O *DARPA Grand Challenge* teve início em 2005 com foco no desenvolvimento de veículos autônomos que operam em um ambiente fora de estrada com a única interação com outros veículos. Em 2007 o ambiente foi alterado para um ambiente completamente urbano onde os veículos autônomos precisavam atingir objetivos de forma segura através de tráfego urbano.

Cada um dos veículos que participaram foi avaliado por sua forma de operar de maneira segura e efetiva com outros veículos imersos em um ambiente urbano [DARPA 2007].

Apesar de apenas recentemente a comunidade científica ter aceitado o desafio de desenvolver veículos autônomos inteligentes, os recentes avanços demonstrados nas competições do *DARPA Grand Challenge* provam a viabilidade do desenvolvimento desses veículos. Resultando em uma possibilidade realista de que num futuro próximo, veículos autônomos inteligentes sejam capazes de navegar e interagir de forma segura em ambientes urbanos complexos [Seetharaman, Lakhota e Blasch 2006].

Todo esse avanço tecnológico alcançado já não tem mais passado despercebido pela indústria automotiva, mas tem sido visto como um grande diferencial para a marca que conseguir o desenvolvimento de um veículo autônomo inteligente e essa busca tem gerado grandes investimentos pela indústria em pesquisas nessa área.

Existe hoje uma corrida no desenvolvimento de veículos autônomos inteligentes por praticamente todas as grandes marcas da indústria automotiva, mudando as pesquisas nessa área de uma visão puramente de pesquisa científica para uma visão de um grande negócio [Stoll 2007].

O mais recente esforço da indústria automotiva nesse sentido é o desenvolvimento de veículos autônomos inteligentes para a competição em provas de automobilismo de velocidade, disputando com pilotos reais em ambientes complexos. Visando principalmente o aprimoramento da tecnologia de veículos autônomos inteligentes em condições extremas de dirigibilidade, e se aproveitando da visibilidade conseguida para a marca com essa inusitada participação nesse tipo de evento [Williams 2009].

## 1.3 Objetivos

Este trabalho tem como objetivo o desenvolvimento de um sistema de navegação que fará parte de um veículo autônomo inteligente. O sistema de navegação tem que ser capaz de sob determinadas situações gerar sinais de controle capazes de fazer com que o veículo alcance algum objetivo pré estabelecido.

Para o sistema de navegação ser capaz de realizar as tarefas necessárias para a navegação de um veículo em um ambiente complexo a arquitetura do sistema de controle precisa ser desenvolvida de uma forma híbrida agregando tipos diferentes de camadas. As camadas precisam ser desenvolvidas para trabalharem de forma intercalada, permitindo que o resultado de uma camada seja complementado pelo resultado de outra gerando o acionamento necessário para o

veículo atingir o objetivo desejado.

Cada uma dessas camadas precisa de recursos específicos para seu funcionamento, como um modelo cinemático que descreve o comportamento do veículo e um algoritmo de busca capaz de encontrar uma sequência de ações que levam de um estado inicial para um estado objetivo, portanto alguns desses recursos também precisam ser desenvolvidos.

Devido a grande complexidade do sistema de navegação e a grande dependência de todos os outros sistemas que compõem o veículo autônomo inteligente, cada etapa de desenvolvimento precisa ser avaliada e validada. Para avaliar os resultados de cada alteração do sistema de navegação o desenvolvimento de um simulador se faz necessário, através do simulador todo o sistema de navegação pode ser avaliado e validado permitindo uma maior flexibilidade em todo o seu desenvolvimento.

## 2 *Veículo Autônomo Inteligente*

### 2.1 Introdução

Veículos inteligentes têm sido desenvolvidos sobre duas abordagens distintas, os sistemas de auxílio à direção que possibilitam ao motorista recursos de auxílio na condução do veículo e os veículos autônomos inteligentes que não necessitam de nenhuma interferência humana para realizarem determinadas funções [Vlacic 2001].

Este trabalho foi desenvolvido utilizando a abordagem de veículos autônomos inteligentes, que são veículos que operam de forma autônoma ou sem a intervenção de pessoas. Essa característica é obtida através da capacidade do veículo de perceber seu estado e o estado do ambiente em que ele está inserido e processar essa informação de forma a permiti-lo realizar alguma função [Vlacic 2001].

Um automóvel normal foi modificado de forma a possibilitar todo o desenvolvimento deste trabalho, esse veículo mais todo o sistema mecânico, eletrônico e de *software* embarcado desenvolvido para a navegação formam o que é chamado neste trabalho de *Veículo Autônomo Inteligente*.

### 2.2 Considerações iniciais

Em muitos aspectos se difere o modo como uma pessoa dirige um veículo para o modo como um sistema de navegação autônomo atua sobre o veículo. Essa diferença de comportamento se dá principalmente devido a diferença no modo de perceber o ambiente, que é causada pela necessidade de discretização, diferença no domínio de conhecimento e na tomada de decisões [Miller, Garcia e Campbell 2006].

A necessidade de discretização do mundo resulta numa diferença entre o mundo contínuo percebido por uma pessoa e no mundo discretizado processado pelo sistema de navegação autônomo. Uma pessoa atua sobre um veículo sem se preocupar ou até mesmo perceber certos limites, não importa se o veículo não está exatamente onde deveria, essa diferença se necessário

é simplesmente compensada por alguma outra atuação resultando em uma quantidade infinita de possibilidades na tomada de decisões.

A discretização é justamente a limitação na percepção e atuação sobre o mundo, essa limitação deve ser realizada de forma a permitir ao sistema de navegação autônomo um comportamento satisfatório na atuação sobre o veículo.

A diferença no domínio de conhecimento de uma pessoa para um sistema de navegação autônomo está na capacidade da uma pessoa interpretar mudanças no estado do ambiente com base em um conhecimento prévio de como esse ambiente se comporta.

Assim como os sensores do sistema de navegação tem erros de leitura os sensores humanos têm erros parecidos. Mas uma pessoa sabe que uma mudança incomum na sua capacidade de ver o ambiente a sua frente pode ser simplesmente alguma alteração momentânea de sua percepção que não acarretou em mudança nenhuma no ambiente de atuação do veículo.

Essa diferença no modo de interpretar mudanças de estado observável devido ao domínio de conhecimento resulta em grandes diferenças de comportamento entre uma pessoa e um sistema de navegação autônomo.

Diferenças na tomada de decisões são causas pela necessidade de modelar cada uma das decisões que podem ser tomadas pelo sistema de navegação, para um planejamento cada uma das decisões recebe um custo onde a melhor sequência de decisões é a que apresentar um menor custo total, mas não necessariamente será uma sequência que levará a solução mais rápida ou mais segura.

Todas essas considerações têm grande interferência no comportamento de um veículo autônomo inteligente e portanto, são analisadas e consideradas no desenvolvimento deste trabalho de forma a resultar em um comportamento o mais próximo possível do desejado para o *Veículo Autônomo Inteligente*.

## 2.3 Estrutura do trabalho

### 2.3.1 Introdução

O *Veículo Autônomo Inteligente* foi estruturado de forma a permitir que através de um sistema de navegação um automóvel comum automatizado possa realizar determinadas ações de forma autônoma e inteligente sobre o ambiente em que está inserido.

A Figura 2.1 mostra a estrutura básica de funcionamento do *Veículo Autônomo Inteligente*.

Onde o *Sistema de Navegação* é um conjunto de *softwares* desenvolvidos para interpretar sinais passados por sensores instalados no *Veículo Automatizado*, e através dessa informação enviar *Valores de Atuação* para o *Veículo Automatizado* de forma a realizar alguma função pré determinada.

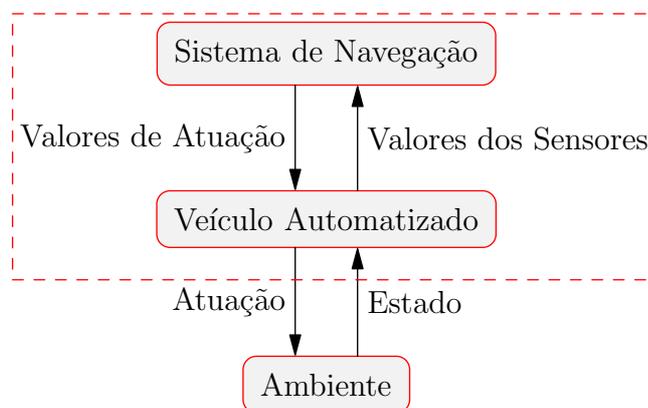


Figura 2.1: Estrutura do *Veículo Autônomo Inteligente*

*Valores de Atuação* são valores numéricos gerados pelo *Sistema de Navegação* que são interpretados pelo *Veículo Automatizado* e traduzidos em sinais de atuação para os atuadores que fazem parte do *Veículo Automatizado*. *Valores dos Sensores* são valores numéricos representando os sinais gerados pelos sensores que fazem parte do *Veículo Automatizado*.

Todas as ações realizadas pelo *Veículo Autônomo Inteligente* podem alterar o seu estado ou quando são resultado de sua *Atuação* sobre o *Ambiente* podem alterar o estado do *Ambiente* em que ele está inserido.

### 2.3.2 Sistema de Navegação

De uma forma geral, navegação é o desenvolvimento de um conjunto de ações que são capazes de fazer com que o veículo se movimente de um estado inicial para um estado objetivo, portanto é uma generalização de um problema de descoberta de rotas [Russell e Norvig 2003].

Dessa forma, o *Sistema de Navegação* é composto por todos os componentes de *software* de alto nível responsáveis pelo controle do *Veículo Automatizado*. É responsável por interpretar todos os *Valores dos Sensores* passados pelo *Veículo Automatizado* para determinar o estado atual e gerar *Valores de Atuação* que são enviados para o *Veículo Automatizado* de forma a alcançar o estado objetivo desejado.

O *Sistema de Navegação* foi desenvolvido de forma modular para permitir uma constante evolução de seus componentes de forma independente e a inclusão de novos componentes, ga-

rantindo assim uma flexibilidade em todo seu desenvolvimento. A Figura 2.2 mostra a estrutura desenvolvida para o *Sistema de Navegação*.

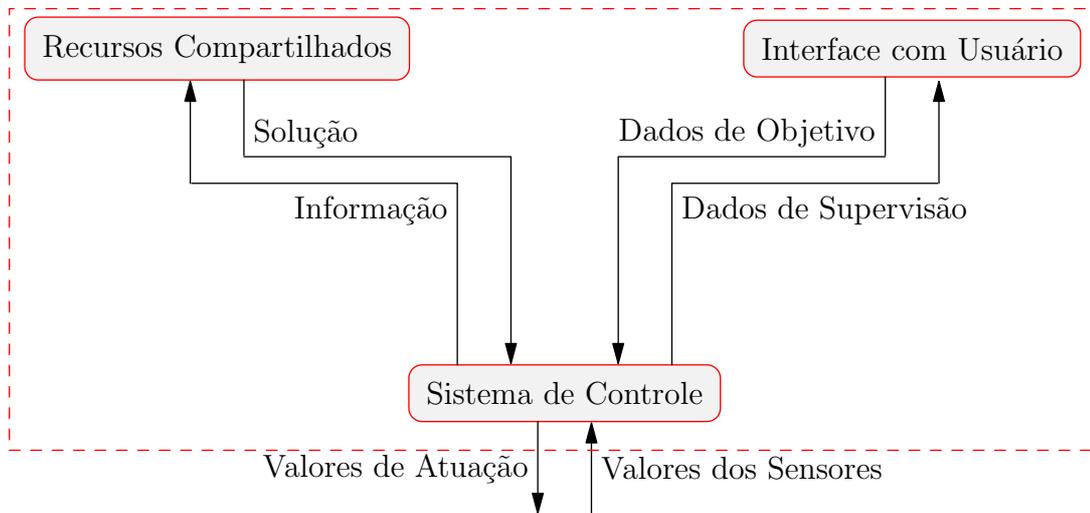


Figura 2.2: Estrutura do *Sistema de Navegação*

Os principais componentes que formam o *Sistema de Navegação* serão melhor discutidos nos próximos capítulos, mas inicialmente se faz necessária uma discussão mais geral de cada um desses componentes para uma melhor compreensão do que foi desenvolvido neste trabalho.

### Interface com Usuário

O primeiro dos componentes do *Sistema de Navegação* é a *Interface com o Usuário*. O *Sistema de Navegação* precisa receber objetivos para gerar *Valores de Atuação* para que o *Veículo Automatizado* alcance esses objetivos, a *Interface com o Usuário* é responsável por permitir que o usuário passe esses objetivos e traduzi-los na forma de *Dados de Objetivo* para o *Sistema de Navegação*.

A *Interface com o Usuário* foi desenvolvida de forma a possibilitar uma maneira intuitiva para o usuário do *Veículo Autônomo Inteligente* ter acesso a todos os recursos disponíveis pelo sistema. Recebe do *Sistema de Controle* os *Dados de Supervisão* que são informações sobre o processo do *Sistema de Controle* que são mostradas ao usuário permitindo uma supervisão em tempo real.

Não faz parte dos objetivos deste trabalho o desenvolvimento de uma discussão sobre interfaces com usuário, mas basta saber que a *Interface com Usuário* foi desenvolvida tendo em vista sua utilização no desenvolvimento do *Sistema de Navegação* e não como uma interface destinada a utilização de um usuário final.

## Recursos Compartilhados

Para a solução de grande parte dos problemas de navegação, o *Sistema de Navegação* precisa ter a capacidade de tratar determinadas informações inerentes a cada um dos problemas, esse tratamento de informações pode ser desde uma simples mudança de representação da informação como uma solução complexa para parte do problema.

Os componentes do *Sistema de Navegação* responsáveis por tratar essas informações estão agrupados na forma de *Recursos Compartilhados*. Cada um dos *Recursos Compartilhados* é formado por classes estáticas e recebe a *Informação* que é tratada resultando em uma *Solução*.

## Sistema de Controle

O *Sistema de Controle* é responsável por determinar os *Valores de Atuação* que são enviados ao *Veículo Automatizado* para obter um comportamento desejado do *Veículo Inteligente*, para levá-lo de um estado inicial a um estado objetivo.

Os *Valores dos Sensores* passados pelo *Veículo Automatizado* são interpretados para determinar o estado atual do *Veículo Autônomo Inteligente* e do *Ambiente*, com base nesses estados são gerados *Valores de Atuação* para alcançar um estado objetivo desejado.

O *Sistema de Controle* tem acesso aos *Recursos Compartilhados* que são componentes essenciais para a solução dos problemas de navegação tratados neste trabalho.

### 2.3.3 Veículo Automatizado

O *Veículo Automatizado* é formado pelo *Veículo* um grupo de *Atuadores* e um de *Sensores*, como pode ser visto em detalhes da Figura 2.3.

O *Veículo* é um automóvel Chevrolet Zafira modelo Elite ano 2008 que recebeu uma série de adaptações mecânicas para receber todos os equipamentos do sistema de eletrônica, *software* embarcado, atuadores e sensores necessários para sua automatização.

Cada um dos *Atuadores* possui um *Controle de Atuação* que é um sistema de controle dedicado responsável por traduzir os *Valores de Atuação* passados pelo *Sistema de Navegação* em *Sinais de Acionamento* necessários para que os *Atuadores* assumam cada estado passado pelo *Sistema de Navegação*.

Sensor é qualquer componente que pode ler alguma informação do ambiente e passar essa informação para o sistema, podendo ser desde sensores simples como chaves digitais a sensores

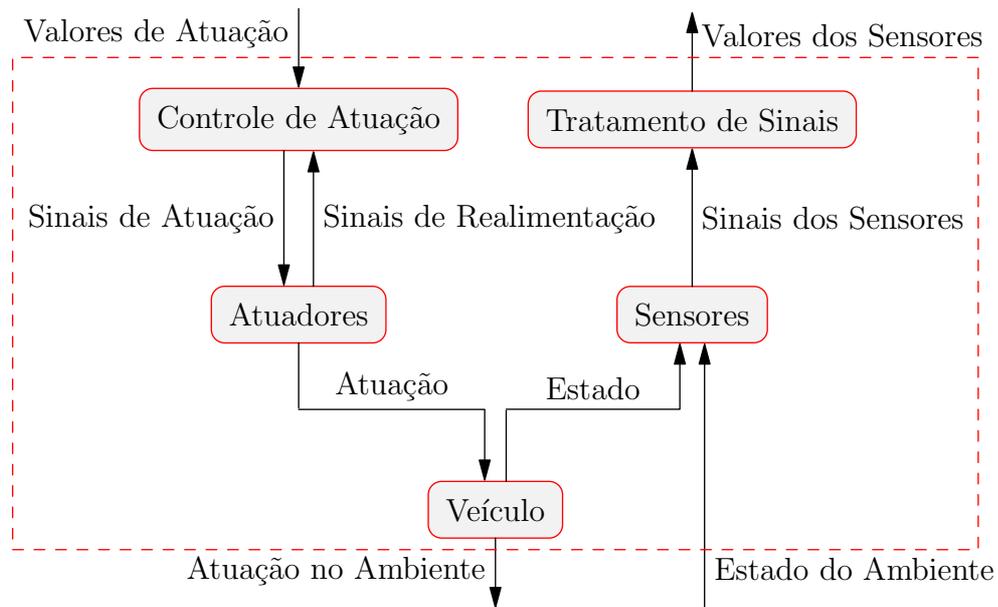


Figura 2.3: Estrutura do *Veículo Automatizado*

capazes de medir muitas informações simultâneas [Russell e Norvig 2003].

Cada um dos *Sensores* percebe alguns dos estados do *Veículo Automatizado* ou do *Ambiente*, a partir desses estados são gerados sinais que através do *Tratamento de Siniais* são traduzidos para *Valores dos Sensores* e passados para o *Sistema de Navegação*.

### 2.3.4 Ambiente

O *Ambiente* em que o *Veículo Autônomo Inteligente* está inserido apresenta as seguintes características:

- Apenas variações no ambiente próximas o suficiente para serem captadas pelos sensores instalados no *Veículo* são percebidas;
- O comportamento do ambiente não pode ser previsto já que é um ambiente constantemente alterado por outros agentes além do veículo;
- Cada ação a curto prazo executada pelo *Veículo* tem consequências no estado do ambiente a longo prazo;
- O ambiente está sempre alterando seu estado mesmo que o *Veículo* não esteja executando nenhuma ação;
- É um ambiente físico real;

---

Essas características determinam o ambiente como sendo parcialmente observável, estocástico, sequencial, dinâmico e contínuo [Russell e Norvig 2003].

Esses fatores resultam em um *Ambiente* complexo que necessita de algumas simplificações para ser interpretado de forma conveniente. As simplificações são impostas pelo *Sistema de Controle* permitindo que para cada problema de navegação simplificações diferentes sejam realizadas, portanto o *Ambiente* pode ter o nível de complexidade necessário para cada tipo de problema que vai ser solucionado.

## 3 *Recursos Compartilhados*

### 3.1 Introdução

*Recursos Compartilhados* são basicamente classes estáticas que realizam alguma operação que pode ser necessária para mais de um *Controlador* implementado no *Sistema de Controle*.

Para os *Controladores* desenvolvidos neste trabalho são necessários três *Recursos Compartilhados*, um capaz de estimar o *Comportamento* do *Veículo Automatizado*, um responsável por planificar as coordenadas globais determinando todas as posições em um plano bi dimensional e um capaz de solucionar problemas modelados na forma de árvores de busca.

Devido a grande diferença entre o tipo de informação tratada por cada um dos *Recursos Compartilhados* implementados, os valores de entrada recebidos pelos *Recursos Compartilhados* foram chamados apenas de *Informação* e a resposta encontrada chamada de *Solução*.

### 3.2 Modelo Cinemático

#### 3.2.1 Introdução

No desenvolvimento de *softwares* de navegação, a tarefa inicial é compreender o comportamento do sistema a ser controlado, para dessa forma construir sistemas de navegação que vão alcançar os resultados desejados. A forma de representar a variação de qualquer sistema no tempo é chamada de modelo cinemático.

Um modelo cinemático pode ser basicamente descrito utilizando álgebra ou equações diferenciais que descrevem as forças ou movimentos de interesse para serem controlados. Através desse modelo matemático é possível a predição do comportamento do sistema modelado causado pela variação no tempo das suas variáveis de entrada [Thomas D 1992].

Em sistemas complexos como veículos existe um número muito grande de variáveis, componentes e subsistemas não lineares que influenciam no comportamento do sistema, o que faz com que a modelagem de todos esses detalhes do sistema seja virtualmente impossível

[Thomas D 1992]. Portanto, por mais abrangente que seja o modelo cinemático ele será uma aproximação do modelo real.

Devido a essa limitação e ao objetivo deste trabalho, algumas simplificações são feitas no *Modelo Cinemático*, sem causarem prejuízos ao resultado final. As considerações feitas simplificam o modelo desprezando muitos dos fatores de baixa influência no comportamento resultante do *Veículo* para as condições impostas neste trabalho.

Neste trabalho o *Modelo Cinemático* foi projetado de forma a descrever o comportamento do *Veículo* nas condições previstas para este trabalho, com algumas considerações para simplificação do modelo e uma melhor representação. O *Modelo Cinemático* é um recurso utilizado pelo *Sistema de Controle* para a solução de problemas em que o comportamento do *Veículo* precisa ser estimado.

### 3.2.2 Posicionamento e orientação

Neste trabalho o ambiente em que o veículo está inserido é considerado como bidimensional, onde existe apenas movimentação no plano horizontal representado pelos eixos  $X$  e  $Y$  considerando o ambiente completamente plano. O eixo  $Y$  está orientado ao eixo de coordenadas geográficas Norte e o eixo  $X$  ao eixo de coordenadas Leste. Esse é o sistema de coordenadas globais em que o veículo está inserido.

Para a orientação do veículo em relação a esse plano é necessário um sistema de coordenadas locais que é fixado em relação à orientação do veículo, onde o eixo que representa a movimentação longitudinal do veículo é chamado de  $x_i$  e o eixo que representa a movimentação lateral é chamado de  $y_j$ .

Para determinar o posicionamento e orientação do veículo em relação ao ambiente é necessário um vetor  $\vec{P}$  contendo os valores de sua posição global  $P_x$  e  $P_y$  e um valor representando o ângulo  $\psi$  formado entre o eixo de coordenadas locais do veículo  $x_i$  e o eixo de coordenadas globais  $X$ , dessa forma:

$$\vec{P} = [P_x, P_y, \psi] \quad (3.1)$$

Resultando em uma movimentação com três graus de liberdade, dois para a posição e um para o ângulo do veículo em relação ao ambiente [Siegwart e Nourbakhsh 2004]. Esse modelo de representação é mostrado na Figura 3.1 e é utilizado em todo este trabalho para a representação da posição do veículo em relação ao ambiente.

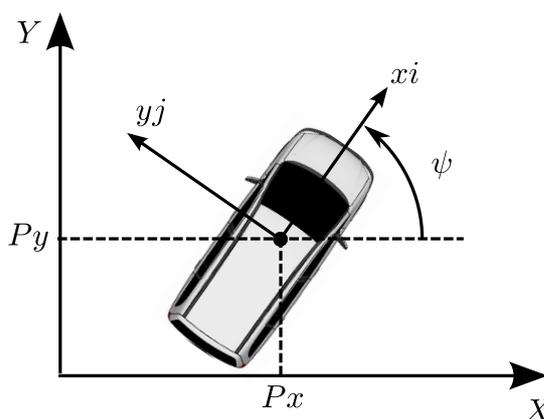


Figura 3.1: Posicionamento do veículo

O ponto de origem dos vetores  $x_i$  e  $y_j$  é o centro de massa do veículo e o ponto de origem dos eixos  $X$  e  $Y$  é foi definido como a localização do grupo de pesquisa CRTi (Centro de Referência em Tecnologia da Informação) localizado dentro do campus da Universidade Federal de Itajubá UNIFEI.

### 3.2.3 Modelo Desenvolvido

O *Modelo Cinemático* foi desenvolvido para determinar a movimentação do veículo em um ambiente bidimensional de coordenadas fixas, causada pelas variações da velocidade e do ângulo das rodas dianteiras do veículo em relação ao tempo.

Para um pequeno período de tempo a velocidade pode ser considerada constante e o ângulo das rodas dianteiras do veículo se mantêm fixos. Nessas condições o veículo realiza um movimento circular com um raio de curvatura constante, chamado de curva de estado estacionário [Abe 2009].

Partindo dessas considerações o modelo cinemático pode ser deduzindo através de relações trigonométricas, para isso é utilizado o modelo de veículo bicíclico mostrado abaixo na Figura 3.2.

Neste modelo as rodas traseiras e dianteiras do veículo são substituídas por apenas uma roda dianteira e uma traseira, posicionadas respectivamente no ponto  $A$  e ponto  $B$ , o ângulo da roda dianteira é representado por  $\delta$ . O ponto  $C$  representa o centro de massa do veículo e  $l_f$  e  $l_r$  representam respectivamente a distância do ponto  $C$  para o ponto  $A$  e o ponto  $B$ .

O veículo está inserido em um ambiente com representação bidimensional de coordenadas  $X$  e  $Y$ , e sua posição em relação a esse ambiente é representada através da posição  $P_x$  e  $P_y$  do centro de gravidade e um ângulo  $\psi$  que descreve a orientação do veículo com relação ao

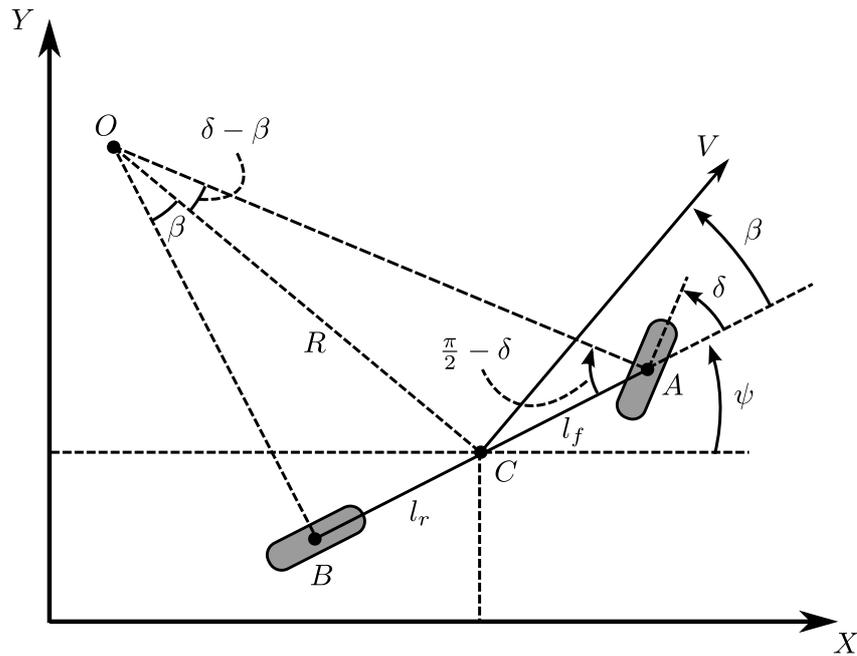


Figura 3.2: Modelo de Veículo Bicíclico

ambiente. A velocidade é representada pelo vetor  $\vec{V}$  medida no centro de gravidade com um ângulo  $\beta$  em relação ao eixo longitudinal do veículo.

Para esse sistema desenvolvido o veículo se locomove em velocidades consideradas baixas resultando em uma foça lateral gerada pelos pneus pequena, e nesse caso é possível considerar que os vetores de velocidade das rodas são na direção de cada uma das rodas [Rajamani 2006].

Nessas condições o ponto  $O$  é o centro instantâneo de curvatura do veículo definido pela intercessão de duas linhas perpendiculares a cada uma das rodas, respectivamente  $AO$  e  $BO$ . O raio de curvatura é definido pela distância do ponto  $C$  ao ponto  $O$ , representado pela linha  $OC$ .

Pelo modelo descrito, aplicando a lei dos senos no triângulo  $OCA$ .

$$\frac{\text{sen}(\delta - \beta)}{l_f} = \frac{\text{sen}(\frac{\pi}{2} - \delta)}{R} \quad (3.2)$$

Aplicando a lei dos senos no triângulo  $OCB$ .

$$\frac{\text{sen}(\beta)}{l_r} = \frac{\text{sen}(\frac{\pi}{2})}{R} \quad (3.3)$$

Aplicando transformações trigonométricas em (3.2).

$$\frac{\text{sen}(\delta)\cos(\beta) - \text{sen}(\beta)\cos(\delta)}{l_f} = \frac{\cos(\delta)}{R} \quad (3.4)$$

Aplicando transformações trigonométricas em (3.3).

$$\frac{\text{sen}(\beta)}{l_r} = \frac{1}{R} \quad (3.5)$$

Multiplicando os dois lados da equação (3.4) por  $\frac{l_f}{\cos(\delta)}$ .

$$\tan(\delta)\cos(\beta) - \text{sen}(\beta) = \frac{l_f}{R} \quad (3.6)$$

Multiplicando os dois lados da equação (3.5) por  $l_r$ .

$$\text{sen}(\beta) = \frac{l_r}{R} \quad (3.7)$$

Somando as equações (3.6) e (3.7).

$$\tan(\delta)\cos(\beta) = \frac{l_f + l_r}{R} \quad (3.8)$$

Assumindo que devido a baixa velocidade de movimentação do veículo seu raio de curvatura muda lentamente, pode ser considerado que a variação na orientação do veículo é igual a sua velocidade angular [Rajamani 2006].

Portanto:

$$\psi = \frac{V}{R} \quad (3.9)$$

Reescrevendo (3.8) em função de (3.9) e obtendo as equações de movimento para o modelo.

$$\dot{\psi} = V \frac{\tan(\delta)\cos(\beta)}{l_f + l_r} \quad (3.10)$$

Dessa forma:

$$\dot{X} = V \cos(\psi + \beta) \quad (3.11)$$

$$\dot{Y} = V \text{sen}(\psi + \beta) \quad (3.12)$$

Reescrevendo (3.6) em função de (3.7) e obtendo o valor de  $\beta$ .

$$\beta = \tan^{-1}\left(\frac{l_r}{l_f + l_r} \tan(\delta)\right) \quad (3.13)$$

Cada roda dianteira precisa ter um ângulo ligeiramente diferente para que o centro da curva realizada pelo veículo seja a intercessão das retas perpendiculares a cada uma das rodas do veículo. A relação entre cada um dos ângulos das rodas dianteiras é chamado de geometria de Ackerman [Abe 2009].

No modelo de veículo bicíclico desenvolvido, as duas rodas dianteiras são substituídas por apenas uma, essa aproximação é válida mas no caso do veículo real temos duas rodas dianteiras que têm ângulos que apesar de parecidos, são ligeiramente diferentes. Essa diferença existente entre as rodas dianteiras interna e externa à curva, respectivamente  $\delta_i$  e  $\delta_o$ , é devido a distância  $l_w$  entre as rodas dianteiras como mostrado abaixo na Figura 3.3.

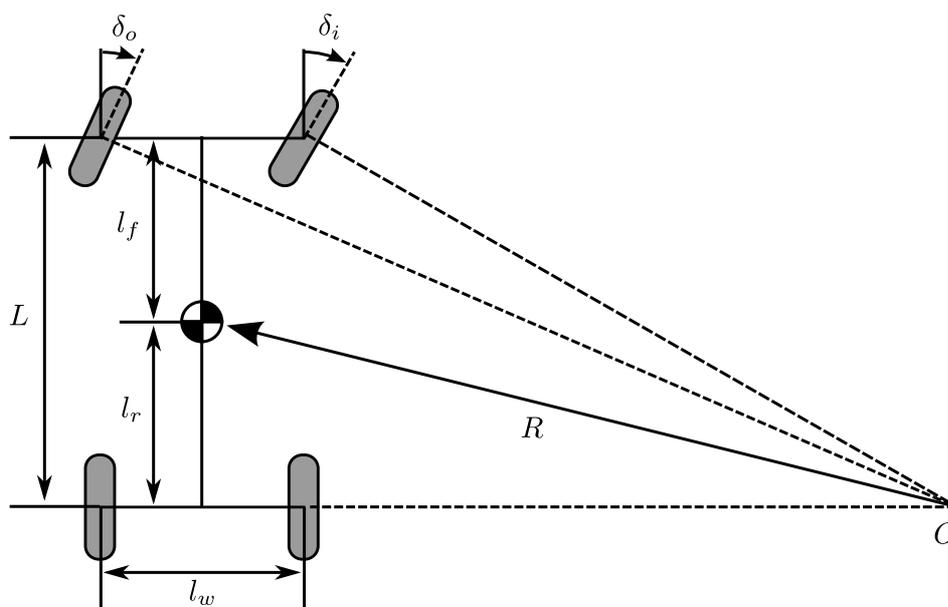


Figura 3.3: Modelo do Veículo

Para uma melhor aproximação ao modelo bicíclico desenvolvido, a roda dianteira é considerada como tendo um ângulo igual a  $\delta = \frac{\delta_o + \delta_i}{2}$ .

Considerando  $R = \frac{L}{\tan(\delta)\cos(\beta)}$ , através das equações (3.9) e (3.10) se obtém que:

$$R = \frac{L}{\tan(\delta)\cos(\beta)} \quad (3.14)$$

Considerando os ângulos  $\delta$  e  $\beta$  pequenos a equação (3.14) pode ainda ser aproximada para:

$$R = \frac{L}{\delta} \quad (3.15)$$

Discretizando as equações de movimento (3.10), (3.11), (3.12) e (3.13) obtemos:

$$\beta_{(k)} = \tan^{-1} \left( \frac{l_r}{l_f + l_r} \tan(\delta_{(k)}) \right) \quad (3.16)$$

$$\psi_{(k)} = \psi_{(k-1)} + V_{(k)} \frac{\tan(\delta_{(k)}) \cos(\beta_{(k)})}{l_f + l_r} (T_{(k)} - T_{(k-1)}) \quad (3.17)$$

$$P_{x(k)} = P_{x(k-1)} + V_{(k)} \cos(\psi_{(k)} + \beta_{(k)}) (T_{(k)} - T_{(k-1)}) \quad (3.18)$$

$$P_{y(k)} = P_{y(k-1)} + V_{(k)} \sin(\psi_{(k)} + \beta_{(k)}) (T_{(k)} - T_{(k-1)}) \quad (3.19)$$

Onde  $T_{(k-1)}$  é o tempo necessário para ir do estado inicial ao estado  $k-1$  e  $T_{(k)}$  é o tempo necessário para ir do estado inicial ao estado  $k$ .

Através das equações discretizadas de movimento (3.16), (3.17), (3.18) e (3.19) é realizada toda a estimação de movimentação do veículo, bastando conhecer os valores  $P_x$ ,  $P_y$  e  $\psi$  da posição inicial e os parâmetros  $l_f$  e  $l_r$  tendo como sinais de acionamento a velocidade  $V$  e o ângulo médio das rodas dianteiras  $\delta$ .

## 3.3 Planificação de Coordenadas

### 3.3.1 Introdução

O *Recurso Compartilhado de Planificação de Coordenadas* é responsável por determinar o posicionamento global através de uma planificação dos valores das coordenadas globais com base em um ponto de referência.

Localização é a determinação de onde os objetos importantes para o sistema de navegação estão, é um dos problemas mais importantes porque a localização dos objetos é o principal fator que vai influenciar as interações físicas de todo o sistema [Russell e Norvig 2003].

Para uma melhor representação da localização é utilizada a planificação das coordenadas globais que é uma representação das posições globais mais simples de ser tratada e analisada,

representando todas as posições como pontos em um plano bi dimensional com um centro conhecido.

### 3.3.2 Posicionamento Global

Posicionamento global é a posição do *Veículo Autônomo Inteligente* em relação à superfície da Terra, utilizando coordenadas geográficas de latitude e longitude determinadas pelo Sistema de Navegação Global.

O sistema de navegação global GPS foi criado pelo Departamento de Defesa dos Estados Unidos no programa de satélites NAVSTAR. Para que o sistema opere em toda a superfície da terra são necessários 24 ou mais satélites divididos em seis órbitas com quatro ou mais satélites em cada, bastando quatro ou mais satélites para a determinação da posição do observador [Grewal, Weill e Andrews 2007].

Neste trabalho o sistema de coordenadas globais é determinado pelos valores de leitura do DGPS (Differential GPS), uma variação do sistema de localização global GPS, que através de sinais de satélites é capaz de determinar a localização precisa de seu receptor.

O sistema de coordenadas adotado para o posicionamento global é o ECEF sigla para o termo em inglês *Earth-centered, Earth-fixed* [Kaplan e Hegarty 2006]. O sistema de coordenadas ECEF é mostrado na Figura 3.4.

O posicionamento é determinado através dos ângulos de latitude e longitude, onde o ponto de  $0^\circ$  de latitude é definido como sendo qualquer ponto na linha do Equador e o ponto de  $0^\circ$  de longitude é definido como qualquer ponto na linha do meridiano de Greenwich. Essa convenção foi definida no *International Meridian Conference* ocorrido na cidade de Washington nos Estados Unidos em Outubro de 1884 [Grewal, Weill e Andrews 2007].

O ângulo  $\phi$  de latitude é definido como o ângulo entre o plano formado pela linha do Equador e a linha formada entre o ponto  $C$  no centro da Terra e o ponto  $P$  de referência, os ângulos formados ao Norte da linha do Equador são positivos e os formado ao Sul são negativos.

O ângulo  $L$  de longitude é definido como o ângulo entre o Meridiano de Greenwich e a linha formada entre o ponto  $C$  no centro da Terra e o ponto  $P$  de referência, os ângulos formados ao Leste do Meridiano de Greenwich são positivos e os formado ao Oeste são negativos.

A melhor representação da superfície da Terra é através de uma elipsóide, e seus parâmetros são o semi-eixo maior  $a$  e o semi-eixo menor  $b$ . Os parâmetros utilizados neste trabalho para a representação são:

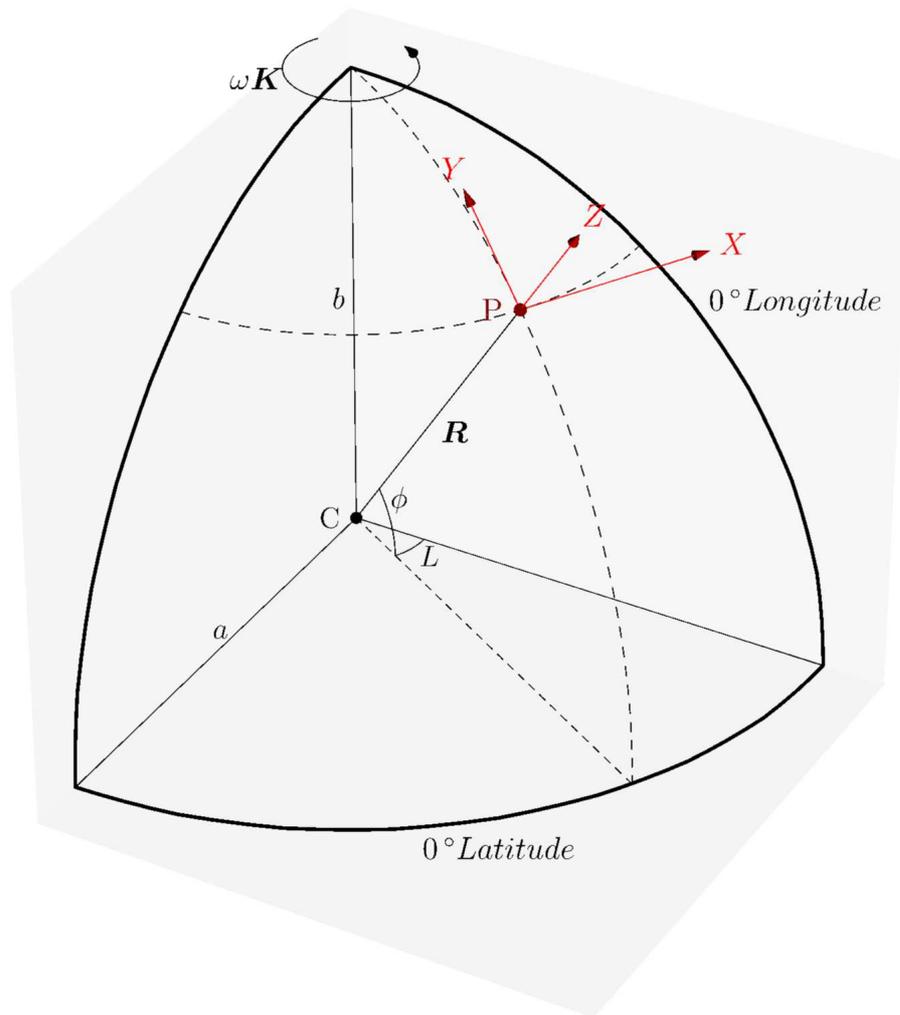


Figura 3.4: Sistema de Coordenadas ECEF

$$a = 6378137m \quad (3.20)$$

$$\frac{1}{f} = 298,257223563 \quad (3.21)$$

Onde o fator de achatamento da elipsóide é definido como:

$$f = \frac{a-b}{a} \quad (3.22)$$

Esses parametros são definidos em *MGS84* sigla para World Geodetic System realizado em 1984 e são os mais utilizados nos sistemas de posicionamento global [NIMA 2000].

### 3.3.3 Metodologia

Para uma melhor interpretação e processamento da posição de qualquer objeto é necessário uma planificação das coordenadas de latitude e longitude para posições em um plano formado pelos eixos  $X$  e  $Y$ . Essa planificação é realizada através do cálculo de distâncias entre pontos, foi definido um ponto central localizado dentro do *Campus* da universidade onde toda posição é determinada como sendo sua distância nos eixos  $X$  e  $Y$  para esse ponto central.

Um dos métodos de calcular as distâncias dos arcos do elipsóide formado entre dois pontos qualquer na superfície da Terra foi desenvolvido por *T. Vincenty* no seu trabalho "*Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations*", e esse método é utilizado no processo de planificação das coordenadas [Vincenty 1975].

Utilizando esse método foi desenvolvido o operador  $D_{xy}(\phi, L)$  onde  $\phi$  e  $L$  são respectivamente as coordenadas de latitude e longitude do ponto  $P$ , retornando  $P_x$  e  $P_y$  que são respectivamente as coordenadas planificadas do ponto  $P$  nos eixos  $X$  e  $Y$ .

## 3.4 Algoritmo de Busca

### 3.4.1 Introdução

Neste trabalho o *Algoritmo de Busca* é um dos *Recursos Compartilhados* disponibilizados ao *Sistema de Controle* para a solução de alguns dos problemas de navegação em que a solução depende de busca.

Um dos maiores desafios encontrados no desenvolvimento de robôs autônomos é o planejamento automático de rotas [Choset et al. 2005]. E também é base de qualquer sistema de navegação para veículos autônomos inteligentes.

A busca de rotas é um dos principais objetivos do *Sistema de Navegação* e consiste em encontrar uma rota que leve o *Veículo Autônomo Inteligente* do seu estado inicial para um estado objetivo. Portanto para a solução de grande parte dos problemas de busca de rotas é necessário o desenvolvimento de um algoritmo de busca.

Busca de rotas é um problema de busca, portanto pode ser definido como tendo um estado inicial indicando a posição inicial, uma função de sucessão que determina cada estado que pode ser alcançado, um teste de objetivo que verifica se o estado é o estado objetivo e uma função de custo que determina o custo para alcançar um determinado estado, onde a solução ótima é a que apresentar o menor custo [Russell e Norvig 2003].

O algoritmo de busca precisa receber um estado inicial e através de um operador de sucessão ir buscando novos estados para alcançar algum dos estados objetivo, e como resultado da busca devolver todos os estados necessários para a partir do estado inicial alcançar o estado objetivo.

### 3.4.2 Busca em Árvore

A solução de alguns problemas é alcançada realizando uma busca através do espaço de estados, quando esse espaço de estados é gerado a partir de um estado inicial que pode ser expandido através de uma função de sucessão, essa busca é chamada de busca em árvore [Russell e Norvig 2003].

Uma árvore de busca  $A$  é definida como um conjunto de elementos  $\{n\}$  chamados de nós e um conjunto de elementos  $\{a\}$  que são ligações entre nós chamados de arcos. Se existir um arco partindo do nó  $n_a$  ao nó  $n_b$ , o nó  $n_b$  é um sucessor de  $n_a$  [Hart, Nilsson e Raphael 1968].

A busca inicia com um estado inicial que é determinado pelo problema que vai ser resolvido e é representado por um nó. Para alcançar novos nós, é necessário a expansão do nó atual, isso é realizado executando a função de sucessão no nó atual que gera um conjunto de nós sucessores do nó atual.

O operador de sucessão  $\Gamma$  quando aplicado sobre o nó  $n_i$  resulta no conjunto  $\{n_j, C_{i,j}\}$ , onde  $n_j$  são os nós sucessores de  $n_i$ , e  $C_{i,j}$  são os custos para ir do nó  $n_i$  para cada nó sucessor  $n_j$ . O custo de cada nó determinado quando operador  $\Gamma$  é executado e é a base para a seleção de em qual dos estados gerados será executado o operador  $\Gamma$ . A Figura 3.5 mostra essa estrutura de árvore de busca.

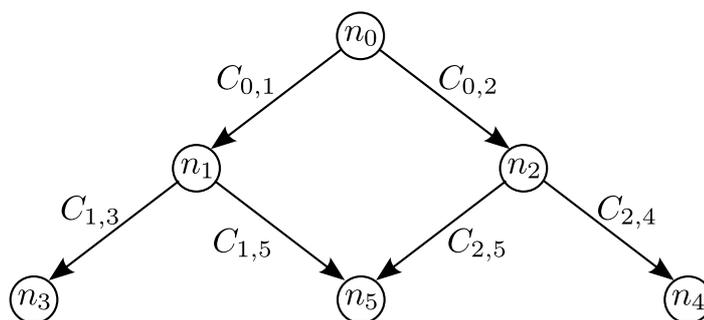


Figura 3.5: Árvore de busca

A cada novo conjunto de nós gerado é feita a verificação se algum deles é um nó objetivo. Portanto seja o conjunto  $O_n$  o conjunto de nós objetivo, um nó  $n$  será um nó objetivo quando  $n \in O_n$ .

O caminho ótimo de  $n_i$  para  $n_j$  é definido como o caminho que tem o menor custo de todos

os caminhos de  $n_i$  para  $n_j$ , esse custo é representado por  $h(n_i, n_j)$ . O conjunto de nós objetivo é chamado de  $T$  e o caminho ótimo de um nó  $n$  para um nó objetivo  $t$  onde  $t \in T$ , é considerado ótimo se:

$$h(n) = \min_{t \in T} h(n, t) \quad (3.23)$$

Essa é a base da busca em árvore e esses passos são repetidos até que um caminho ótimo ao estado objetivo é alcançado ou todos os estados possíveis são gerados. Muitas estratégias de busca são utilizadas para uma melhor eficiência para se alcançar um estado objetivo.

### 3.4.3 Metodologia

O algoritmo de busca é implementado com uma metodologia onde o algoritmo de busca é independente do domínio do problema, fazendo com que o mesmo algoritmo seja capaz de solucionar qualquer problema de busca em árvore contanto que o problema seja implementado seguindo uma série de regras.

A independência de domínio possibilita que no caso de uma substituição do algoritmo de busca por outro não seja necessário o desenvolvimento de uma nova modelagem para os problemas já modelados, bastando que o novo algoritmo de busca possua as mesmas características do algoritmo desenvolvido neste trabalho.

Pela metodologia empregada no algoritmo de busca, todos os nós são instâncias de classes filhas da classe *Nó* e possuem métodos responsáveis por determinar características de cada nó que são inerentes ao domínio do problema de interesse em ser solucionado. Os métodos da classe *Nó* são:

- Método  $\Lambda$  responsável por determinar quando um nó será considerado igual ao nó atual;
- Método  $\Omega$  que indica se o nó atual é um dos nós de destino;
- Método  $\Gamma$  que determina os nós sucessores ao nó atual;
- Métodos  $\hat{g}$  e  $\hat{h}$  para cálculo dos custos;

A Figura 3.6 mostra o diagrama de classes simplificado para a classe *Nó*.

O método  $\Gamma$  também adiciona a cada nó sucessor  $n$  o seu nó antecedente, dessa forma é possível conhecendo apenas  $n$  determinar todos os nós que formam o caminho do nó inicial até

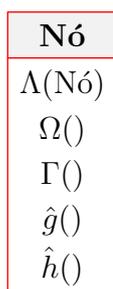


Figura 3.6: Diagrama Simplificado da Classe *Nó*

o nó  $n$ . Portanto se o nó  $n$  for aberto quando já estiver na lista de nós fechados  $F$ , basta ter seu valor  $\hat{f}(n)$  atualizado e seu nó antecessor substituído que todo o caminho do nó inicial ao nó  $n$  será automaticamente atualizado.

Da forma como foi desenvolvido, o algoritmo de busca fica independente do domínio e pode ser utilizado para a solução de vários problemas de busca, bastando que as classes filhas da classe *Nó* sejam implementadas seguindo as regras descritas, onde cada instância representa um estado do problema que vai ser solucionado.

### 3.4.4 A\* (A Estrela)

#### Introdução

A\* (A estrela) é o algoritmo de busca mais bem conhecido, sua heurística se baseia no conhecimento prévio do domínio do problema para determinar custos que são utilizados na busca. Pelas características de algoritmos de busca, nenhum outro algoritmo de busca ótimo garante abrir menos nós que o A\* [Russell e Norvig 2003].

O algoritmo A\* foi escolhido devido as suas características de, quando modelado corretamente, garantir a solução ótima e de sua heurística funcionar muito bem em problemas de rotas, já que o conhecimento prévio que ele necessita nesse caso é basicamente a distancia entre a posição inicial e a posição objetivo.

Para expandir um número mínimo de nós o algoritmo necessita de alguma informação indicando qual dos nós será expandido, se um nó que não leva ao caminho ótimo for expandido será um esforço inútil para a solução final e se nós que podem levar a um caminho ótimo forem ignorados o resultado final poderá não ser o resultado ótimo [Hart, Nilsson e Raphael 1968].

A informação que o algoritmo utiliza para a seleção de qual dos nós será aberto é passada pelos valores dos custos  $\hat{g}$  e  $\hat{h}$ , condicionando o desempenho do algoritmo ao quão bem esses valores dos custos representam os custos reais do problema que será solucionado.

### Função de Avaliação

Para qualquer árvore de busca  $A$  e qualquer conjunto de nós objetivos  $T$ , seja  $f(n)$  o custo de um caminho ótimo indo do nó inicial a um nó objetivo passando pelo nó  $n$ . Normalmente o valor do custo  $f(n)$  não é conhecido até que se obtenha a solução ótima, portanto o valor de  $f(n)$  precisa ser estimado pela função de avaliação  $\hat{f}(n)$  [Hart, Nilsson e Raphael 1968].

A função de avaliação pode ser escrita como sendo:

$$f(n) = g(n) + h(n) \quad (3.24)$$

Onde  $g(n)$  é o custo atual do caminho ótimo do nó inicial ao nó  $n$  e  $h(n)$  é o custo atual do caminho ótimo do nó  $n$  a um nó objetivo, portanto para estimar o valor de  $f$  é necessário estimar os valores de  $g$  e  $h$ , resultando em:

$$\hat{f}(n) = \hat{g}(n) + \hat{h}(n) \quad (3.25)$$

O valor de  $\hat{g}(n)$  é uma estimativa de  $g(n)$  resultando custo do caminho ótimo encontrado pelo algoritmo do nó inicial ao nó  $n$ , o que resulta em  $\hat{g}(n) \geq g(n)$ , portanto a estimativa do caminho de menor custo encontrado nunca terá um custo menor que o custo ótimo [Hart, Nilsson e Raphael 1968].

O valor de  $\hat{h}(n)$  é uma estimativa de  $h(n)$  resultando em um custo estimado do nó  $n$  a um nó objetivo, esse custo depende de um conhecimento do domínio do problema.

Para que o algoritmo seja ótimo a condição  $\hat{h}(n) \leq h(n)$  precisa ser respeitada, se essa condição não for respeitada o caminho encontrado pelo algoritmo não será garantidamente ótimo, mas se o valor de  $\hat{h}(n)$  for muito menor que o valor de  $h(n)$  o algoritmo vai abrir nós que não levam a um caminho ótimo resultando em um esforço inútil para o algoritmo.

### Algoritmo

A base da heurística do algoritmo de busca  $A^*$  é o operador de avaliação  $\hat{f}(n)$ , onde o nó que apresentar o menor valor de  $\hat{f}(n)$  será o próximo nó a ser expandido. Para o nó inicial definido como  $n_i$  e o conjunto de nós objetivo definido como  $T$ , a algoritmo de busca é descrito pelo algoritmo da Figura 3.7.

O operador  $Menor_{\hat{f}}(N)$  é responsável por selecionar o nó de menor valor de  $\hat{f}$  do conjunto

de nós  $N$ .

---

**Algoritmo 1: A\***


---

**Entrada:**  $n_i, n_o$   
**Saída:** Caminho

```

1  $F \leftarrow \emptyset$  ;
2  $A \leftarrow \emptyset + n_i$  ;
3  $\hat{f}(A)$  ;
4 enquanto  $A \neq \emptyset$  faça
5   |  $n = \text{Menor}_{\hat{f}}(A)$ ;
6   |  $F \leftarrow F + n$ ;
7   | se  $n \in T$  então
8   |   | Fim;
9   | senão
10  |   |  $A \leftarrow A + \Gamma(n)$ ;
11  |   |  $\hat{f}(A)$  ;
12  |   fim
13 fim
```

---

Figura 3.7: Algoritmo de busca A\*

O algoritmo de busca pode ser descrito basicamente pelos seguintes passos:

1. Mover o nó  $n_i$  para o conjunto de nós abertos  $A$ .
2. Seleciona o nó  $n$  do conjunto de nós abertos  $A$  que apresentar o menor valor de  $\hat{f}$ . No caso de valores iguais resolver de forma arbitrária sempre em favor de  $n \in T$
3. Se  $n \in T$  move  $n$  para o conjunto de nós fechados  $F$  e termina o algoritmo.
4. Se não, move  $n$  para o conjunto  $F$  e aplica o operador  $\Gamma$  para  $n$ . Calcula  $\hat{f}$  para todos os nós sucessores de  $n$  e move para o conjunto  $A$  todos os nós sucessores que não pertencerem ao conjunto  $F$ . Os nós sucessores que pertencem a  $F$  e apresentam um novo  $\hat{f}$  menor são movidos para o conjunto  $A$ . Volta ao passo 2.

Com uma heurística correta no cálculo da função de avaliação  $\hat{f}$  o algoritmo A\* vai encontrar um caminho ótimo ligando o nó inicial a um nó de destino [Hart, Nilsson e Raphael 1968].

Em alguns problemas é mais importante obter soluções válidas com um baixo tempo de resposta do que uma solução ótima mas com um alto gasto de tempo para ser encontrada. Para esses casos o algoritmo A\* teve uma alteração, permitindo que não apenas a solução ótima seja determinada mas que também um número limitado das melhores soluções válidas encontradas seja mostrado.

## 4 Arquitetura do Sistema de Controle

### 4.1 Introdução

Arquitetura de *software* é uma metodologia para estruturar o algoritmo determinando como os controles reativo, deliberativo e executivo vão ser combinados para utilizar as características de cada um deles [Russell e Norvig 2003].

O *Sistema de Controle* é a parte do *Sistema de Navegação* responsável por interpretar os *Valores dos Sensores* passados pelo *Veículo Automatizado* para através deles determinar os *Valores de Atuação* necessários para fazer com que o *Veículo Automatizado* alcance um estado objetivo pré estabelecido.

A arquitetura desenvolvida para o *Sistema de Controle* é uma arquitetura híbrida de três camadas de controle, deliberativa, executiva e reativa e uma camada de responsável pelo gerenciamento das outras, chamada de coordenadora. A Figura 4.1 mostra a estrutura desenvolvida para o *Sistema de Controle*.

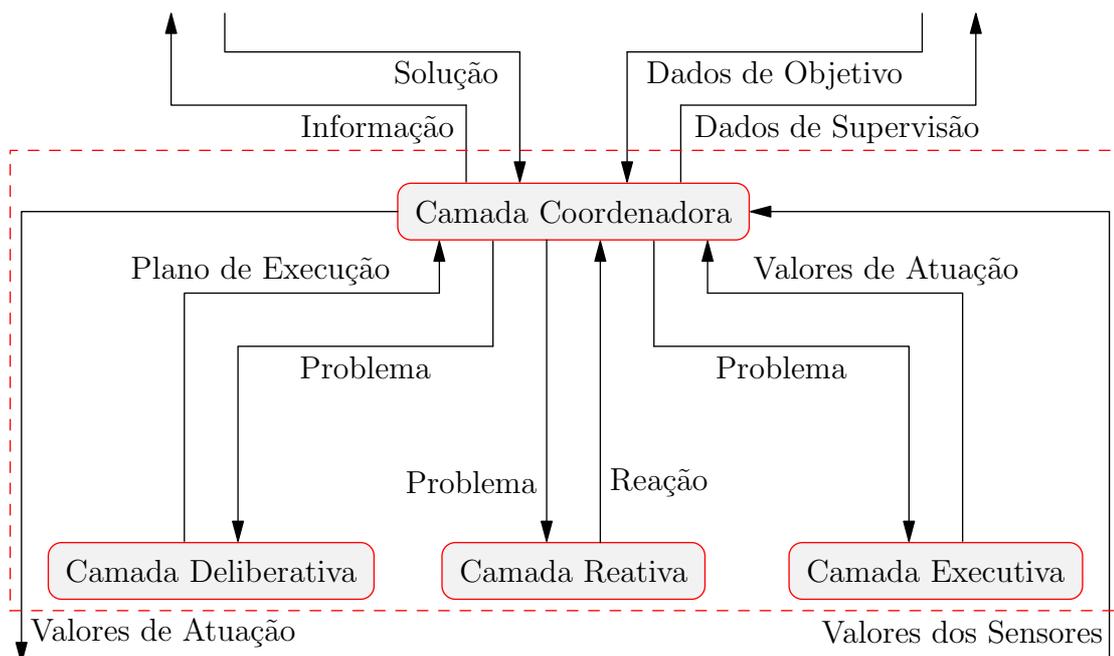


Figura 4.1: Estrutura do *Sistema de Controle*

Cada camada é responsável por uma etapa do *software* de navegação, as camadas são complementares e o resultado de uma camada é utilizado por outra mas são executadas de forma independente.

Em cada uma das camadas precisa existir um grupo de *Controladores* implementados para o *Sistema de Controle* ser capaz de determinar os *Sinais de Atuação* necessários para o *Veículo Autônomo Inteligente* ser capaz de alcançar os objetivos desejados.

Neste trabalho um *Controlador* é um sistema de *software* capaz de solucionar um *Problema* específico do *Sistema de Controle*, normalmente a solução desse *Problema* é uma etapa necessária para alcançar um estado objetivo passado pela *Interface com Usuário*.

Um *Problema* é normalmente definido simplesmente com um estado objetivo que deve ser alcançado a partir de um estado inicial, cada problema possui características próprias que definem qual dos *Controladores* implementados é responsável por determinar sua solução.

A solução encontrada por um *Controlador* é uma informação capaz de gerar *Valores de Atuação* ou um novo *Problema* capaz de ser solucionado por algum dos *Controladores*.

## 4.2 Camada Coordenadora

A *Camada Coordenadora* é responsável por determinar qual dos *Controladores* pertencentes a uma das três outras camadas de controle será executado para solucionar determinado *Problema*.

No caso da necessidade de mais de um *Controlador* ser executado para a solução de um *Problema*, é a *Camada Coordenadora* quem determina a ordem de execução e como os *Controladores* vão interagir entre si.

Portanto a *Camada Coordenadora* precisa conhecer cada um dos *Controladores* implementados nas três camadas de controle, só dessa forma a pode determinar qual *Controlador* utilizar e como modelar o *Problema* de forma adequada e interpretar a *Solução* encontrada pelo *Controlador*.

O estado atual do *Veículo Autônomo Inteligente* é determinado pela *Camada Coordenadora* através dos *Valores dos Sensores* passados pelo *Veículo Automatizado* mais o estado interno do *Sistema de Controle*. Essas informações são analisadas para determinar os mais prováveis estados atuais do *Veículo Autônomo Inteligente* e do *Ambiente*, esses estados são utilizados por todo o *Sistema de Controle*.

Através do estado objetivo passado pela *Interface com Usuário* e a determinação do estado atual a *Camada Coordenadora* determina quais dos *Controladores* implementados nas camadas de controle serão executados para alcançar o estado objetivo a partir do estado atual.

### 4.3 Camada Deliberativa

A camada deliberativa gera soluções globais para tarefas complexas utilizando planejamento e modelos de tomada de decisão que são pré determinados e utiliza outras camadas para obter informações sobre os estados do *Ambiente* e do *Veículo Automatizado*

[Russell e Norvig 2003].

Os *Controladores* da *Camada Deliberativa* recebem *Problemas* que normalmente necessitam de soluções complexas, e basicamente quebram o *Problema* inicial em um grupo de *Problemas* mais simples resultando no *Plano de Execução*.

Os *Planos de Execução* gerados são interpretados pela *Camada Coordenadora* e são modelados na forma de novos *Problemas* até que possam ser solucionados por um controlador da *Camada Executiva* na forma de *Valores de Atuação*.

### 4.4 Camada Reativa

A camada reativa é de mais baixo nível e trabalha em tempo real utilizando informações de sensores, realizando um simples controle realimentado que normalmente é utilizado em caso de estados não previstos que podem resultar em algum tipo de risco [Russell e Norvig 2003].

Devido a suas características a *Camada Reativa* tem prioridade sobre as outras camadas, é executada por eventos gerados pela *Camada Coordenadora*. Seus *Controladores* realizam operações de resposta rápida reagindo a situações que não podem ser tratadas por *Controladores* pertencentes as outras camadas.

Os resultados dos *Controladores* da *Camada Reativa* são chamados de *Reação* devido a característica principal de reagir de forma rápida a situações não previstas pelo *Sistema de Controle*. Uma *Reação* é normalmente um *Valores de Atuação* responsável por interromper o atual comportamento do *Veículo Autônomo Inteligente*.

## 4.5 Camada Executiva

A *Camada Executiva* possui os *Controladores* responsáveis por gerar *Valores de Atuação* que permitem ao *Veículo Automatizado* alcançar algum estado desejado em relação ao *Problema* passado pela *Camada Coordenadora*.

Cada *Controlador* pertencente a *Camada Executiva* é projetado para solucionar algum tipo específico de *Problema* que para ser solucionado é necessário uma interpretação do estado atual do *Ambiente* e do *Veículo Autônomo Inteligente*. São *Controladores* executados em tempo real gerando mudanças de estado e reagindo a elas de forma a gerar os *Valores de Atuação* necessários para solucionar o *Problema*.

## 5 *Simulador*

### 5.1 Introdução

Muitos dos testes necessários ao desenvolvimento do *Sistema de Navegação* não podem ser realizados diretamente no *Veículo Automatizado* devido a dificuldades tanto em relação ao acionamento quanto a necessidade de um ambiente controlado. Portanto um *software* de simulação com comportamento coerente ao do *Veículo Automatizado* foi desenvolvido para facilitar todo o desenvolvimento do *Sistema de Navegação*.

O *Modelo Cinemático* utilizado como um dos *Recursos Compartilhados* também pode ser utilizado no desenvolvimento de um *software* de simulação do *Veículo Automatizado*. Com o desenvolvimento do *Simulador* o *Modelo Cinemático* pode ter seus parâmetros otimizados de forma que seu *Comportamento* seja o mais próximo do *Comportamento* do *Veículo Automatizado*.

*Comportamento* do *Veículo Automatizado* é como uma sucessão de estados onde o estado atual é uma sucessão do estado anterior. Dessa forma o *Comportamento* do *Veículo Automatizado* pode ser descrito por uma série de vetores de posicionamento  $\vec{P}$  descrevendo os estados intermediários para se ir de um estado inicial a um estado final.

O *Simulador* é validado utilizando dados do *Comportamento* do *Veículo Automatizado*, Uma vez com o *Simulador* validado ele é usado para demonstrar a maior parte dos resultados obtidos neste trabalho.

### 5.2 Estrutura do Simulador

#### 5.2.1 Introdução

Pela definição, agente é qualquer coisa que pode perceber seu ambiente através de sensores e agir sobre o ambiente através de atuadores [Russell e Norvig 2003]. Portanto o *Veículo Automatizado* pode ser considerado um agente e sua simulação é desenvolvida como um sistema de

simulação de um agente.

A simulação é desenvolvida para descrever o comportamento de um agente que pode interagir com o ambiente em que está inserido, tendo como consequência dessa interação a alteração dos estados do agente e do ambiente que é percebida através de sensores.

O *Simulador* modela o *Comportamento* do *Veículo Automatizado* e o *Ambiente* em que ele está inserido, recebe do *Sistema de Navegação* os *Valores de Atuação* e devolve os *Valores dos Sensores* com informações sobre o estado do *Veículo Automatizado* e do *Ambiente*. A Figura 5.1 mostra essa estrutura.

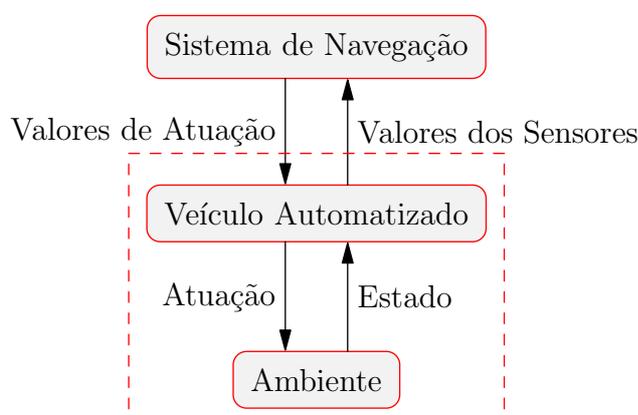


Figura 5.1: Estrutura do *Simulador*

O *Simulador* recebe como variáveis de entrada a posição inicial do *Veículo Automatizado* e os *Valores de Atuação* que são traduzidos em velocidade longitudinal e ângulo das rodas dianteiras. Os *Valores de Atuação* variam no tempo enquanto o *Simulador* está sendo executado gerando o *Comportamento* resultante que é traduzido na forma de *Valores de Sensores*.

O *Sistema de Navegação* é implementado sobre o *Simulador* da mesma forma que quando implementado sobre o *Veículo Automatizado*, tendo as mesmas saídas como *Valores de Atuação* e recebendo de volta os mesmos *Valores dos Sensores* que receberia se estivesse implementado no *Veículo Automatizado* nas mesmas condições de valores de estado.

### 5.2.2 Algoritmo

A implementação do algoritmo de simulação começa com a definição do ambiente através de uma classe estática chamada de *Ambiente*, onde uma instância da classe *Veículo* chamada de *Veículo Simulado* está inserida. O *Ambiente* é uma representação matemática do sistema de coordenadas globais das posições do *Veículo Simulado* e das instâncias da classe *Objeto* contidos no *Ambiente*.

A classe *Veículo* é a parte do algoritmo responsável por simular todo o *Comportamento* do *Veículo Automatizado*, que é medido através de sensores. O *Veículo Simulado* atua sobre o *Ambiente* através de atuadores podendo dessa forma alterar seu estado e do *Ambiente*. A Figura 5.2 mostra um diagrama de classes simplificado dessa arquitetura.

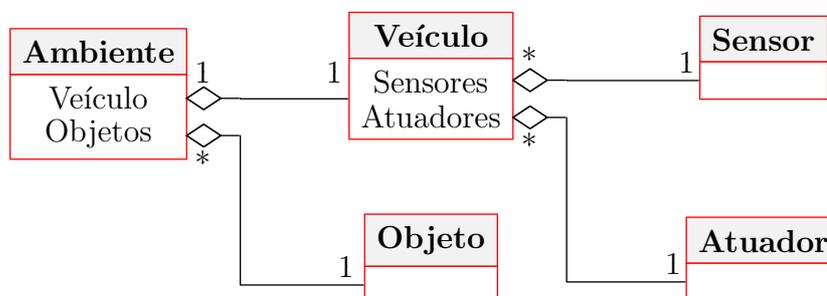


Figura 5.2: Diagrama de Classes Simplificado

O *Ambiente* é responsável por armazenar a posição do *Veículo Simulado* e de todos os *Objetos*, o *Veículo Simulado* conhece a sua posição e a posição de cada *Objeto* através das medidas dos sensores, portanto é uma informação indireta que pode ser implementada contendo erros.

A classe *Veículo* modela o *Comportamento* do veículo através do *Modelo Cinemático* e cada um dos sensores e atuadores que fazem parte do *Veículo Automatizado* são modelados, portanto o *Veículo Simulado* simula o *Comportamento* do *Veículo Automatizado*.

O *Veículo Simulado* é responsável por tratar de forma apropriada os *Valores de Atuação* que são aplicados sobre o *Modelo Cinemático* resultando em alterações nos estados do *Veículo Simulado* e do *Ambiente*. Além de traduzir os estados do *Ambiente* e do *Veículo Simulado* através das *Informação dos Sensores* que são lidas pelo *Sistema de Navegação*.

No algoritmo, sensores são instâncias de classes filhas da classe *Sensor* que modelam sensores reais fixados em posições específicas no *Veículo Automatizado* que geram sinais com determinada frequência de amostragem representando informações sobre o estado do *Veículo* ou o estado do *Ambiente* chamadas de *Informação dos Sensores*.

*Informação dos Sensores* geradas pelas instâncias da classe *Sensor* podem apresentar um erro correspondendo ao erro real dos sensores modelados, portanto a percepção do estado pelo *Veículo Simulado* pode conter erro.

O erro é uma forma de fazer com o que o *Veículo Simulado* corresponda mais precisamente ao *Veículo Automatizado*. Dessa forma para o *Sistema de Controle* acionar o *Veículo Simulado* precisa ter a capacidade de lidar com esses erros de medições assim como no *Veículo Automatizado*.

A Figura 5.3 mostra um diagrama de classes simplificado dos sensores implementados no *Simulador*.

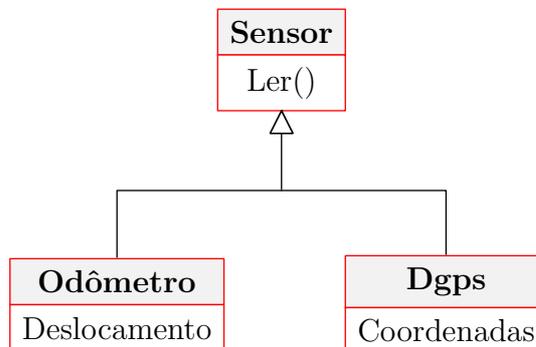


Figura 5.3: Diagrama de Classes simplificado dos sensores

Todas as as classes sensores são filhas da classe *Sensor* e herdam o método *Ler()* que é responsável por determinar o sinal de cada um dos sensores para os estados do *Veículo Simulado* e do *Ambiente*. O pseudo código desse método é mostrado na Figura 5.4.

---

#### Algoritmo 2: Ler

---

**Saída:** Sinais

- 1 Tratar Frequência de Amostragem;
- 2 **se** *Sensor Pronto* **então**
- 3     Verificar Estados;
- 4     Atualizar Sinais;
- 5 **fim**
- 6 Fim;

---

Figura 5.4: Método de Ler Sensores

Esse método é responsável por realizar todas as etapas de simulação do sensor real, que é basicamente: tratar a frequência de amostragem de forma que a leitura só seja realizada respeitando a frequência de amostragem do sensor real; verificar os estados que influenciam a leitura do *Sensor* seja do *Veículo Simulado* ou do *Ambiente*; atualizar os sinais de saída do sensor com a leitura dos estados e enviar esse sinal como resultado do método.

Os atuadores são objetos que tem acesso a algum método da classe *Veículo* responsável por alguma mudança no estado do veículo ou no estado do ambiente. Figura 5.5.

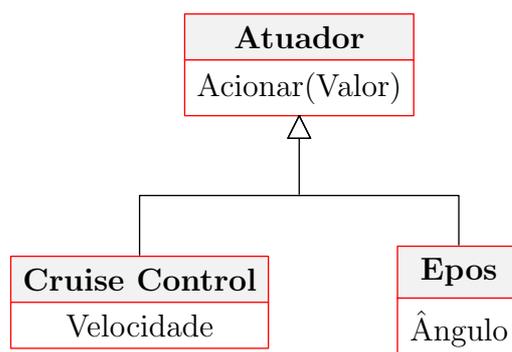


Figura 5.5: Diagrama de Classes Simplificado dos Atuadores

## 5.3 Otimização

### 5.3.1 Introdução

Um dos mais fundamentais princípios da natureza é a busca por um estado ótimo, com início no microcosmos onde átomos tentam formar ligações para minimizarem a energia de seus elétrons. Nesse caso, esse princípio fundamental é regido puramente por leis físicas, não necessita de nenhuma interferência para alcançar seu estado ótimo que pode ser no sentido de minimizar perdas ou de maximizar ganhos [Weise 2009].

Neste trabalho a otimização se dá no sentido de otimizar certos *Parâmetros* no *Modelo Cinemático* de forma que seu *Comportamento* esteja o mais próximo possível do *Comportamento* do *Veículo Automatizado*. Esses *Parâmetros* otimizados estão todos relacionado as características físicas do *Veículo Automatizado* como centro de gravidade, distancia de entre eixos e diferenças nos arcos de curvaturas para direita e para esquerda.

Portanto o objetivo da otimização é o de minimizar a diferença do *Comportamento* do *Modelo Cinemático* em relação ao *Comportamento* do *Veículo Automatizado*, minimizando erros de simulação. Isso é realizado através de um algoritmo de otimização global, que utilizando heurísticas consegue encontrar valores ótimos que podem estar em qualquer ponto do espaço de busca.

O objetivo da otimização global é encontrar os melhores elementos possíveis  $x^*$  de um conjunto  $\mathbb{X}$  de acordo com um conjunto de critérios  $F = \{f_1, f_2, \dots, f_n\}$  que são definidos através de funções de avaliação  $f: \mathbb{X} \mapsto Y$  com  $Y \subseteq \mathbb{R}$ . O domínio  $\mathbb{X}$  de  $f$  é chamado de espaço de busca e pode ser representado por qualquer tipo de elemento como números reais, listas, planos e assim por diante [Weise 2009].

### 5.3.2 Metodologia

O *Comportamento do Veículo Automatizado* é medido através da atuação de um *Controlador* implementado no *Sistema de Controle*, o desenvolvimento desse *Controlador* não faz parte deste trabalho e é apenas discutido brevemente.

A arquitetura de otimização foi desenvolvida de forma que o *Otimizador* tenha acesso tanto ao *Sistema de Navegação* quanto ao *Simulador*. A Figura 5.6 mostra essa estrutura.

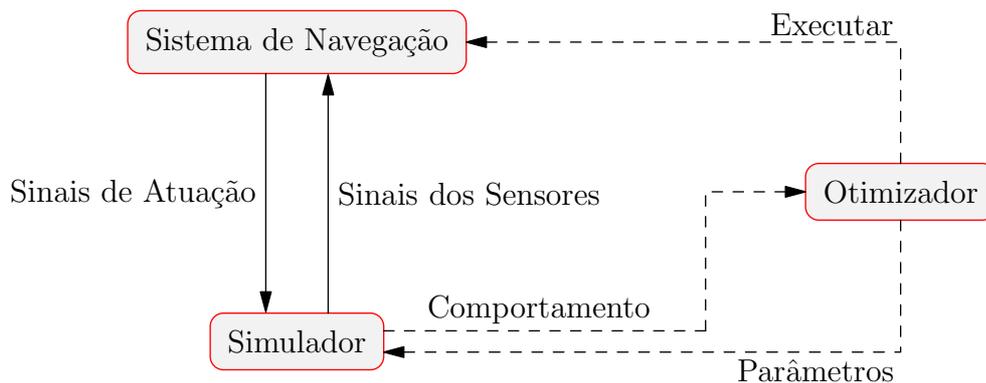


Figura 5.6: Estrutura da Metodologia de Otimização

O processo de otimização inicia com o *Otimizador* passando os *Parâmetros* para o *Simulador* que atualiza o *Modelo Cinemático*, depois é executado o *Sistema de Navegação* sobre o *Simulador* para as mesmas condições de quando foi aplicado sobre o *Veículo Automatizado*. O *Simulador* devolve ao *Otimizador* o *Comportamento* resultante para ser avaliado e comparado ao *Comportamento* do *Veículo Automatizado*. A cada final de ciclo novos *Parâmetros* são gerados e o ciclo é reiniciado.

### 5.3.3 Estrutura do Otimizador

A otimização é formada por três blocos, o *Algoritmo* de otimização a *Função de Avaliação* e o *Critério de Parada*. A Figura 5.7 mostra essa estrutura.

Neste trabalho o *Algoritmo* de otimização é responsável por determinar *Parâmetros* que apresentam melhores resultados no *Comportamento* resultante do *Simulador* em relação ao *Comportamento* do *Veículo Automatizado*. A avaliação desses *Parâmetros* é realizada pela *Função de Avaliação*, e quando o *Algoritmo* alcançar um resultado dentro de uma margem de erro ou tempo predeterminados pelo *Critério de Parada* o resultado é aceito.

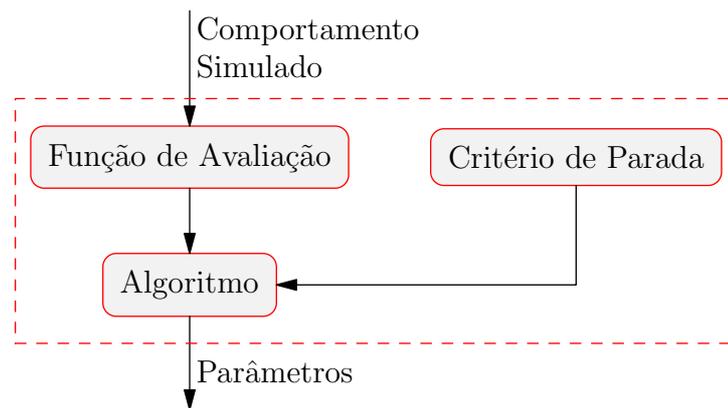


Figura 5.7: Estrutura do Otimizador

### Função de Avaliação

O resultado da função de avaliação  $v(x) \in \mathbb{V}$  para um elemento  $x$  de um espaço de busca  $\mathbb{X}$ , corresponde a utilidade desse valor como solução ou sua prioridade nos passos subsequentes do processo de otimização [Weise 2009].

O *Veículo Automatizado* e uma instância da classe *Veículo* com uma mesma configuração de estados são submetidos ao mesmo *Sistema de Controle* com um estado objetivo, a *Função de Avaliação* é responsável por avaliar a diferença de *Comportamento* resultante.

A *Base de Dados* é uma estrutura de dados que armazena o *Comportamento* do *Veículo Automatizado* e os *Sinais de Acionamento* resultantes do *Sistema de Controle* que foi submetido ao *Veículo Automatizado*, com os intervalos de tempo entre a inicialização do *Sistema de Controle* e cada *Sinal de Acionamento*.

A estrutura recebe como entrada o *Comportamento* resultante do *Simulador* e possui um bloco chamado de *Avaliação* que é responsável por determinar a diferença entre o *Comportamento* simulado e o *Comportamento* da *Base de Dados*. A Figura 5.8 mostra a estrutura da *Função de Avaliação*.

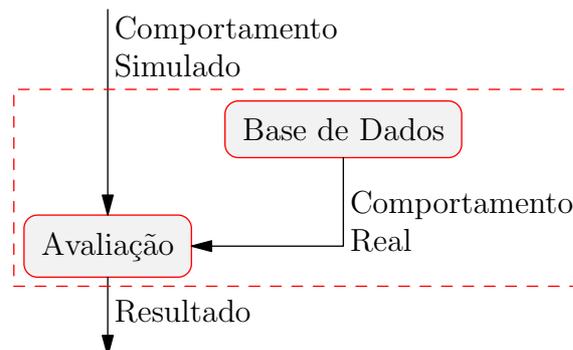


Figura 5.8: Estrutura da Função de Avaliação

A configuração de estados iniciais é basicamente o vetor posição inicial  $\vec{P} = [Px, Py, \psi]$  do *Veículo Automatizado* que também é utilizado como vetor inicial do *Veículo*. O *Comportamento* do *Veículo Automatizado* e do *Veículo* são representados respectivamente por  $\vec{P}_r(k)$  e  $\vec{P}_s(k)$ .

Dessa forma o erro pode ser determinado pelas equações:

$$E_\psi(k) = \left| \psi_r(k) - \psi_s(k) \right| \quad (5.1)$$

$$E_d(k) = \sqrt{\left( Px_r(k) - Px_s(k) \right)^2 + \left( Py_r(k) - Py_s(k) \right)^2} \quad (5.2)$$

$$E_{total} = \sum_{k=1}^N \left( K_\psi E_\psi(k) + K_d E_d(k) \right) \quad (5.3)$$

Na equação (5.3) as constantes  $K_\psi$  e  $K_d$  são utilizadas para determinar a diferença de peso que vão receber respectivamente o resultados das equações (5.1) e (5.2). O valor resultante de  $E_{total}$  é responsável por medir a diferença de *Comportamento* entre o *Veículo Automatizado* e o *Veículo* e dessa forma avaliar os *Parâmetros* passados ao *Modelo Cinemático*.

### Algoritmo

Devido a suas características o *Algoritmo* de otimização utilizado neste trabalho é um algoritmo genético.

Algoritmos genéticos pertencem ao grupo de algoritmos chamados de evolucionários, que são algoritmos de otimização meta heurística baseados em populações de indivíduos que usam mecanismos naturais como mutação, cruzamento, seleção natural e a sobrevivência do mais apto para se obter a melhor solução [Weise 2009].

Cada indivíduo do *Algoritmo Genético* é formado por um vetor definido como  $\vec{I} = [V_1, V_2, V_3, \dots, V_N]$  contendo em cada uma de suas  $N$  posições um parâmetro do *Modelo Cinemático* que será otimizado. Todas as operações de otimização são executadas diretamente no vetor que representa cada um dos indivíduos.

O espaço de busca representa os limites de variação que cada posição do vetor  $\vec{I}$ . O espaço de busca é representado por uma matriz  $E$  de tamanho  $2 \times N$  onde  $N$  é o número de posições do vetor  $\vec{I}$ , dessa forma as posições  $E_{1,k}$  e  $E_{2,k}$  representam respectivamente os valores mínimo e máximo que  $V_k$  pode apresentar.

O *Algoritmo Genético* desenvolvido pode ser basicamente dividido em três operações do processo de otimização: *Seleção*, *Cruzamento* e *Mutação*. A Figura 5.9 mostra a estrutura do *Algoritmo Genético*.

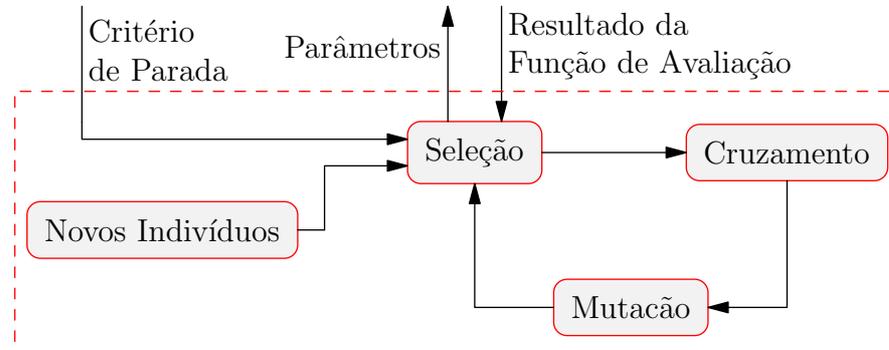


Figura 5.9: Estrutura do Algoritmo Genético

O *Algoritmo Genético* se inicia com a criação de novos indivíduos que formam a população. Toda a população passa pelo processo de *Seleção* depois alguns indivíduos são selecionados para passarem pelos processos de *Cruzamento* e *Mutação*, quando a população inicial volta ao processo de *Seleção* os piores indivíduos são descartados e substituídos por novos completando uma geração, todo esse processo é repetido até alcançar o *Critério de Parada*.

Na operação de *Seleção* cada indivíduo é avaliado se está dentro do espaço de busca, se o indivíduo estiver fora do espaço de busca é deslocado para a borda mais próxima de seu valor, e tem seu valor de erro  $E_{total}$  determinado.

O cálculo do erro  $E_{total}$  é requisitado pela *Função de Avaliação* para cada um dos indivíduos da população. Os *Parâmetros* que determinam cada indivíduo são enviados ao *Simulador* resultando em um *Comportamento* decorrente desses parâmetros que é enviado a *Função de Avaliação*, obtendo um valor de  $E_{total}$  para cada um dos indivíduos que são classificados com base nesses valores pela etapa de *Seleção*.

Alguns dos indivíduos são selecionados e passam pelo processo de *Cruzamento* onde as características de cada indivíduo serão combinadas para gerar um novo indivíduo filho que vai substituir o pior dos que realizaram o cruzamento. A Figura 5.10 demonstra o processo de *Cruzamento* para dois indivíduos  $\vec{I}_a$  e  $\vec{I}_b$  gerando o indivíduo filho  $\vec{I}_f$ , onde o indivíduo  $\vec{I}_b$  é pior que o indivíduo  $\vec{I}_a$ .

O processo de *Cruzamento* é realizado escolhendo dois indivíduos da população onde o pior deles terá pelo menos o valor de uma de suas posições alterado, cada posição é escolhida de forma aleatória e sendo  $k$  a posição escolhida temos o novo valor definido como:



variar de 1 até  $N$ , onde  $N$  é o número total de posições de cada indivíduo, assim como o valor da constante  $Km$ .

### **Critério de Parada**

O *Critério de Parada* é basicamente um contador que monitora a quantidade de gerações do *Algoritmo Genético*, baseado nessa contagem é gerado um sinal indicando o fim da otimização.

Existem dois contadores, um que monitora a quantidade de gerações totais e outro que monitora a contagem de gerações sem que um novo melhor indivíduo foi encontrado, com uma combinação desses dois contadores é gerado um sinal de parada ao *Algoritmo Genético*.

### **5.3.4 Resultados da Otimização**

O processo de otimização se inicia com a aquisição dos dados que formam a *Base de Dados*, que foi realizada através da aquisição de *Waypoints* e sinais de atuação do *Sistema de Controle*, durante um circuito realizado pelo *Veículo Automatizado*.

O *Controlador* implementado no *Sistema de Controle* utilizado para a otimização do *Modelo Cinemático* funciona basicamente como um seguidor de *Waypoints*, onde existe um circuito formado por uma lista de *Waypoints* que é a rota desejada. O *Sistema de Controle* baseado na posição atual determina os *Sinais de Atuação* que são necessários para alcançar o próximo *Waypoint* do circuito.

O *Controlador* nesse caso foi especificado de forma que a atuação foi executada de forma discreta e com uma frequência fixa de  $1Hz$  igual a frequência aquisição configurada para o DGPS, para a aquisição dos dados da *Base de Dados* foram utilizados dois circuitos um no sentido anti-horário e o outro no sentido horário. A *Base de Dados* decorrente do comportamento do *Veículo Automatizado* está representada na Figura 5.12.

A *Base de Dados* é formada por uma matriz para cada circuito com um número de linhas igual ao número de *Waypoints* e as colunas são formadas por informações de tempo decorrido desde o início da operação do *Sistema de Controle*, a posição do *Veículo Automatizado* e o *Sinal de Atuação* utilizado para alcançar o próximo ponto da *Base de Dados*.

Com essas informações é possível determinar todas as variáveis necessárias para as equações de movimento (3.16), (3.17), (3.18) e (3.19) desenvolvidas no *Modelo Cinemático*. As constantes dessas mesmas equações serão os *Parâmetros* otimizados pelo *Algoritmo Genético*.

Em cada um dos circuitos utilizados na otimização o *Veículo* recebe a mesma posição inicial

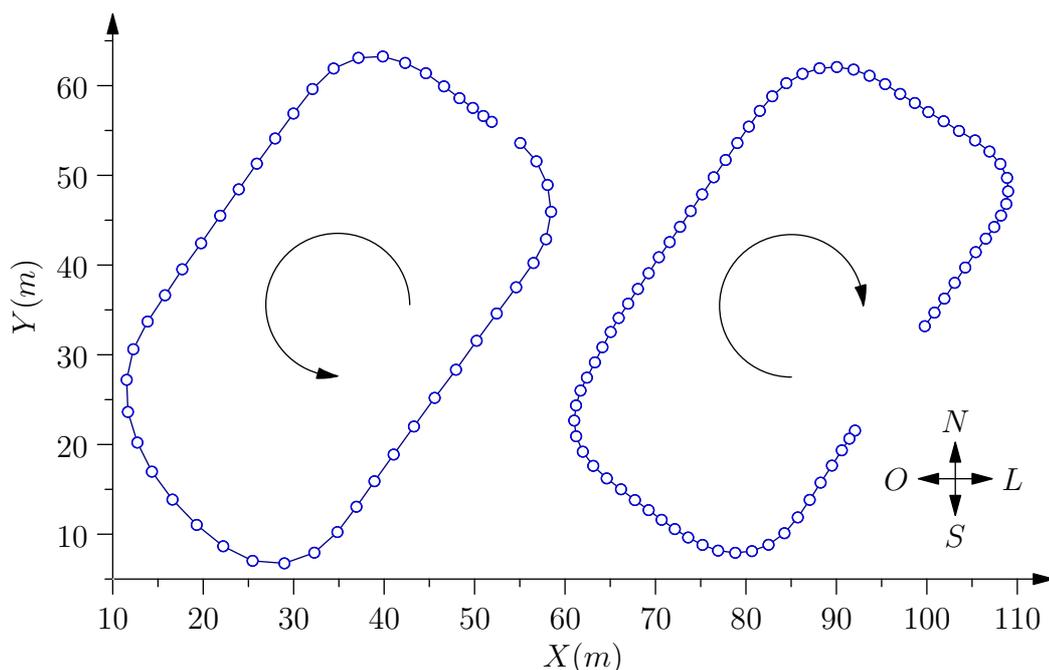


Figura 5.12: Base de Dados para Otimização

do *Veículo Automatizado*, e o *Simulador* recebe os mesmos *Sinais de Atuação* que o *Sistema de Controle* gerou para a atuação do *Veículo Automatizado* nos mesmos intervalos de tempo resultando no *Comportamento do Veículo*.

Dois circuitos foram utilizados para a *Otimização*, o primeiro no sentido anti-horário com velocidade média de  $11,48\text{km/h}$  e o segundo no sentido horário com velocidade média de  $6,86\text{km/h}$ . A Figura 5.13 mostra os gráficos do resultado da *Otimização*.

Pela análise dos resultados após a otimização, nos dois trajetos realizados o *Simulador* alcançou um resultado muito próximo do *Veículo Automatizado*. É possível perceber que no primeiro trajeto como a velocidade é mais alta ocorreu um erro maior na simulação, resultado esperado devido as características do *Simulador* já discutidas.

Com esses resultados alcançados pelo *Simulador* com a *Otimização* do *Modelo Cinemático* o *Simulador* é validado e seu *Comportamento* é considerado próximo o suficiente do *Comportamento* do *Veículo Automatizado* para ser utilizado neste trabalho.

## 5.4 Resultados

Para uma demonstração do *Simulador* é implementado no *Sistema de Controle* o mesmo *Controlador* utilizado na otimização do *Modelo Cinemático*. O circuito de *Waypoints* utilizado nessa demonstração é mostrado na Figura 5.14.

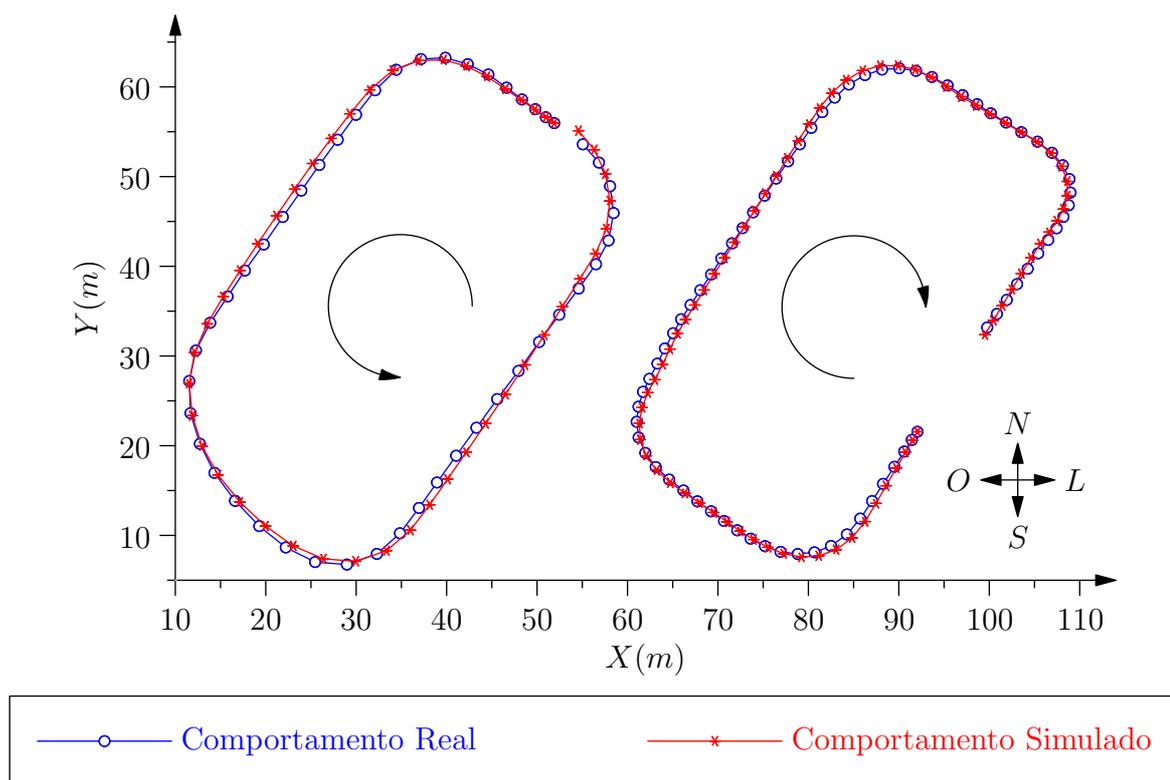


Figura 5.13: Resultado da Otimização

Pra o mesmo circuito de *Waypoints* foram simuladas seis configurações diferentes do *Controlador* com combinações de velocidades e frequências de amostragem diferentes. A Figura 5.15 mostra o *Comportamento* resultante das seis configurações.

Através dos resultados da simulação é possível uma prévia análise do *Comportamento* do *Veículo Automatizado* quando controlado através do *Controlador*. A diferença de *Comportamento* resultante da alteração nos *Parâmetros*, tanto do *Controlador* quanto dos sensores e atuadores, é facilmente percebida através dos gráficos de *Comportamento*.

Nesse exemplo mostrado é possível perceber que para uma frequência de amostragem do DGPS de  $1Hz$ , quando a velocidade do veículo é superior a  $10km/h$  o *Veículo* não é capaz de permanecer no trajeto correto dentro da rota pré determinada. Já para uma frequência de amostragem do DGPS de  $2Hz$  o *Veículo* permanece no trajeto correto até quando o veículo está a  $20km/h$ .

Quanto maior a velocidade do *Veículo* na simulação maior vai ser a diferença de *Comportamento* em relação ao *Veículo Automatizado*, para resultados mais coerentes não são realizadas simulações com a velocidade do *Veículo* superior a  $20km/h$ . Para a simulação com o *Veículo* a velocidades mais elevadas é necessária a realização de alterações no *Modelo Cinemático*, com uma revisão nas considerações realizadas discutidas anteriormente.

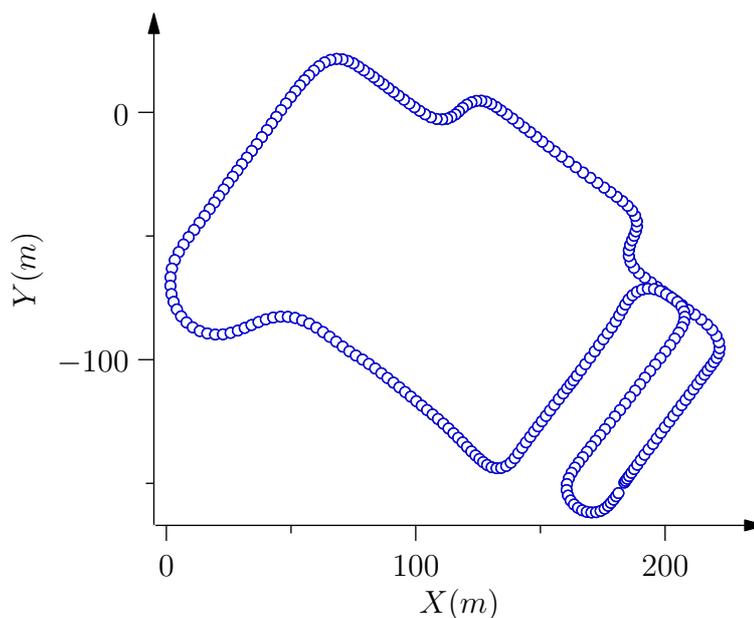


Figura 5.14: Circuito

Para uma melhor avaliação do *Simulador* um *Controlador* foi aplicado sobre o *Veículo Automatizado* para cumprir o circuito mostrado na Figura 5.15, o *Comportamento* resultante foi medido e para comparar a diferença de *Comportamento* o mesmo *Controlador* foi aplicado sobre o *Simulador* nas mesmas condições. Os *Comportamentos* resultantes real e simulado são mostrados na Figura 5.16.

O acionamento do *Veículo Automatizado* nesse caso foi realizado com uma variação na velocidade determinada por um condutor, enquanto o acionamento da instância *Veículo* a velocidade foi sempre fixa e igual a velocidade média do *Veículo Automatizado* executando o mesmo percurso. Essa diferença de acionamento é proposital devido a grande dificuldade em determinar a variação de velocidade ideal para cada trecho do circuito, dessa forma o *Simulador* é utilizado com uma velocidade fixa para a análise do *Controlador* implementado.

Pelo resultado mostrado na Figura 5.16 a diferença entre os *Comportamentos* real e simulado é baixa, existindo uma maior diferença nos trechos formados por curvas que pode ser explicada devido as diferenças no acionamento do *Veículo Automatizado* e da instância *Veículo*.

Por todos os resultados demonstrados, o *Comportamento* do *Veículo* pode não ser exatamente o *Comportamento* do *Veículo Automatizado* devido ao grande número de variáveis envolvidas em todo o sistema, mas sempre será no mínimo um comportamento coerente em condições ideais de acionamento e de ambiente.

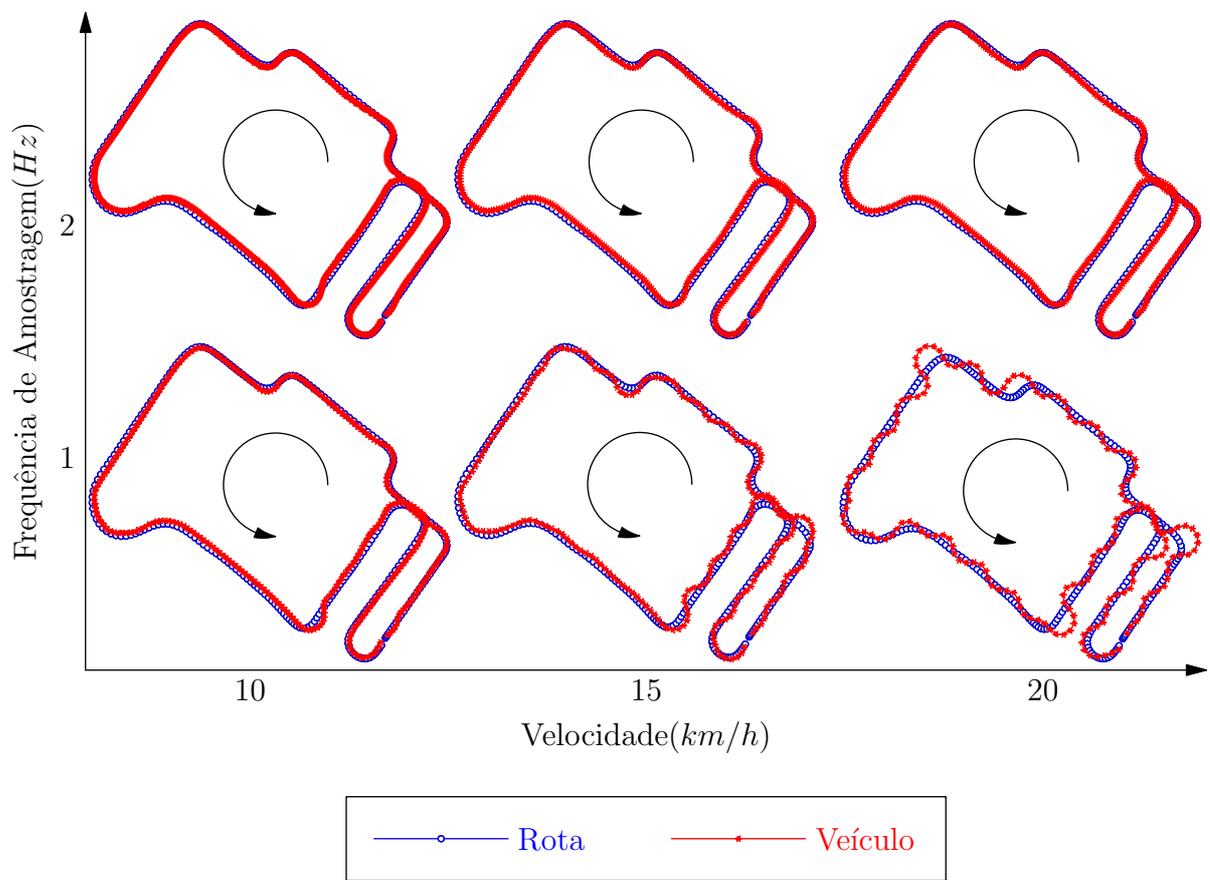


Figura 5.15: Comportamento do Veículo

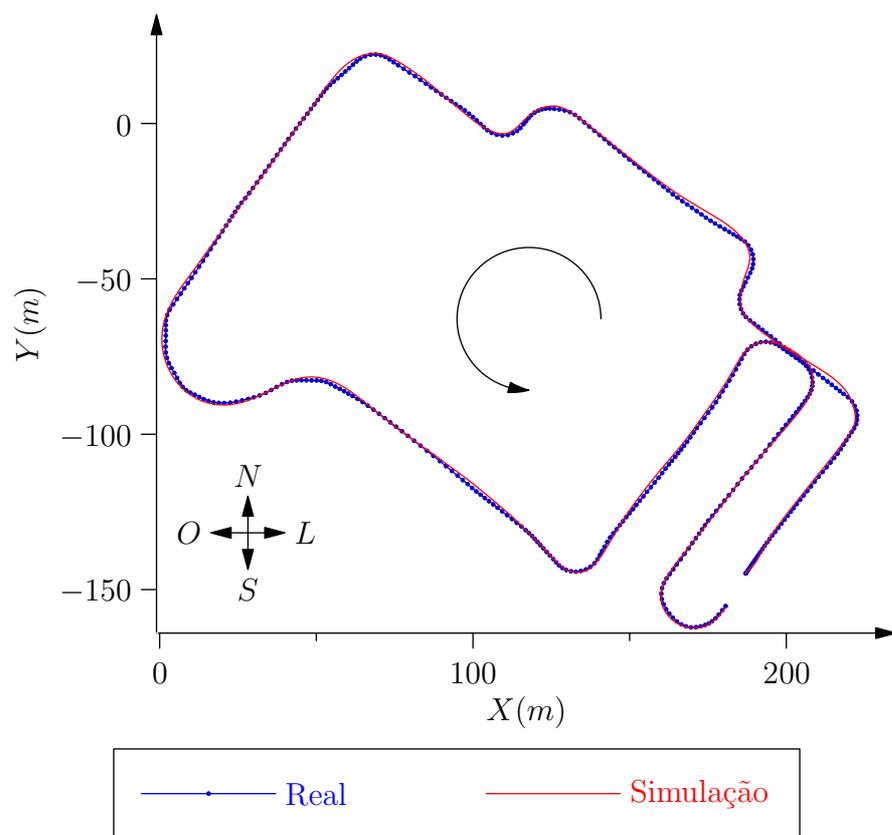


Figura 5.16: Comportamento real do Veículo

## 6 *Controladores Desenvolvidos*

### 6.1 Introdução

Neste trabalho um *Controlador* é definido como um componente fundamental para o *Sistema de Controle* responsável pela solução de problemas, e deve pertencer a uma das camadas do *Sistema de Controle*.

Se o *Controlador* pertence a *Camada Deliberativa* sua solução para um determinado problema é uma série de problemas menores que são solucionados por outro controlador. Se o controlador pertencer a *Camada Executiva* ou a *Camada Reativa* sua solução é um *Valor de Acionamento* responsável por fazer com que o *Veículo Automatizado* alcance um estado desejado.

Cada um dos *Controladores* é desenvolvido para a solução de um problema específico e pode utilizar qualquer um dos recursos disponibilizados pelo *Sistema de Navegação*, permitindo ao *Sistema de Navegação* uma flexibilidade de desenvolvimento sem uma necessidade de alteração na arquitetura desenvolvida.

Não é objetivo deste trabalho desenvolver todo o conjunto de *Controladores* necessários para uma navegação completa do *Veículo Inteligente*, mas com o *Sistema de Controle* construído alguns *Controladores* foram desenvolvidos usando os *Recursos Compartilhados* disponibilizados pelo *Sistema de Navegação*.

Esses *Controladores* desenvolvidos demonstram a utilização do *Sistema de Controle* com uma interação com os *Recursos Compartilhados* desenvolvidos, enfatizando a flexibilidade permitida pelo *Sistema de Navegação* para a solução de problemas com abordagens diferentes mas que se completam na solução de problemas complexos de navegação.

Foi desenvolvido o *Controlador Navegador Global* pertencente a *Camada Deliberativa*, que dado um número de rotas permitidas que foram pré determinadas, determina a melhor rota para o *Veículo Automatizado* iniciando em uma posição global inicial chegue a uma posição global objetivo.

O *Controlador Navegador Local*, também pertencente a *Camada Deliberativa*, determina rotas locais para se ir de uma posição local inicial a uma posição local objetivo. As posições locais inicial e objetivo são representadas por estados inicial e objetivo e através da determinação de estados alcançáveis utilizando o *Modelo Cinemático* é gerada uma sequência de estados que leva o *Veículo Automatizado* do estado inicial ao estado objetivo.

O *Controlador Seguidor de Estados* é um *Controlador* que complementa o *Controlador Navegador Local*, dado a sequência de estados determinada pelo *Controlador Navegador Local* o *Controlador Seguidor de Estados* determina os *Valores de Acionamento* necessários para levar o *Veículo Automatizado* para cada um dos estados da sequência de estados.

## 6.2 Navegador Global

### 6.2.1 Introdução

O *Navegador Global* é um *Controlador* pertencente a *Camada Deliberativa* do *Sistema de Controle* e tem como objetivo determinar uma rota que leve o *Veículo Automatizado* de uma posição global inicial para uma posição global objetivo.

Os caminhos foram modelados de forma a permitirem que existam caminhos possíveis para qualquer ponto de destino partindo de qualquer ponto de origem. Podem existir mais de uma solução possível, mas o algoritmo de busca vai determinar a solução ótima com base nas características modeladas.

Para a navegação autônoma é necessário também a localização de possíveis obstáculos que podem obstruir a passagem do veículo, e nesse caso será feito um replanejamento de rotas levando em conta a obstrução desse caminho. Todas essas informações do ambiente são obtidas com o uso de sensores [Thrun, Burgard e Fox 2005].

### 6.2.2 Modelagem

Para determinar rotas globais é necessário modelar os dados que serão utilizados pelo *Algoritmo de Busca* para determinar a melhor rota, essa estrutura foi desenvolvida através da criação de uma árvore de busca. A árvore de busca é chamada de *Circuito* e suas ramificações são *Sub Circuitos* que são formados por *Waypoints*.

*Waypoints* são pontos que representam coordenadas globais válidas e alcançáveis pelo *Veículo Automatizado*, um ou mais *Waypoints* representam um dos *Sub Circuitos* em que o

*Veículo Automatizado* pode trafegar.

O conjunto de *Sub Circuitos* forma o *Circuito* que representa os *Sub Circuitos* e suas interligações, permitindo dessa forma a construção de uma árvore de busca. Essa estrutura é mostrada na Figura 6.1.

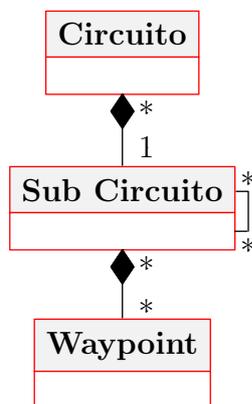


Figura 6.1: Diagrama de Classes simplificado

O *Circuito*  $C$  é formado por um conjunto de *Sub Circuitos*  $S$ , portanto pode ser representado por  $C = \{S_0, S_1, S_2, \dots, S_k\}$ . Cada um dos *Sub Circuitos*  $S$  são formados por uma sequência ordenada de *Waypoints*  $W$ , podendo ser representado por  $S = \{W_0, W_1, W_2, \dots, W_k\}$  que determinam o sentido de movimentação do *Veículo Automatizado*, que pode ser no sentido de  $W_0$  para  $W_k$  ou de  $W_k$  para  $W_0$ .

O sentido de cada *Sub Circuito* é determinado por suas interligações, na Figura 6.1 se a interligação for apenas no sentido de  $S_i$  para  $S_j$  a rota formada pelos dois *Sub Circuitos* será apenas nesse sentido, mas se também existir a interligação  $S_j$  para  $S_i$  a rota resultante permitirá a navegação nos dois sentidos.

### 6.2.3 Operador $\Gamma$

Cada *Sub Circuito* tem uma lista de *Sub Circuitos* que contem pelo menos outro *Sub Circuito* incluindo ele mesmo, esses *Sub Circuitos* são os *Sub Circuitos* sucessores. Seja o operador  $\Gamma$  responsável por determinar quais são os *Sub Circuitos* sucessores, portanto  $\Gamma(S_n) = \{S_n, S_0, S_1, \dots, S_k\}$ .

Os *Sub Circuitos* sucessores são responsáveis por determinar as ligações entre cada *Sub Circuito*, que só tem ligação com seus *Sub Circuitos* sucessores. Para existir essa ligação é necessário que o *Suc Circuito* tenha um único *Waypoint* em comum com cada um dos seus *Sub Circuitos* sucessores, esse *Waypoint* precisa necessariamente ser o primeiro *waypoint*  $W_0$  ou o

ultimo waypoint  $W_k$  de cada um dos sub circuitos.

Todos os caminhos possíveis a partir de cada *Sub Circuito* são realizados apenas por algum de seus *Sub Circuitos* sucessores. Quando é selecionado como sucessor o próprio *Sub Circuito* resulta que o *Veículo Automatizado* precisou fazer uma manobra de mudança de sentido no *Sub Circuito*.

Seja  $S_j \in \Gamma(S_i)$ , o *Waypoint* em comum entre o *Sub Circuito*  $S_i$  e o *Sub Circuito*  $S_j$  é chamado de  $W_{ij}$ , onde  $W_{ij} \in S_i$  e  $W_{ij} \in S_j$ . Se  $S_j \in \Gamma(S_i)$  e  $S_i \in \Gamma(S_j)$ , necessariamente o *Waypoint*  $W_{ji}$  será igual ao *Waypoint*  $W_{ij}$ .

O *Waypoint*  $W_{ij}$  é quem determina a ligação entre os *Sub Circuitos*  $S_i$  e  $S_j$  e o sentido dessa ligação. Onde o sentido será sempre do sub circuito  $S_i$  iniciando pelo *Waypoint* oposto ao  $W_{ij}$  e passando para o sub circuito  $S_j$  iniciando pelo *Waypoint*  $W_{ij}$ . A Figura 6.2 mostra essa ligação.

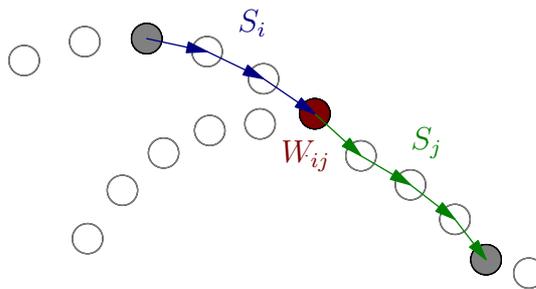


Figura 6.2: Ligação entre *Sub Circuitos*

Para a solução da rota são necessários dois pontos, um ponto inicial chamado de  $P_I$  e um ponto objetivo chamado de  $P_O$ , cada ponto é formado por um *Waypoint*  $W$  e um sentido de navegação  $\alpha$ , portanto  $P = \{W, \alpha\}$ .

O operador  $\Theta$  verifica em qual sub circuito  $S$  está um determinado ponto  $P$ , essa verificação é realizada determinando qual o *Waypoint*  $W$  é o mais próximo de  $P$  para determinar quais sub circuitos possuem  $W$  e determinar entre os sub circuitos que possuem  $W$  qual apresenta o mesmo sentido de  $\alpha$ .

#### 6.2.4 Operador $\Omega$

O objetivo da busca é encontrar um caminho ótimo que leve de um ponto inicial a um dos pontos objetivo, o operador  $\Omega(S)$  é responsável por determinar se o *Sub Circuito*  $S$  é um dos *Sub Circuitos* objetivo.

Com a utilização do operador  $\Theta(P_I)$  sobre o ponto inicial  $P_I$  para determinar o conjunto dos *Sub Circuitos* objetivo  $S_O$ , o operador  $\Omega(S)$  simplesmente verifica se  $S \in S_O$ .

### 6.2.5 Operador $\Lambda$

O operador  $\Lambda$  determina quando dois *Sub Circuitos* podem ser considerados iguais, para o *Navegador Global* existe um número limitado de *Sub Circuitos* com cada um representado um trajeto único do *Circuito* portanto um *Sub Circuito* só vai ser considerado igual a outro se for exatamente o mesmo *Sub Circuito*.

Dessa forma, o operador  $\Lambda_{S_i}(S_j)$  para verificar se o *Sub Circuito*  $S_i$  é considerado igual ao *Sub Circuito*  $S_j$  apenas verifica se  $S_i = S_j$ .

### 6.2.6 Custo $\hat{g}$

O custo  $\hat{g}$  foi definido como sendo em função da distancia necessária para percorrer cada uma das rotas, seja  $\hat{d}(S)$  o operador que determina o comprimento do sub circuito  $S$ . Com  $W_I$  o *Waypoint* inicial e  $W_O$  o *Waypoints* objetivo, se  $W_I \in S$  o comprimento será calculado tendo como ponto inicial o *Waypoint*  $W_I$  e se  $W_O \in S$  o comprimento será calculado tendo como ponto final o *Waypoint*  $W_O$ .

Considerando  $R$  o conjunto de *Sub Circuitos* que constituem

$$\hat{g}(S) = \hat{d}(\forall(S \in R)) \quad (6.1)$$

### 6.2.7 Custo $\hat{h}$

Seja  $\hat{d}_r(S)$  o operador responsável por calcular a distancia em linha reta do *waypoint* final do sub circuito  $S$  até o *waypoint* objetivo  $W_O$ . Portanto:

$$\hat{h}(S) = \hat{d}_r(S) \quad (6.2)$$

Resultando em:

$$\hat{g}(S) = \hat{d}_p(S) \quad (6.3)$$

Seja  $\hat{d}_r(S)$  o operador responsável por calcular a distancia em linha reta do *waypoint* final do sub circuito  $S$  até o *waypoint* objetivo  $W_O$ . Portanto:

$$\hat{h}(S) = \hat{d}_r(S) \quad (6.4)$$

### 6.2.8 Resultados

Foi escolhida para ser a área de navegação do *Veículo Automatizado* uma parte do *campus* da UNIFEI, que foi mapeada através de *Waypoints* capturados enquanto o *Veículo Automatizado* controlado manualmente percorria alguns percursos. A área mapeada é mostrada na Figura 6.3.



Figura 6.3: Área mapeada para o *Navegador Global*

A área de navegação foi escolhida tendo em vista o número de rotas disponíveis permitindo uma grande liberdade de navegação ao *Veículo Automatizado*. A área mapeada é utilizada por veículos normais portanto cada uma das rotas tem sentido de navegação e interligação com outras rotas, resultando em um *Ambiente* completo para ser modelado.

Os *Waypoints* foram capturados utilizando o DGPS instalado no veículo, com uma frequência de amostragem de  $1Hz$ , depois selecionados e agrupados em *Sub Circuitos* com seus respectivos *Sub Circuitos* sucessores formando o *Circuito* final. O *Circuito* mapeado é mostrado na Figura 6.4.

Uma vez com o *Circuito* mapeado a um *Navegador Global* determina a melhor rota para o veículo se mover de um ponto inicial para um ponto objetivo. Devido as características do *Veículo Automatizado* toda a busca de rotas é realizada levando em conta sua orientação e a orientação desejada no ponto objetivo, portanto a rota obtida é a que necessita de menos manobras para ser executada com base nos parâmetros modelados.

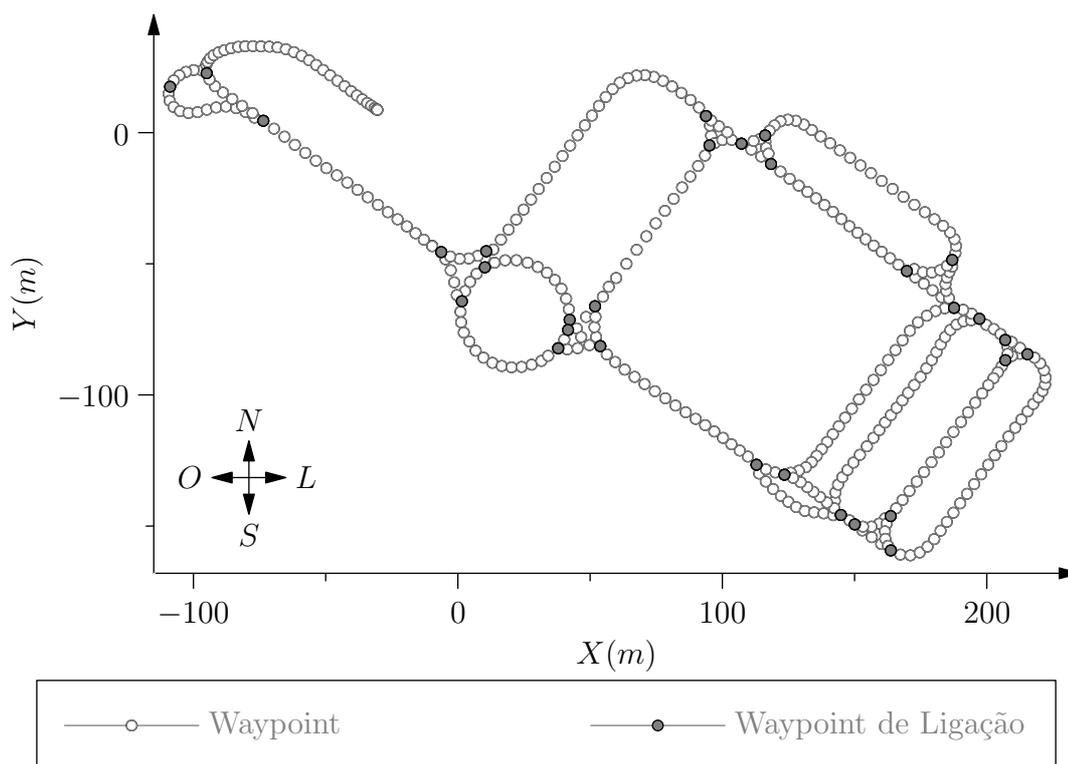


Figura 6.4: *Circuito* resultante da área mapeada para o *Navegador Global*

Para exemplificar o funcionamento do sistema de busca foi definido para o *Circuito* desenvolvido um ponto inicial e um ponto objetivo. A Figura 6.5 mostra os custos para uma etapa intermediária da busca de rotas. Onde é mostrado o custo do sub circuito  $S$  como a soma de  $\hat{g}(S)$  e  $\hat{h}(S)$ .

Na Figura 6.6 é mostrada a rota encontrada como solução pela camada deliberativa para um ponto inicial e um ponto objetivo. A rota resultante respeita o sentido correto de fluxo de veículos, chegando a um caminho ótimo para as características determinadas no sistema de custos do sistema.

Como resultado para a camada deliberativa é gerada uma lista de coordenadas de latitude e longitude que vai ser utilizada pela camada executiva para realizar a movimentação do veículo, onde a ordem com que as coordenadas são listadas determina o sentido de navegação do veículo resultando assim na rota que deve ser executada.

Devido a característica de o ambiente em que o veículo está inserido ser um ambiente parcialmente observável, existe a possibilidade de a rota planejada não poder ser executada devido a uma mudança no estado do ambiente não considerada no planejamento da rota. Nesse caso é necessário um replanejamento na rota considerando a mudança no estado do ambiente.

A Figura 6.7 mostra um replanejamento no caso em que o veículo encontra um bloqueio

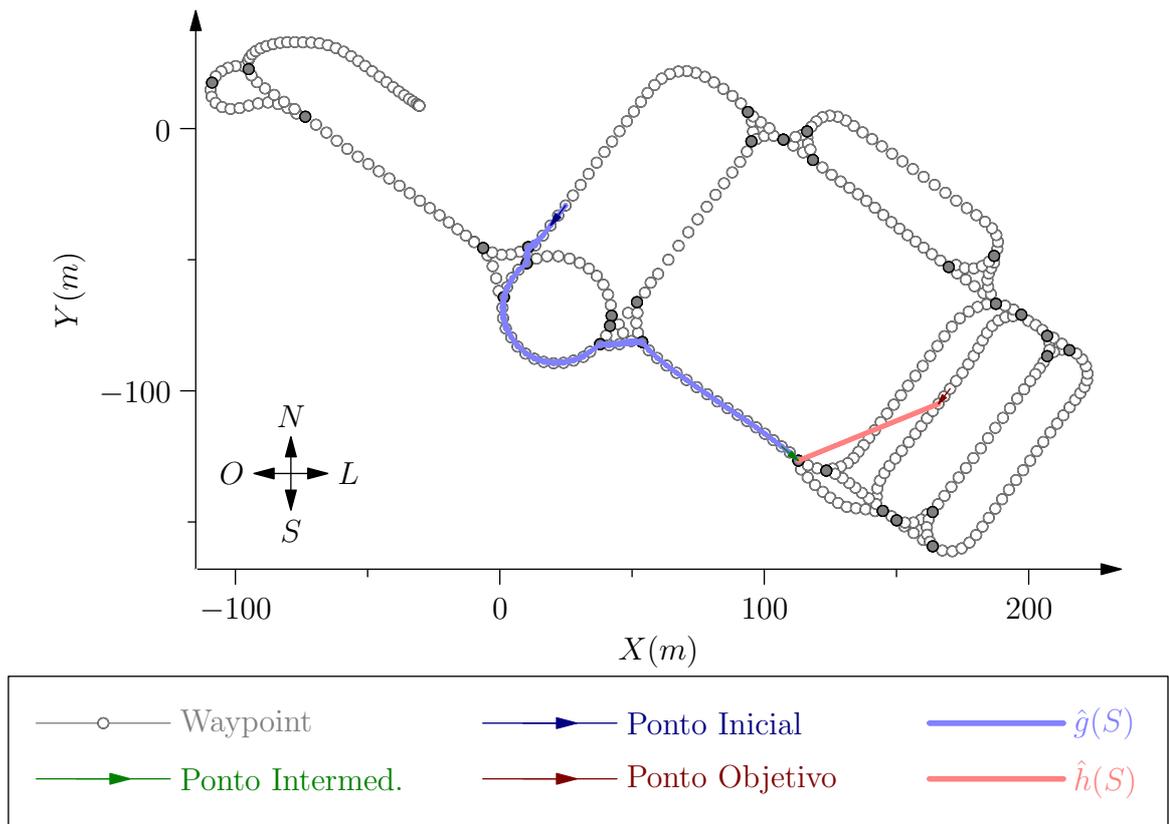


Figura 6.5: Busca de rota sendo executada pelo *Navegador Global*

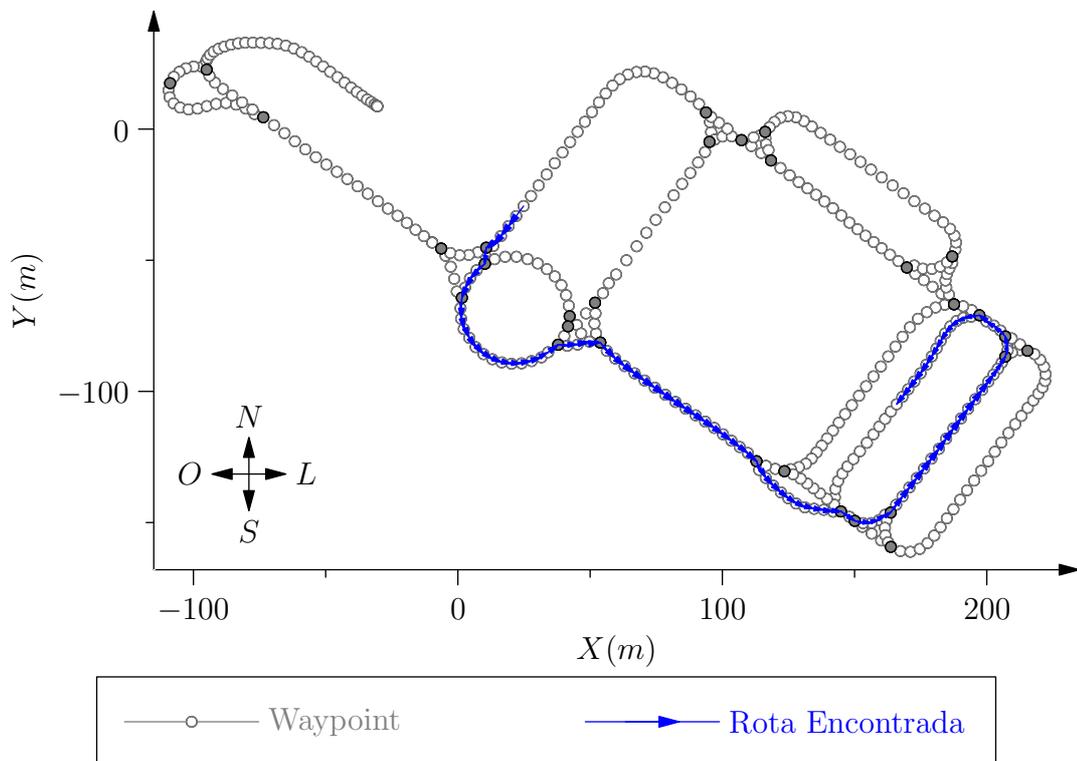


Figura 6.6: Rota resultante da busca executada pelo *Navegador Global*

que não pode ser atravessado em um segmento da rota planejada. Nesse caso quando o veículo percebe esse bloqueio a rota é replanejada tendo como ponto inicial o ponto de bloqueio e como ponto objetivo o mesmo ponto do planejamento de rota anterior.

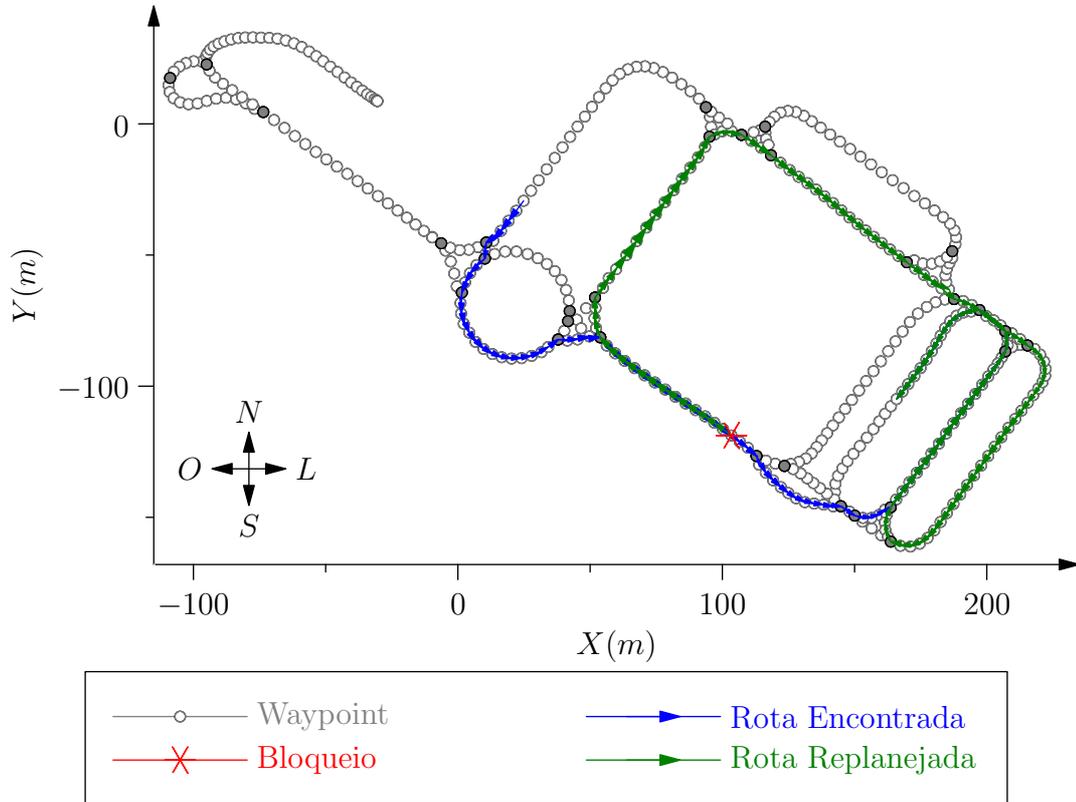


Figura 6.7: Rota replanejada pelo *Navegador Global*

Quando o sistema percebe o bloqueio em um de seus sub circuitos, o sub circuito onde se encontra o bloqueio é alterado, de forma que o sub circuito é dividido em dois, um formado do *Waypoint* inicial a um *Waypoint* anterior ao bloqueio, e um sub circuito tendo como *Waypoint* inicial o *Waypoint* a frente do bloqueio e como *Waypoint* final o *Waypoint* final do sub circuito original, onde esses dois novos sub circuitos não são inteligados devido ao bloqueio entre eles.

Essa alteração pode ser considerada pelo sistema, permanente ao ambiente ou apenas temporária dependendo do tipo de informação que será passada ao sistema sobre o bloqueio.

## 6.3 Navegador Local

### 6.3.1 Introdução

O *Navegador Local* é um *Controlador* pertencente a *Camada Deliberativa* responsável para, dado um *Ambiente* conhecido, determinar uma sequência de posições intermediárias ne-

cessárias para ir de uma posição inicial para uma posição objetivo.

Os *Recursos Compartilhados* necessários para o funcionamento do *Navegador Local* são o *Modelo Cinemático* e o *Algoritmo de Busca*.

A busca de rotas locais é executada quando existe a necessidade de navegação em um espaço de movimentação reduzido para o *Veículo Automatizado*, a partir de uma posição inicial, alcançar uma posição objetivo.

Cada posição representa um estado que o *Veículo Automatizado* pode apresentar em relação ao *Ambiente*, os estados podem ser modelados como um nó de uma árvore de busca, portanto é um problema que deve ser solucionado pelo *Algoritmo de Busca*.

Como os estados dependem do comportamento do *Veículo Automatizado* o *Modelo Dinâmico* é utilizado para a modelagem da árvore necessária para a solução do problema pelo *Algoritmo de Busca*. A modelagem da árvore é basicamente o desenvolvimento dos métodos necessários para a implementação de uma classe filha da classe *Nó* que é usada pelo *Algoritmo de Busca*.

### 6.3.2 Modelagem

Para determinar rotas de navegação local através de busca em árvore é necessária uma limitação das transições que determinam as mudanças de posicionamento do *Veículo Automatizado*. Sem essa limitação existirá a partir de cada nova posição uma explosão de posições alcançáveis, resultando em uma árvore de busca onde para cada nó terá uma explosão de nós sucessores.

Cada transição de posicionamento é representada por estados possíveis de serem alcançados a partir do estado atual. Portanto utilizando o *Modelo Cinemático* foi desenvolvido uma modelagem capaz de discretizar as transições dos estados que *Veículo Automatizado* pode assumir.

Como visto no desenvolvimento do *Modelo Cinemático*, o *Veículo Automatizado* realiza curvas através de arcos de círculos com um cento de curvatura determinado pelo ângulo das rodas, a Figura 6.8(a) mostra alguns desses arcos de círculos.

Para a discretização das transições dos estados que o *Veículo Automatizado* pode assumir foi desenvolvido uma estrutura construída através do *Modelo Cinemático* que é chamada de *Célula* e é mostrada na Figura 6.8(b).

A *Célula* tem como limite um círculo com raio  $R$  e centro igual a posição do *Veículo Automatizado*. É determinado um número  $N$  de variações no ângulo da roda do *Veículo Automa-*

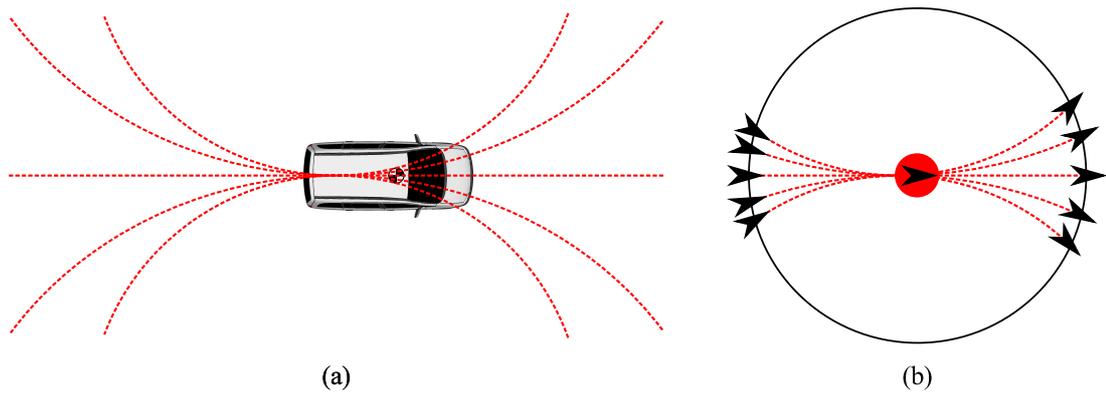


Figura 6.8: Raio de curvatura e *Célula*

tizado e para cada ângulo é determinado a posição com que o *Veículo Automatizado* cruzaria o limite da *Célula*.

Na Figura 6.8(b) a seta no centro da *Célula* representa a posição da *Célula*, as setas da direita e da esquerda representam respectivamente as posições que o *Veículo Automatizado* pode alcançar se estiver posicionado exatamente no centro da *Célula* quando se movimenta para frente e quando se movimenta para trás.

### 6.3.3 Operador $\Gamma$

Na Figura 6.9 é mostrado um exemplo de uma *Célula* com 6 divisões. A *Célula* do centro da figura representa o estado inicial e as outras *Células* representam os estados alcançáveis a partir do estado inicial.

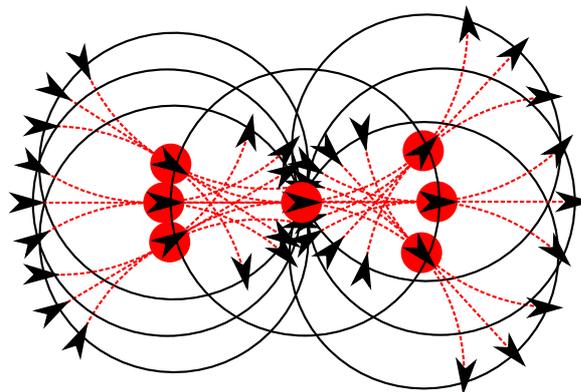


Figura 6.9: *Células*

O cálculo de construção das *Células* utilizando o *Modelo Cinemático* é realizado apenas uma vez, cada *Célula* representando algum dos estados é a *Célula* calculada posicionada de forma a representar o estado desejado.

O número  $N$  de divisões que cada *Célula* pode ter, equivale ao número de estados que o *Veículo Automatizado* pode alcançar a partir do estado representado pela *Célula*. Uma *Célula* é posicionada para cada novo estado alcançável, portando cada estado representado por uma *Célula* com  $N$  divisões resulta em  $N$  novos estados também representados pela *Célula*.

O operador  $\Gamma_{\vec{P}_i}$  determina cada um dos estados alcançáveis a partir do estado  $\vec{P}_i$ , resultando em um conjunto de novos estados e cada um representado por uma *Célula*.

Utilizando essa modelagem para cada estado alcançado existirão novos  $N$  estados, onde  $N$  é o número de divisões da *Célula* utilizada. Portanto seja  $E_{(k)}$  o número de estados já alcançados e  $E_{(k+1)}$  o número de estados alcançáveis a partir de  $E_{(k)}$ ,  $E_{(k+1)} = E_{(k)} + N$ .

### 6.3.4 Operador $\Omega$

O operador  $\Omega$  é responsável por indicar ao algoritmo de busca quando um nó pertence ao conjunto de nós objetivo.

Nessa busca local desenvolvida o estado  $P_i$  será considerado um dos estados objetivo quando a partir dele, com um único arco de curvatura possível de ser realizado pelo *Veículo Automatizado*, for possível alcançar o estado objetivo.

Dessa forma o operador  $\Omega_{\vec{P}_i}$  determina se com uma única curva realizada pelo *Veículo Automatizado* a partir do estado  $\vec{P}_i$  o estado objetivo pode ser alcançado.

### 6.3.5 Operador $\Lambda$

No centro da Figura 6.9 é possível observar que existe um grande número de possíveis estados alcançáveis que tem posições semelhantes mas não necessariamente iguais. Para uma redução no número de estados, alguns estados que são semelhantes ao ponto de poderem ser considerados estados equivalentes são representados por um único estado.

Foi criado operador  $\Lambda_{\vec{P}_i}(\vec{P}_j)$  capaz de verificar se estado  $\vec{P}_i$  é equivalente ao estado  $\vec{P}_j$ , onde os estados são considerados vetores posição do tipo  $\vec{P} = [Px, Py, \psi]$ . Para o resultado do operador  $\Lambda_{\vec{P}_i}(\vec{P}_j)$  ser verdadeiro a seguinte condição precisa ser respeitada:

$$E_{max} \geq E_d + E_\psi \quad (6.5)$$

Onde  $E_d$  e  $E_\psi$  são definidos como:

$$E_d = K_d \sqrt{(Px_i - Px_j)^2 + (Py_i - Py_j)^2} \quad (6.6)$$

$$E_\psi = K_\psi |\psi_i - \psi_j| \quad (6.7)$$

As constantes  $K_d$  e  $K_\psi$  são para normalizar os valores resultantes respectivamente dos erros  $E_d$  e  $E_\psi$  que apresentam unidades diferentes.  $E_{max}$  é o erro máximo que um estado pode apresentar em relação a outro para serem considerados equivalentes, portanto quanto maior o valor estipulado para  $E_{max}$  maior será o relaxamento para nas condições para os estados serem considerados equivalentes.

### 6.3.6 Custo $\hat{g}$

O custo  $\hat{g}$  é o custo do caminho ótimo conhecido para do estado inicial atingir o estado  $E$ , é definido simplesmente pelo comprimento de cada um dos arcos que ligam todos os estados do caminho ótimo conhecido.

Dessa forma o custo  $\hat{g}(E)$  é o comprimento do caminho que liga o estado inicial ao estado  $E$ , o custo do estado atual é determinado somando o custo do estado anterior ao custo do comprimento do arco que ligou o estado anterior ao estado atual.

Seja o operador  $Arc(E)$  responsável por determinar o comprimento do arco que liga o estado  $E$  ao seu estado antecessor, dessa forma com  $E_{(k-1)}$  sendo o estado antecessor de  $E_{(k)}$  temos:

$$\hat{g}(E_{(k)}) = \hat{g}(E_{(k-1)}) + Arc(E_{(k)}) \quad (6.8)$$

O operador  $Arc(E)$  determina o valor do comprimento do arco com base nos arcos calculados com o *Modelo Cinemático* para a construção da *Célula*. O operador leva em conta a direção que *Veículo Automatizado* precisa tomar pra alcançar o estado  $E$ , dessa forma estados em que o *Veículo Automatizado* necessita andar em marcha ré para serem alcançados podem ter um valor maior resultando em uma punição para estes estados.

Quando existirem obstáculos no *Ambiente* cada estado é avaliado para verificar se existe colisão entre algum dos obstáculos e o *Veículo Automatizado* no estado avaliado, caso exista alguma colisão o custo  $\hat{g}$  para o estado é considerado infinito.

### 6.3.7 Custo $\hat{h}$

O custo  $\hat{h}$  pode ser considerado como um dos mais importantes fatores para a solução da árvore de busca como foi modelada para a navegação local, ele é responsável por avaliar o custo estimado de um determinado estado para o estado objetivo.

A maior dificuldade no cálculo do custo  $\hat{h}$  é determinar uma heurística capaz de avaliar tanto as distâncias  $P_x$  e  $P_y$  quanto a diferença de ângulo  $\psi$  entre o estado avaliado e o estado objetivo. A heurística desenvolvida para determinar o custo  $\hat{h}$  é mostrada na Figura 6.10.

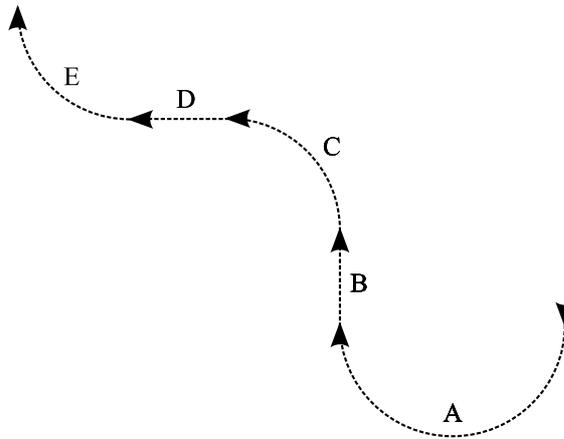


Figura 6.10: Custo  $\hat{h}$  para o Navegador Local

Utilizando os seguimentos de círculos  $A$ ,  $C$  e  $E$  e retas  $B$  e  $D$  mostrados na Figura 6.10 é possível a partir de qualquer estado alcançar qualquer outro estado. As retas  $B$  e  $D$  têm comprimentos variáveis com sinal indicando o sentido, os seguimentos de círculos  $A$ ,  $C$  e  $E$  são todos com raio igual ao raio de curvatura mínimo possível de ser realizado pelo *Veículo Automatizado* mas com ângulos variáveis.

A soma dos comprimentos dos seguimentos de círculos  $A$ ,  $C$  e  $E$  e das retas  $B$  e  $D$  que vai determinar o valor do custo  $\hat{h}$ , e os valores de cada um dos comprimentos são determinados por trigonometria.

O comprimento de  $A$  está relacionado com a diferença de ângulos  $\psi$  entre estado inicial e o estado objetivo.  $C$  e  $E$  têm comprimentos iguais e a soma dos dois é igual ao segmento de círculo com raio de curvatura mínimo formado com um ângulo de  $\pi$ . As retas  $B$  e  $D$  têm comprimentos relacionados com as distâncias entre os estados.

### 6.3.8 Busca

A busca é iniciada com o posicionamento da *Célula* na posição inicial para representar o estado inicial, e com o estado objetivo já é conhecido. O operador  $\Gamma$  é executado para o estado inicial gerando um conjunto de novos estados já com seus custos  $\hat{g}$  e  $\hat{h}$  calculados.

De acordo com o algoritmo de busca utilizado para resolver a árvore de busca novos estados serão selecionados entre os estados possíveis, gerando novos estados até alcançar o caminho ótimo do estado inicial ao estado objetivo.

A Figura 6.11 mostra uma busca incompleta, onde a partir da *Célula* representando o estado inicial, novos quatro estados foram selecionados.

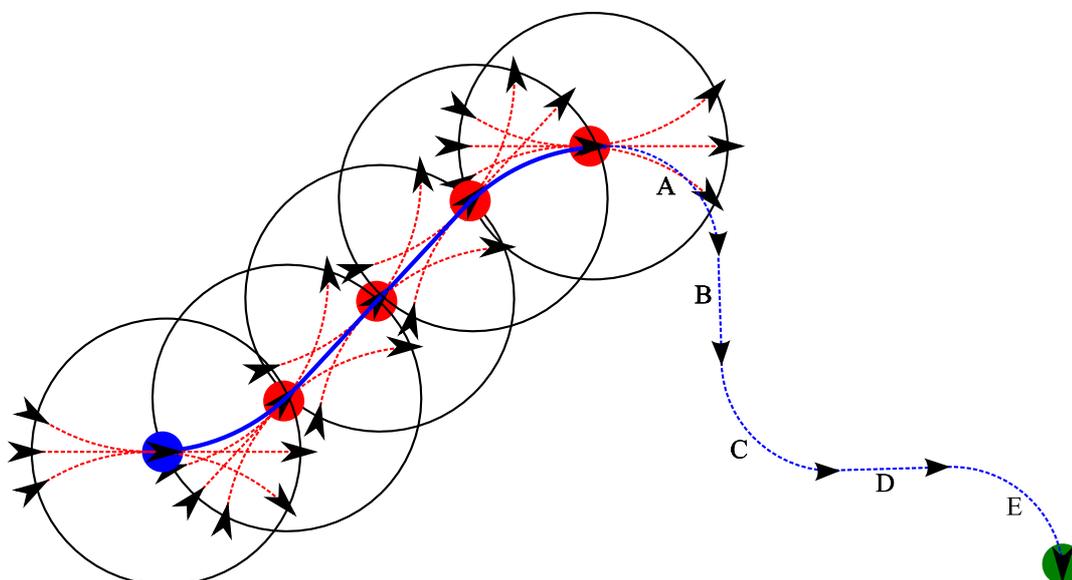


Figura 6.11: Busca de rota sendo executada pelo *Navegador Local*

Na Figura 6.11 a linha azul contínua ligando o estado inicial ao último estado selecionado representa o custo  $\hat{g}$  para o último estado selecionado e a linha azul pontilhada ligando o último estado selecionado ao estado objetivo representa o custo  $\hat{h}$  para o último estado selecionado.

A Figura 6.12 mostra a busca finalizada para os mesmos pontos inicial e objetivo da busca incompleta mostrada na Figura 6.11.

Como mostrado na Figura 6.12, devido a metodologia desenvolvida o ultimo estado selecionado não é o estado objetivo, mas sim um estado que a partir dele é possível alcançar o estado objetivo utilizando um único segmento de círculo.

A busca apenas determina a sequência de estados que levam o *Veículo Automatizado* do estado inicial ao estado objetivo, como cada um dos estados foi determinado através do *Modelo*

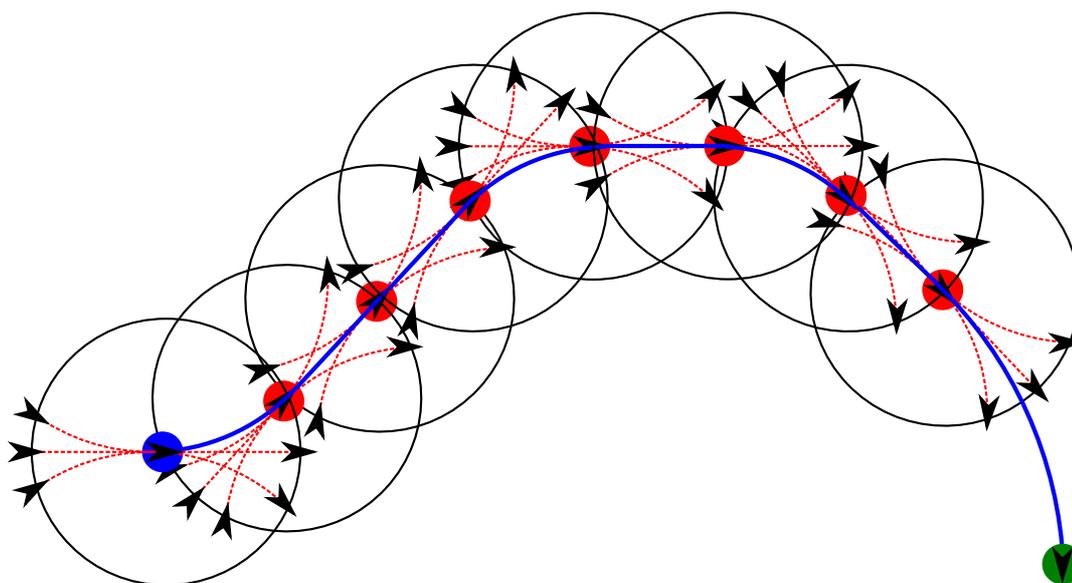


Figura 6.12: Busca de rota executada pelo *Navegador Local*

*Cinemático* todos os estados da sequência são alcançáveis a partir do estado anterior.

Com a sequência de estados determinada um *Controlador* pode ser desenvolvido para gerar os *Valores de Acionamento* necessários para o *Veículo Automatizado* seguir a sequência de estados e alcançar o estado objetivo.

### 6.3.9 Resultados

Para demonstrar a efetividade do *Navegador Local* na solução de problemas com características diversas alguns problemas foram modelados e tem suas soluções mostradas a seguir.

O primeiro exemplo demonstra a solução para o exemplo de busca mostrado na Figura 6.12, foi modelado um problema com estados inicial e objetivo parecidos com os do exemplo.

A Figura 6.14 mostra a solução encontrada para o estado inicial de  $[0, 0, 0]$  e o estado objetivo de  $[0, 20, -90]$ , com a *Célula* construída com 10 divisões e raio de  $3m$ . Os segmentos em azul representam os estados necessários para a partir do estado inicial alcançar o estado objetivo com setas indicando a sequência dos estados.

Foi modelado um problema em que para a solução ótima o *Veículo Automatizado* precisa realizar manobras tando no sentido positivo de seu deslocamento longitudinal quanto em sentido negativo. Nesse exemplo as posições inicial e objetivo são iguais mas com ângulos  $\psi$  apresentando uma diferença de  $180^\circ$ .

A Figura 6.14 mostra a solução encontrada para o estado inicial de  $[0, 0, 0]$  e o estado obje-

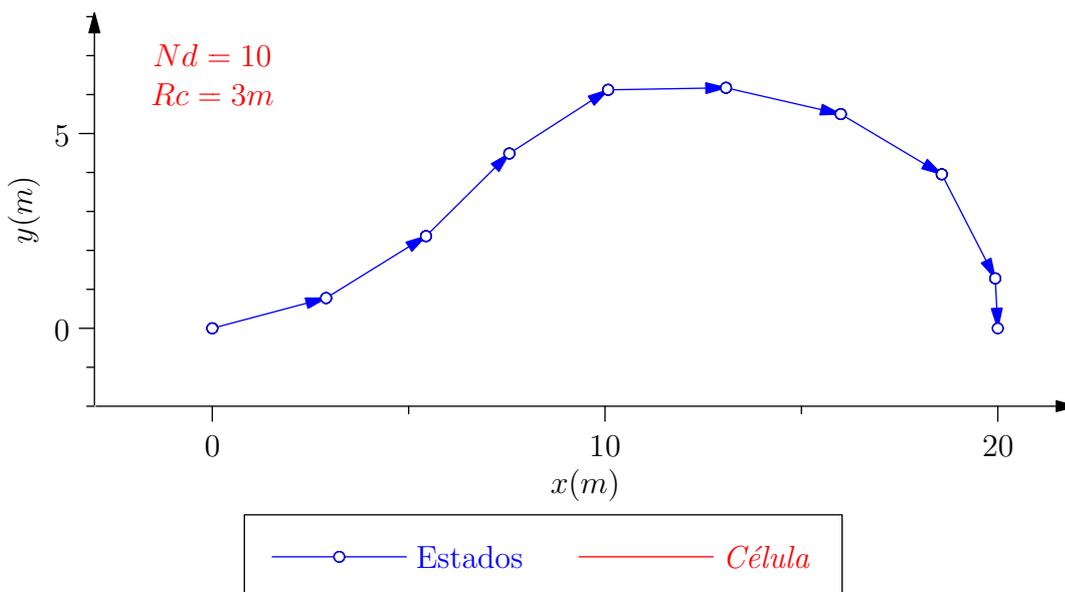


Figura 6.13: Resultado do primeiro exemplo do *Navegador Local*

tivo de  $[0, 0, 180]$ , com a *Célula* construída com 10 divisões e raio de  $3m$ .

Mesmo com uma penalização para manobras de ré realizadas pelo *Veículo Automatizado*, em muitos casos a rota de menor custo envolve manobras de mudança de sentido como no exemplo demonstrado na Figura 6.14. O estado inicial do veículo é  $[0, 0, 0]$  e estado objetivo  $[0, 0, 180]$ .

O próximo exemplo demonstra a solução encontrada quando o problema é modelado de forma que o *Veículo Automatizado* tem sua movimentação limitada por *Obstáculos* posicionados no *Ambiente*, simulando o tipo de situação encontrada para o posicionamento em vagas de garagens.

A Figura 6.15 mostra a solução encontrada para o estado inicial de  $[0, 0, 0]$  e o estado objetivo de  $[35, 15, 180]$ , com a *Célula* construída com 10 divisões e raio de  $5m$ .

Quando existem *Obstáculos* no *Ambiente* o número de rotas válidas do estado inicial ao estado objetivo se reduz muito, resultando em configurações de estados inicial e objetivo onde não existe como solução um rota válida. Nesses casos é necessário variar os parâmetros de construção da *Célula* e o relaxamento do operador  $\Lambda$ .

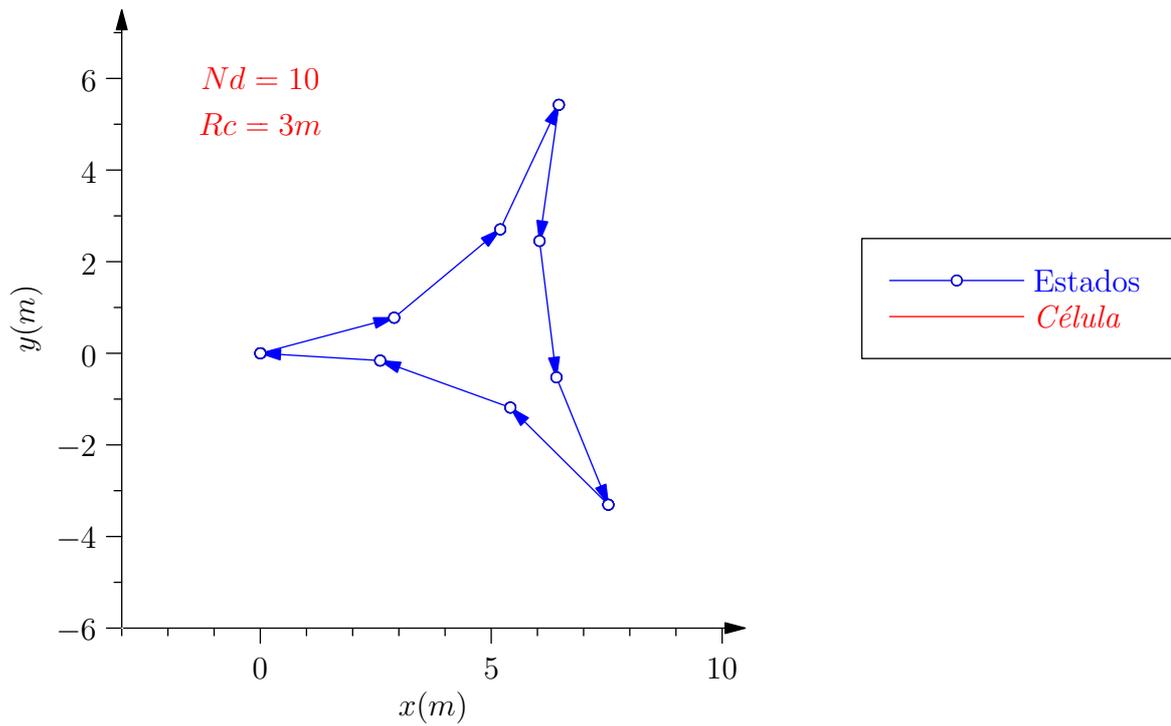


Figura 6.14: Resultado do segundo exemplo do *Navegador Local*

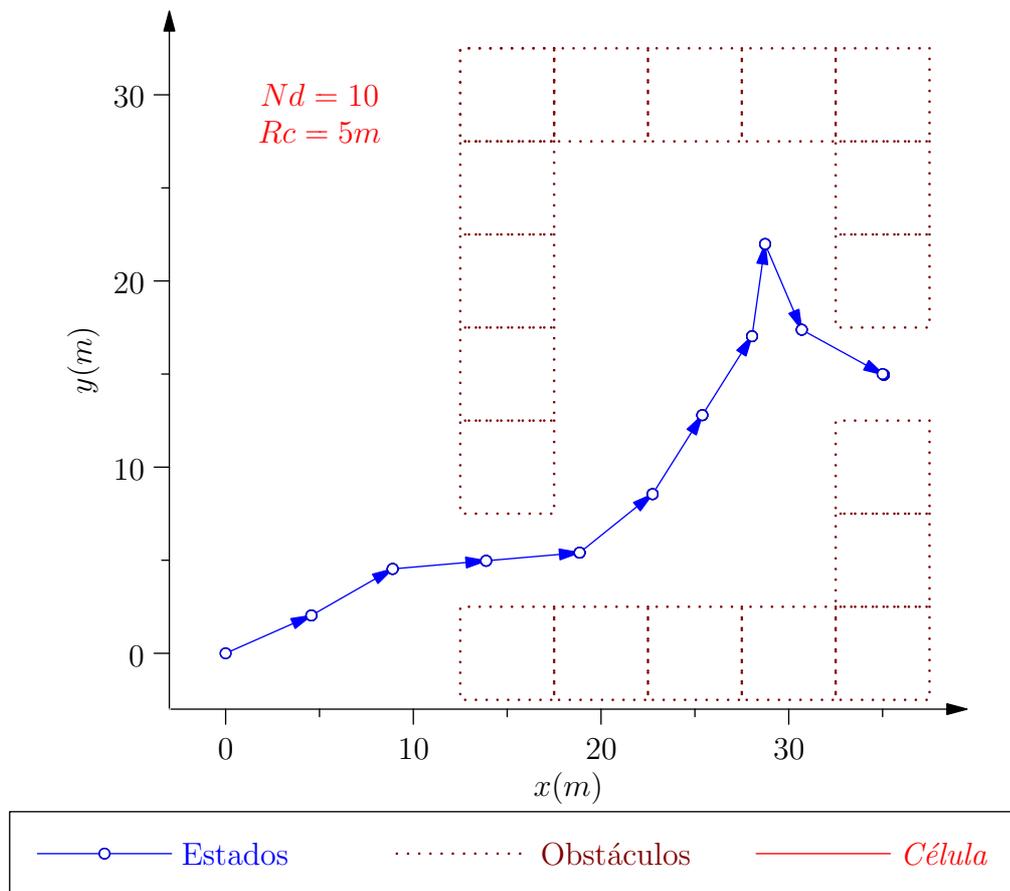


Figura 6.15: Resultado do terceiro exemplo do *Navegador Local*

## 6.4 Seguidor de Estados

### 6.4.1 Introdução

O *Controlador Seguidor de Estados* pertence a *Camada Executiva*, portanto é responsável por gerar *Valores de Atuação* para fazer com que o *Veículo Automatizado* partindo de um estado inicial alcance algum estado objetivo.

É o *Controlador* responsável por tratar a sequência de estados gerada pelo *Controlador Navegador Local* gerando *Valores de Atuação* capazes de fazer com que o *Veículo Automatizado* partindo do primeiro estado da sequência de estado alcance o último estado.

Os *Valores de Atuação* são gerados através da sequência de estados e da variação do estado do *Veículo Automatizado* enquanto o *Seguidor de Estados* está sendo executado. A variação do estado funciona como informação de realimentação que permite com que o *Controlador* evite erros acumulativos no processo.

Com a utilização dos dois *Controladores Navegador Local* e *Seguidor de Estados* é possível realizar um processo completo de navegação local em um ambiente conhecido. Com um estado inicial e um estado objetivo é determinada sequência de estados e os *Valores de Acionamento* necessários para fazer com que o *Veículo Automatizado* alcance o estado objetivo.

### 6.4.2 Metodologia

Para o *Veículo Automatizado* passar de um estado para outro é feita uma busca utilizando o *Modelo Cinemático* para determinar qual é o ângulo das rodas dianteiras que a partir do estado atual leve o *Veículo Automatizado* o mais próximo do estado seguinte.

Foi desenvolvido um operador chamado de  $Ar(E_i, E_o)$  capaz que dado um estado inicial  $E_i$  e um estado objetivo  $E_o$ , é capaz de determinar o ângulo das rodas dianteiras que faz com que o *Veículo Automatizado* partindo do estado inicial mais próximo se aproxime do estado objetivo com apenas uma curva.

Com o ângulo das rodas necessário para alcançar o estado desejado para o *Veículo Automatizado* determinado, o valor o ângulo é enviado como *Valor de Acionamento* ao *Veículo Automatizado*, quando o *Atuador* alcançar o valor é gerado um *Valor de Acionamento* responsável por colocar o *Veículo Automatizado* em movimento, esse movimento é realizado enquanto o *Veículo Automatizado* está se aproximando do estado desejado.

A cada novo estado da sequência alcançado, o estado atual utilizado para alcançar o próxi-

mo não é o estado da sequencia mas sim o estado atual do *Veículo Automatizado*. Essa metodologia evita que ocorra erros acumulativos, já que a cada estado alcançado pra alcançar o próximo é refeito o operador  $Ar(E_i, E_o)$  com o estado  $E_i$  determinado pelos *Valores dos Sensores*.

### 6.4.3 Resultados

Foram utilizados as três sequências de estados geradas como exemplo no *Controlador Navegador Local* como *Problemas* para serem solucionados pelo *Seguidor de Estados*.

Todos os gráficos apresentam os resultados numéricos encontrados quando o *Seguidor de Estados* foi implementado sobre o *Simulador*. O *Seguidor de Estados* não foi implementado sobre o *Veículo Automatizado* por problemas de acionamento e de precisão nas medidas de posicionamento, que até a conclusão deste trabalho não haviam sido solucionados.

O resultado do *Seguidor de Estados* para o primeiro exemplo desenvolvido no *Navegador Local* é mostrado na Figura 6.16.

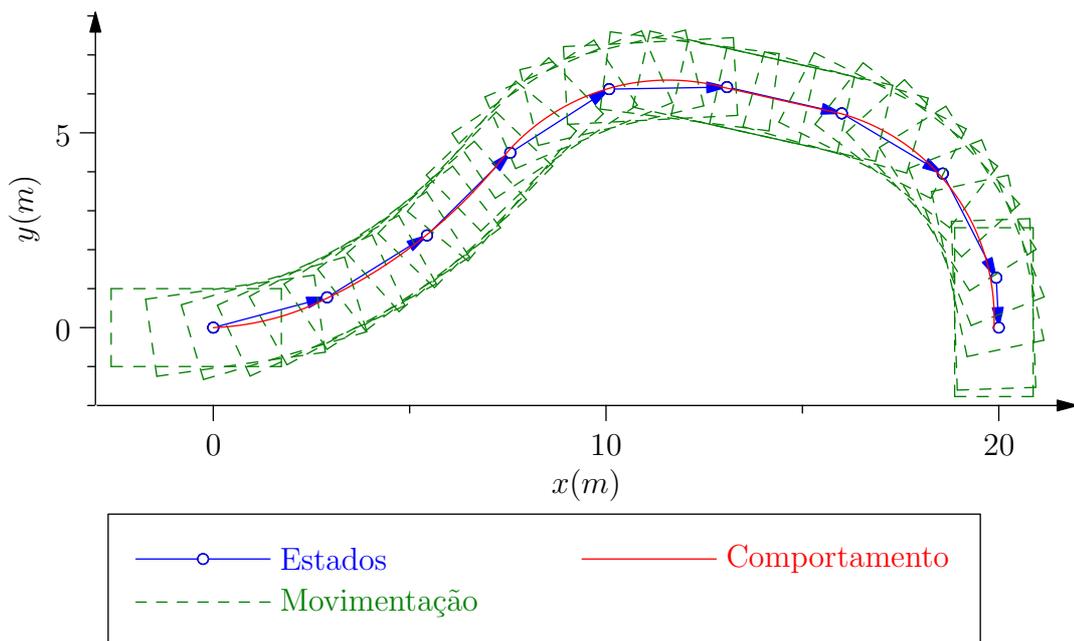


Figura 6.16: Resultado do primeiro exemplo do *Seguidor de Estados*

Os segmentos em azul representam os estados necessários para a partir do estado inicial alcançar o estado objetivo, a linha vermelha representa o comportamento resultante do *Simulador* para o *Veículo* passar por todos os estados encontrados pelo algoritmo de busca e os retângulos em verde pontilhado representam alguns estados da movimentação resultante de comportamento do *Veículo*.

A Tabela 6.1 mostra os resultados numéricos da busca para a solução mostrada na Figura

6.16.

Resultados			
$\vec{P}$	Posição Inicial	Posição Objetivo	Posição Resultante
$P_x(m)$	0	20	19,87541
$P_y(m)$	0	0	-0,03646159
$\psi(^{\circ})$	0	-90	-90,01169

Tabela 6.1: Tabela de Resultados do primeiro exemplo do *Seguidor de Estados*

A diferença existente entre o comportamento simulado e os estados encontrados na busca é devido ao relaxamento das restrições para o operador  $\Lambda$  considerar estados próximos equivalentes. Quanto maior for o relaxamento menor será o número de estados considerados na busca, mas maior será a diferença máxima encontrada entre os estados resultantes da busca e o estado resultante com comportamento do *Veículo Automatizado*.

Mesmo existindo essa diferença entre os estados da sequência de estados e os estados alcançados no *Simulador*, é possível perceber que a diferença não foi se acumulando de um estado para outro. O erro resultante entre os estados é unicamente determinado pelo relaxamento de restrições do operador  $\Lambda$  e por erros de acionamento e leitura dos sensores, portanto é um sistema sem erro acumulativo.

O resultado do *Seguidor de Estados* para o segundo exemplo desenvolvido no *Navegador Local* é mostrado na Figura 6.17.

A Tabela 6.2 mostra os resultado numéricos da busca para a solução mostrada na Figura 6.17.

Resultados			
$\vec{P}$	Posição Inicial	Posição Objetivo	Posição Resultante
$P_x(m)$	0	0	-0,01248567
$P_y(m)$	0	0	0,02777386
$\psi(^{\circ})$	0	180	179,8206

Tabela 6.2: Tabela de Resultados do segundo exemplo do *Seguidor de Estados*

O resultado do *Seguidor de Estados* para o terceiro exemplo desenvolvido no *Navegador Local* é mostrado na Figura 6.16.

A Tabela 6.3 mostra os resultados numéricos da busca para a solução mostrada na Figura 6.18.

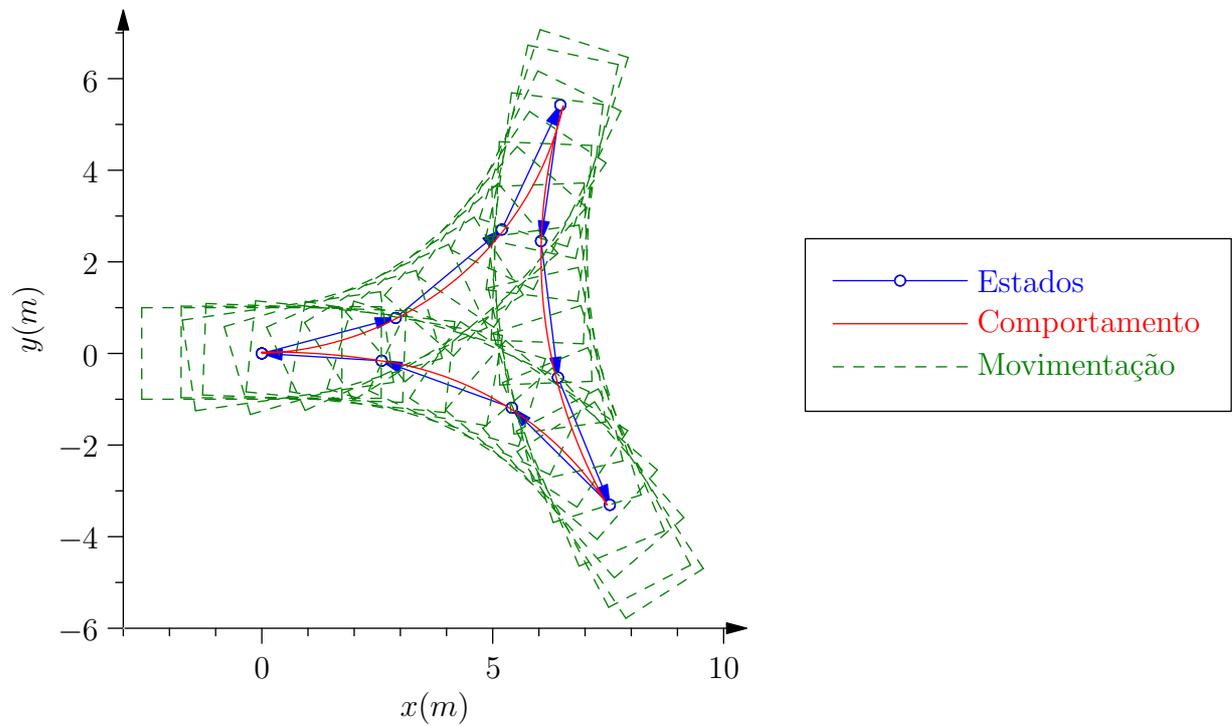
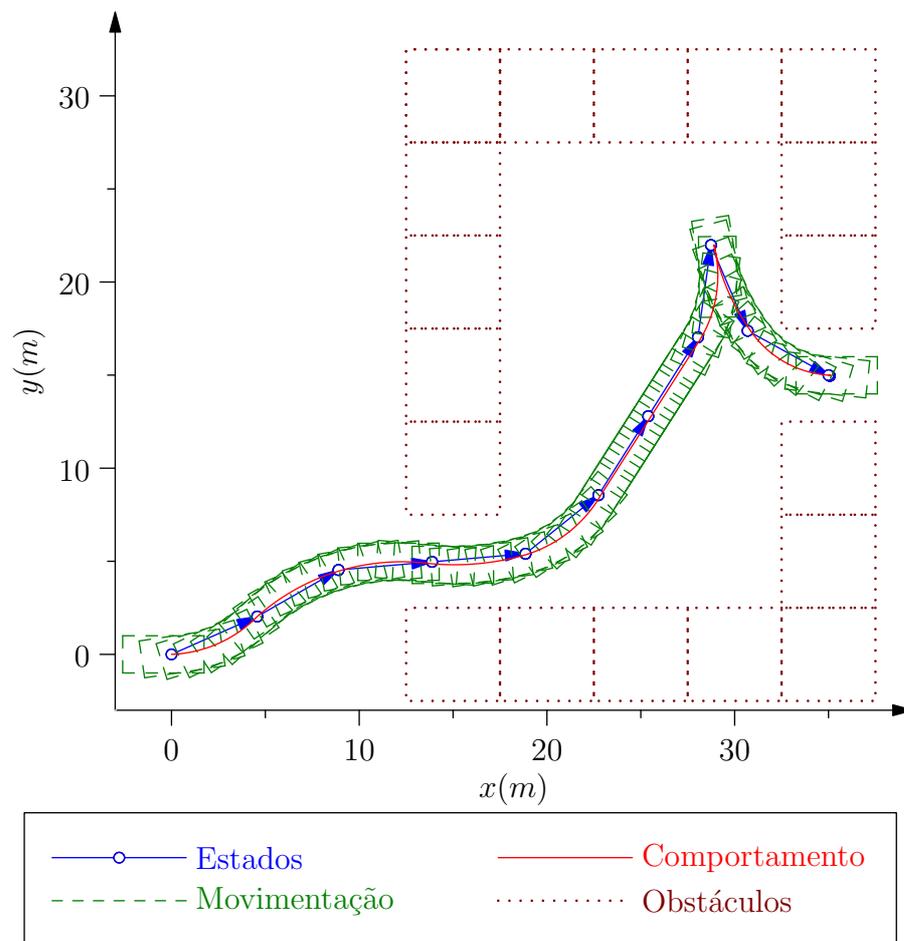


Figura 6.17: Resultado do segundo exemplo do *Seguidor de Estados*

Resultados			
$\vec{P}$	Posição Inicial	Posição Objetivo	Posição Resultante
$P_x(m)$	0	35	35,00357
$P_y(m)$	0	15	14,99126
$\psi(^{\circ})$	0	180	179,9895

Tabela 6.3: Tabela de Resultados do terceiro exemplo do *Seguidor de Estados*

Figura 6.18: Resultado do terceiro exemplo do *Seguidor de Estados*

# 7 *Conclusões e propostas para trabalhos futuros*

## 7.1 Introdução

Pelos resultados apresentados neste trabalho o *Sistema de Navegação* desenvolvido para o *Veículo Autônomo Inteligente* cumpre de forma geral os objetivos propostos.

Como demonstrado nos resultados, a arquitetura desenvolvida para o *Sistema de Navegação* permite através dos *Recursos Compartilhados* o desenvolvimento de *Controladores* capazes de fazer com que o *Veículo Automatizado* alcance algum objetivo pré determinado.

Não foram encontrados trabalhos semelhantes a este com resultados que permitissem uma comparação de desempenho, portanto nenhum dos componentes desenvolvidos neste trabalho tiveram seus resultados comparados com resultados de outros trabalhos.

A comparação de resultados com outros trabalhos através de uma maior pesquisa ou enquadramento dos resultados obtidos neste trabalho com os resultados de outros trabalhos, fica como uma das propostas para trabalhos futuros.

Apesar dos resultados demonstrando que este trabalho cumpre todos os objetivos propostos, claramente alguns componentes desenvolvidos necessitam de alterações para que a arquitetura desenvolvida se torne mais funcional e flexível. Os resultados de cada um dos principais componentes desenvolvidos são discutidos a seguir.

## 7.2 Recursos Compartilhados

### 7.2.1 Introdução

A introdução dos *Recursos Compartilhados* nos *Sistema de Navegação* permitiu uma maior flexibilidade no desenvolvimento de *Controladores* através da possibilidade de utilização dos *Recursos Compartilhados* já existentes ou do desenvolvimento de novos.

Como cada um dos *Recursos Compartilhados* pode ser utilizado por mais de um *Controlador* existe um fator de reutilização de código, e cada alteração em algum dos *Recursos Compartilhados* é repassada para todos os *Controladores* que utilizam o *Recurso Compartilhado* alterado.

Devidas as grandes diferenças existentes entre cada um dos *Recursos Compartilhados* desenvolvidos existe uma clara falta de padronização, como proposta para trabalhos futuros está uma melhor definição de como devem ser modelados os *Recursos Compartilhados* permitindo uma maior facilidade de utilização e desenvolvimento de novos *Recursos Compartilhados*.

### 7.2.2 Modelo Cinemático

Pelos resultados obtidos o *Modelo Cinemático* permite uma boa estimativa do comportamento do *Veículo Automatizado*, e pode ser utilizado pelo *Sistema de Controle* como um dos *Recursos Compartilhados* para o desenvolvimento de *Controladores* que utilizam dessa informação para determinar o acionamento necessário para o *Veículo Automatizado* alcançar objetivos determinados.

Uma limitação ao *Modelo Cinemático* é sua consideração de que o terreno é completamente plano, uma das propostas para trabalhos futuros é uma generalização fazendo com que o *Modelo Cinemático* também se comporte de maneira coerente para variações de altura do terreno.

### 7.2.3 Planificação de Coordenadas

O *Recurso Compartilhado* de *Planificação de Coordenadas* foi um dos primeiros componentes desenvolvidos neste trabalho por ser responsável por determinar qualquer tipo de medida de distância entre pontos de posicionamento global.

O processo de planificação de coordenadas globais é um dos processos mais utilizados neste trabalho, utilizado nos *Controladores* tanto para cálculos de navegação quanto para a representação de posicionamento disponibilizada para a *Interface com Usuário*.

Devidas as suas características a *Planificação de Coordenadas* tem resultados com uma precisão maior que a do DGPS utilizado no *Veículo Automatizado* para determinar suas coordenadas globais, portanto é uma precisão mais do que suficiente para este trabalho.

### 7.2.4 Algoritmo de Busca

O *Algoritmo de Busca* apesar de utilizar um algoritmo de busca tradicional, foi desenvolvido de forma independente permitindo a troca por outros algoritmos de busca sem a necessidade de alteração nos *Controladores* contanto que os novos algoritmos de busca sigam as regras impostas.

Pelos resultados obtidos neste trabalho se comprovou que a modelagem da árvore de busca em muitos casos é mais importante que a heurística do algoritmo de busca, justificando a utilização do A\* que é comprovadamente um algoritmo de busca ótimo e que apresenta um desempenho suficiente para a solução da maioria dos problemas.

Como propostas para trabalhos futuros no *Algoritmo de Busca* está uma alteração na estrutura desenvolvida para permitir a utilização de outros algoritmos de busca em conjunto com o A\*. Esses novos algoritmos de busca seriam desenvolvidos seguindo o mesmo tipo de modelagem do A\*, dessa forma poderiam ser rodados em paralelo na solução de cada problema para permitir a utilização de uma solução gerada por qualquer um dos algoritmos.

## 7.3 Sistema de Controle

A arquitetura mista utilizada no desenvolvimento do *Sistema de Controle* se mostrou abrangente o suficiente para a solução dos problemas propostos neste trabalho, a *Camada Coordenadora* apesar de mais complexa se tornou necessária para todo o funcionamento do *Sistema de Controle* sendo uma concentradora dos recursos.

Para trabalhos futuros uma proposta é uma reestruturação na *Camada Coordenadora* para que não seja necessária uma grande alteração em sua estrutura para cada novo *Controlador* desenvolvido. Será necessário um desenvolvimento de uma metodologia em que cada novo *Controlador* terá que cumprir certos requisitos, permitindo que a *Camada Coordenadora* sofra apenas pequenas alterações para que o *Sistema de Controle* possa utilizar o novo *Controlador*.

## 7.4 Simulador

Pelos resultados obtidos neste trabalho resultantes da diferença de *Comportamento* entre o *Simulador* e o *Veículo Automatizado* nas mesmas condições, é possível afirmar que o *Simulador* pode ser utilizado para determinar diferenças de *Comportamento* do *Veículo Automatizado* resultantes de alterações em qualquer alteração no *Sistema de Navegação*.

O *Simulador* ainda permitiu a otimização de parâmetros do *Modelo Cinemático*, fazendo com que o *Comportamento* resultante fosse bem próximo do *Comportamento* do *Veículo Automatizado*. Resultando em um *Modelo Cinemático* otimizado para ser utilizado pelo *Sistema de Controle*.

Como proposta para trabalhos futuros está uma modelagem mais fiel para os sensores e atuadores que fazem parte do *Veículo Automatizado*

## 7.5 Controladores

### 7.5.1 Introdução

Cada um dos *Controladores* desenvolvidos apresentou resultados que comprovam que os objetivos dos *Controladores* foram alcançados. No caso dos *Controladores* que se complementam na solução de problemas de navegação local, os resultados mostram que a arquitetura desenvolvida permite a interação de mais de um *Controlador* para a solução de problemas mais complexos resultando em uma grande flexibilidade na solução de problemas.

### 7.5.2 Navegador Global

O *Controlador de Rotas Globais* cumpre todos os objetivos que foram determinados para seu desenvolvimento, para toda área mapeada os resultados são sempre ótimos e devido a simplicidade da árvore de busca gerada tem um tempo de processamento muito baixo.

A maior dificuldade no *Controlador de Busca Global* é a necessidade de uma montagem manual da árvore de busca, onde cada *Waypoint* precisa ser selecionado individualmente e organizado em *Sub Circuitos*.

Como proposta para trabalhos futuros está o desenvolvimento de um sistema capaz de dado todos os *Waypoints*, seja capaz de organiza-los de forma automática ou pelo menos eliminar os redundantes.

### 7.5.3 Navegador Local

O *Navegador Local* mesmo sendo um *Controlador* complexo cumpre os objetivos estabelecidos, é capaz de encontrar uma lista de estados que permite ao *Veículo Automatizado* passar de seu estado inicial para praticamente qualquer estado objetivo.

A heurística desenvolvida para determinar o custo  $\hat{h}$  é fundamental para um bom funcionamento do *Controlador de Rotas Locais* e desempenha muito bem seu papel quando não existem obstáculos entre o estado inicial e o estado objetivo, mas seu desempenho quando existem obstáculos é bem pior resultando em um tempo de processamento muito alto.

Como proposta para trabalhos futuros está o desenvolvimento de uma heurística mais geral capaz de ter bom desempenho nas situação com obstáculos e sem obstáculos e uma configuração automática dos parâmetros para a construção da *Célula*, resultando um aumento de flexibilidade e confiabilidade para o *Controlador de Rotas Locais*.

#### 7.5.4 Seguidor de Estados

O *Controlador Seguidor de Estados* a partir da sequência de estados gerado pelo *Controlador Navegador Local*, foi capaz de gerar os *Valores de Acionamento* necessários para fazer o *Veículo Simulado* passar por todos os estados da sequência de estados e alcançar o estado objetivo.

Mesmo com os resultados apresentados cumprindo os objetivos a principal falha no desenvolvimento do *Seguidor de Estados* foi a impossibilidade de ser implementado sobre o *Veículo Automatizado*. Com a implementação apenas sobre o *Simulador* muitos dos possíveis problemas gerados por erros de atuação e de leitura dos sensores não são percebidos, tornando os resultados do *Controlador* menos confiáveis.

Como proposta para trabalhos futuros está a implementação do *Controlador Seguidor de Estados* sobre o *Veículo Automatizado*, para permitir uma melhor análise de seu desempenho em situações não previstas que não ocorrem quando o *Controlador* está implementado no *Simulador*.

## *Referências Bibliográficas*

- [Abe 2009]ABE, M. *Vehicle Handling Dynamics: Theory and Application*. [S.l.]: Elsevier Ltd, 2009. 29, 32
- [Choset et al. 2005]CHOSET, H. et al. *Principles of Robot Motion-Theory, Algorithms, and Implementation*. [S.l.]: The MIT Press, 2005. 36
- [DARPA 2007]DARPA. *Defense Advanced Research Projects Agency: Urban Challenge, Rules*. [S.l.], 2007. 18
- [Geddes 1939]GEDDES, N. B. *Magic Motorways*. [S.l.]: Random House, 1939. 16
- [Grewal, Weill e Andrews 2007]GREWAL, M. S.; WEILL, L. R.; ANDREWS, A. P. *Global Positioning Systems, Inertial Navigation and Integration*. 2. ed. [S.l.]: John Wiley & Sons, Inc, 2007. 34
- [Hart, Nilsson e Raphael 1968]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*, 1968. 37, 39, 40, 41
- [Kaplan e Hegarty 2006]KAPLAN, E. D.; HEGARTY, C. J. *Understanding GPS: Principles and Applications*. 2. ed. [S.l.]: Artech House, Inc, 2006. 34
- [Miller, Garcia e Campbell 2006]MILLER, I.; GARCIA, E.; CAMPBELL, M. To drive is human. *IEEE Computer Society*, 2006. 20
- [NIMA 2000]NIMA. *Department of Defense World Geodetic System 1984 : Its Definition and Relationships with Local Geodetic Systems*. [S.l.], 2000. 35
- [Rajamani 2006]RAJAMANI, R. *Vehicle Dynamics and Control*. [S.l.]: Springer, 2006. 17, 30, 31
- [Russell e Norvig 2003]RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 2. ed. [S.l.]: Pearson Education, Inc, 2003. 22, 25, 26, 33, 36, 37, 39, 42, 44, 46
- [Seetharaman, Lakhota e Blasch 2006]SEETHARAMAN, G.; LAKHOTIA, A.; BLASCH, E. P. Unmanned vehicles come of age: The darpa grand challenge. *IEEE Computer Society*, 2006. 18
- [Siegwart e Nourbakhsh 2004]SIEGWART, R.; NOURBAKHS, I. R. *Introduction to Autonomous Mobile Robots*. [S.l.]: The MIT Press, 2004. 28
- [Stoll 2007]STOLL, J. D. Could gm's salvation be stuff of science fiction? *The Wall Street Journal*, p. B1, jan. 2007. 18

- [Thomas D 1992]THOMAS D, G. *Fundamentals of Vehicle Dynamics*. [S.l.]: Society of Automotive Engineers, Inc., 1992. 27, 28
- [Thrun, Burgard e Fox 2005]THRUN, S.; BURGARD, W.; FOX, D. *Probabilistic Robotics*. [S.l.]: The MIT Press, 2005. 63
- [Vincenty 1975]VINCENTY, T. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey Review XXII*, 1975. 36
- [Vlacic 2001]VLACIC, L. *Intelligent Vehicle Technologies: Theory and Applications*. [S.l.]: Butterworth-Heinemann, 2001. 17, 20
- [Weise 2009]WEISE, T. *Global Optimization Algorithms*. 2. ed. [S.l.]: <http://www.it-weise.de/>, 2009. 50, 52, 53
- [Williams 2009]WILLIAMS, D. Audi's driverless car goes racing. *Telegraph.co.uk*, dez. 2009. 18