



Universidade Federal da Bahia
Escola Politécnica
Departamento de Engenharia Elétrica
Programa de Pós-Graduação em Engenharia Elétrica



DISSERTAÇÃO DE MESTRADO

NAVEGAÇÃO DE VEÍCULOS DIFERENCIAIS BASEADA EM CAMPOS POTENCIAIS ARTIFICIAIS

TADEU ABREU CERQUEIRA

Salvador
2016



Universidade Federal da Bahia
Escola Politécnica
Departamento de Engenharia Elétrica
Programa de Pós-Graduação em Engenharia Elétrica



NAVEGAÇÃO DE VEÍCULOS DIFERENCIAIS BASEADA EM CAMPOS POTENCIAIS ARTIFICIAIS

TADEU ABREU CERQUEIRA

Orientador: Prof. Dr. Tito Luís Maia Santos
Co-orientador: Prof. Dr. André Gustavo Scolari Conceição

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal da Bahia, como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.

Banca Examinadora

Prof. Dr. Tito Luís Maia Santos - UFBA
Prof. Dr. André Gustavo Scolari Conceição - UFBA
Prof. Dr. Eduardo Telmo Fonseca Santos - IFBA
Prof. Dr. Bernardo Ordoñez - UFBA
Prof. Dr. Humberto Xavier de Araújo - UFBA

Salvador
2016

Abreu Cerqueira, Tadeu
Navegação de Veículos Diferenciais Baseada em Campos
Potenciais Artificiais / Tadeu Abreu Cerqueira. --
Salvador, 2016.
91 f. : il

Orientador: Tito Luís Maia Santos.
Coorientador: André Gustavo Scolari Conceição.
Dissertação (Mestrado - Programa de Pós-Graduação em
Engenharia Elétrica) -- Universidade Federal da Bahia,
Escola Politécnica, 2016.

1. Robótica Móvel. 2. Planejamento de Trajetórias. 3.
Campos Potenciais Artificiais. 4. Desvio de Obstáculos
com Veículos Diferenciais. 5. Controle Cinemático. I.
Maia Santos, Tito Luís. II. Scolari Conceição, André
Gustavo. III. Título.

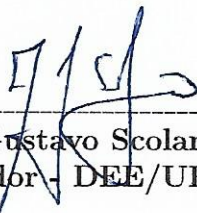
Navegação De Veículos Diferenciais Baseada Em Campos Potenciais Artificiais

Tadeu Abreu Cerqueira

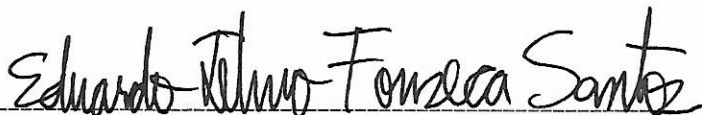
Dissertação de Mestrado aprovada em 13 de dezembro de 2016 pela banca examinadora composta pelos seguintes membros:



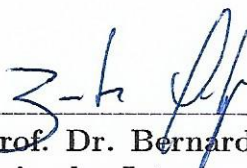
Prof. Dr. Tito Luís Maia Santos
Orientador - DEE/UFBA



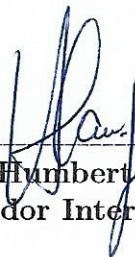
Prof. Dr. André Gustavo Scolari Conceição
Co-orientador - DEE/UFBA



Prof. Dr. Eduardo Telmo Fonseca Santos
Examinador Externo - IFBA



Prof. Dr. Bernardo Ordoñez
Examinador Interno - DEE/UFBA



Prof. Dr. Humberto Xavier de Araújo
Examinador Interno - DEE/UFBA

*A Deus.
À minha Mãe.
À minha Esposa.
Aos familiares e amigos que sempre me apoiam.*

“Tudo posso naquele que me fortalece”
(Filipenses. 4.13)

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema de navegação, aplicado para veículos que possuem locomoção diferencial, através do planejamento de trajetórias e controle cinemático do veículo, visando o desvio de obstáculos previamente detectados. Para o planejamento, é utilizada uma abordagem modificada do método baseado em campos potenciais artificiais, conhecido também como método APF (*Artificial Potential Fields*). Essa nova abordagem consiste em uma forma de superar o problema dos mínimos locais, inerente ao método APF, de modo a planejar trajetórias que evitem os obstáculos e que atinjam o ponto final desejado, caso haja alguma trajetória factível. Referente ao controle cinemático, são estudadas estratégias através de controladores NTI (*Nonlinear Time-Invariant control*) e DFL (*Dynamic Feedback Linearization*), de forma a manter o veículo dentro de um caminho previamente planejado, visando atingir o ponto final desejado e desviar dos obstáculos detectados dispostos no ambiente. Usa-se o robô diferencial Lego EV3 como base de testes para aplicação do sistema de navegação desenvolvido, em que são realizadas simulações e experimentos em laboratório.

Palavras-chave: Robótica Móvel, Planejamento de Trajetórias, Campos Potenciais Artificiais, Desvio de Obstáculos, Veículos Diferenciais, Controle Cinemático.

ABSTRACT

This work presents a development of a navigation system, applied to mobile differential vehicles, through a path planning and kinematic control system with obstacles avoidance of a previously detected environment. It is used a modified approach for a method based on artificial potential fields for path planning, also known as APF (Artificial Potential Fields) method. This new approach is a way to overcome the problem of local minima presents on the APF method and plan paths to avoid the obstacles and reach the desired end point, if there is any feasible path. For the kinematic control, it is studied strategies with NTI (Nonlinear Time-Invariant Control) and DFL (Dynamic Feedback Linearization) controlers to keep the vehicle in a previously planned path in order to reach the desired end point and avoid the obstacles arranged on the environment. It is used a differential robot Lego EV3 as a test basis for application of the developed navigation system, with simulations and experimental results.

Keywords: Mobile Robotics, Path Planning, Artificial Potential Fields, Obstacles Avoidance, Differential Vehicles, Kinematic Control.

Lista de Figuras

1.1	Representação gráfica do mapa de rotas para planejamento de trajetórias	2
1.2	Decomposição do ambiente por células	3
1.3	Escolha da sequência de células para atingir o ponto final desejado . .	4
1.4	Trajectoria com o método baseado em Campos Potenciais Artificiais .	4
1.5	Planejamento e Controle da Trajetória para desvio de obstáculos . . .	6
1.6	Mínimo local devido a um obstáculo côncavo	7
2.1	Estacionamento de um robô móvel com restrições não-holonômicas . .	11
2.2	Geometria básica de um robô móvel com tração diferencial	12
2.3	O robô no sistema de coordenadas global.	13
2.4	Diagrama de blocos do sistema de controle	15
2.5	Trajectoria obtida através do método de campos potenciais	18
2.6	Menor distância para um obstáculo	20
2.7	Exemplo de geração de trajetória usando interpolação cúbica	21
2.8	Ocorrência do Problema dos Mínimos Locais	22
3.1	Robô móvel com tração diferencial montado com o kit Lego EV3 . . .	25
3.2	Resposta em malha aberta para o acionamento do robô	27
3.3	Comparação entre a dinâmica calculada com a obtida no ensaio em laboratório	28
3.4	Diagrama de blocos detalhado para o sistema de controle	29
3.5	Algoritmo de controle simplificado	31
3.6	Resultado para análise da trajetória percorrida com controlador DFL	33
3.7	Resultados para análise do esforço de controle no percurso da trajetória com controlador DFL	33
3.8	Resultado para análise da trajetória percorrida com controlador NTI	34
3.9	Resultados para análise do esforço de controle no percurso da trajetória com controlador NTI	35
3.10	Algoritmo geral para o planejamento de trajetória e controle cinemático	36
3.11	Algoritmo para o planejamento de trajetória	39
3.12	Esquema gráfico do planejamento APF-2 em cada iteração	40

3.13 Fluxograma simplificado do método APF-2 de planejamento de trajetória	42
4.1 Ambiente 1	46
4.2 Ambiente 2	47
4.3 Ambiente 3	47
4.4 Ambiente 4	48
4.5 Simulação 1	50
4.6 Simulação 2	50
4.7 Simulação 3	51
4.8 Simulação 4	51
4.9 Simulação 5	52
4.10 Simulação 6	52
4.11 Simulação 7	53
4.12 Simulação 8	53
4.13 Simulação 9	54
4.14 Simulação 10	54
4.15 Simulação 11	55
4.16 Simulação 12	55
4.17 Simulação 13	56
4.18 Simulação 14	56
4.19 Simulação 15	57
4.20 Simulação 16	57
4.21 Sinais de Controle aplicados na <i>Simulação 2</i> , com o APF-2	58
4.22 Variação dos Parâmetros de Planejamento durante a <i>Simulação 2</i> , com o APF-2	58
4.23 Ambiente 1 Real	60
4.24 Resultado da trajetória percorrida no <i>Experimento 1</i>	62
4.25 Sinais de controle aplicados no <i>Experimento 1</i>	62
4.26 Resultado da trajetória percorrida no <i>Experimento 2</i>	63
4.27 Sinais de controle aplicados no <i>Experimento 2</i>	63
4.28 Resultado da trajetória percorrida no <i>Experimento 3</i>	64
4.29 Sinais de controle aplicados no <i>Experimento 3</i>	64
4.30 Resultado da trajetória percorrida no <i>Experimento 4</i>	65
4.31 Sinais de controle aplicados no <i>Experimento 4</i>	65

Lista de Tabelas

3.1	Rotação das rodas medidas em cada instante do ensaio	26
3.2	Parâmetros escolhidos para o controlador DFL	32
3.3	Parâmetros escolhidos para o controlador NTI	34
3.4	Escolha dos parâmetros de atração e repulsão do método APF	38
4.1	Simulações executadas para o sistema de navegação desenvolvido . . .	49
4.2	Experimentos realizados no <i>Ambiente 1 Real</i> com o robô Lego EV3 .	61

Índice

Resumo	ix
Abstract	x
Lista de Figuras	xi
Lista de Tabelas	xiii
Índice	xiv
1 INTRODUÇÃO	1
1.1 Objetivo	5
1.2 Motivação	6
1.3 Organização da Dissertação	8
2 FUNDAMENTAÇÃO TEÓRICA	9
2.1 Arquitetura de Navegação para Robôs Móveis	9
2.2 Robôs Móveis com Acionamento Diferencial	10
2.2.1 Modelo Cinemático	12
2.2.2 Modelo Dinâmico	14
2.3 Controle Cinemático	15
2.3.1 Linearização por Realimentação Dinâmica - DFL	16
2.3.2 Não-Linear Invariante no Tempo - NTI	17
2.4 Planejamento da Trajetória	18
2.4.1 Funções Potenciais	18
2.4.2 Adaptação da Trajetória com Interpolação Cúbica	20
2.4.3 O Problema dos Mínimos Locais	21
2.5 Conclusão	23
3 SISTEMA DE NAVEGAÇÃO	24
3.1 O Robô Diferencial Lego EV3	24
3.1.1 Ensaio de Acionamento	25
3.1.2 Modelo do Robô Lego	27
3.2 Implementação do Controle	29

3.2.1	O Algoritmo de Controle	30
3.2.2	Controlador DFL	31
3.2.3	Controlador NTI	34
3.3	Implementação do Planejamento	35
3.3.1	Algoritmo do Método APF	37
3.3.2	Modificação do Método APF	39
3.4	Conclusão	43
4	RESULTADOS	45
4.1	Simulações do Sistema	46
4.1.1	Resultados da Trajetória Planejada e Percorrida	49
4.1.2	Resultado do Esforço de Controle	57
4.1.3	Variação dos Parâmetros no Método APF-2	58
4.1.4	Análise dos Resultados	59
4.2	Resultados Experimentais	60
4.3	Conclusão	66
5	CONSIDERAÇÕES FINAIS	67
5.1	Perspectivas de Trabalhos Futuros	69
	Referências Bibliográficas	70
	Apêndice: Código em MATLAB para Controle do Robô Lego EV3 . .	73
	Apêndice: Algoritmo do Planejamento APF-2 em MATLAB	74

Capítulo 1

INTRODUÇÃO

Nas últimas décadas, tem ocorrido significativo desenvolvimento de robôs capazes de executar tarefas cada vez mais complexas, exigindo habilidades de se adaptar a situações inesperadas, como por exemplo, quando há um obstáculo no caminho previamente planejado [1].

Na robótica móvel, sabe-se que existem três etapas necessárias para que um robô execute determinada tarefa com eficácia [2]:

- *Mapeamento*;
- *Planejamento de Trajetória* e
- *Controle*.

Na etapa de *Mapeamento*, o problema consiste em explorar um ambiente desconhecido, através da coleta de dados pelos sensores do robô. Na robótica, os sensores responsáveis por coletar essas informações do meio são classificados como *externoceptivos*.

Existem diversos trabalhos publicados que exploram a etapa de mapeamento do ambiente, através do uso de técnicas para reconstrução do meio. Em [3], observa-se o uso de sensores com ultra-som e infravermelho para detecção de obstáculos. Já em [4], é observada uma forma de realizar a leitura de um ambiente tridimensional, através do uso de um *Kinect* e posterior processamento da imagem, de modo que sejam obtidas as coordenadas onde os obstáculos estão posicionados em relação ao robô. Em todos esses métodos de mapeamento constata-se amplo domínio das

técnicas já consagradas na literatura, sendo que para ambas as técnicas, o mais importante é obter as posições dos obstáculos do meio em relação ao robô, possibilitando que seja realizada a etapa de planejamento com base no objetivo desejado.

O *Planejamento de Trajetória* é a etapa que realiza o traçado do caminho que o robô deverá percorrer, a fim de cumprir os objetivos propostos, tais como atingir uma determinada posição final e desviar de obstáculos já detectados na etapa de mapeamento. A literatura indica o uso de três métodos básicos para desvio de obstáculos, baseados em [2]:

- *Mapa de Rotas;*
- *Decomposição por Células e*
- *Campos Potenciais Artificiais.*

No *Mapa de Rotas* são geradas linhas retas ligando os vértices de cada obstáculo identificado com os pontos inicial e final desejado, de forma que haverá um caminho poligonal conectando o ponto inicial com o final [5], como pode ser observado na Figura 1.1.

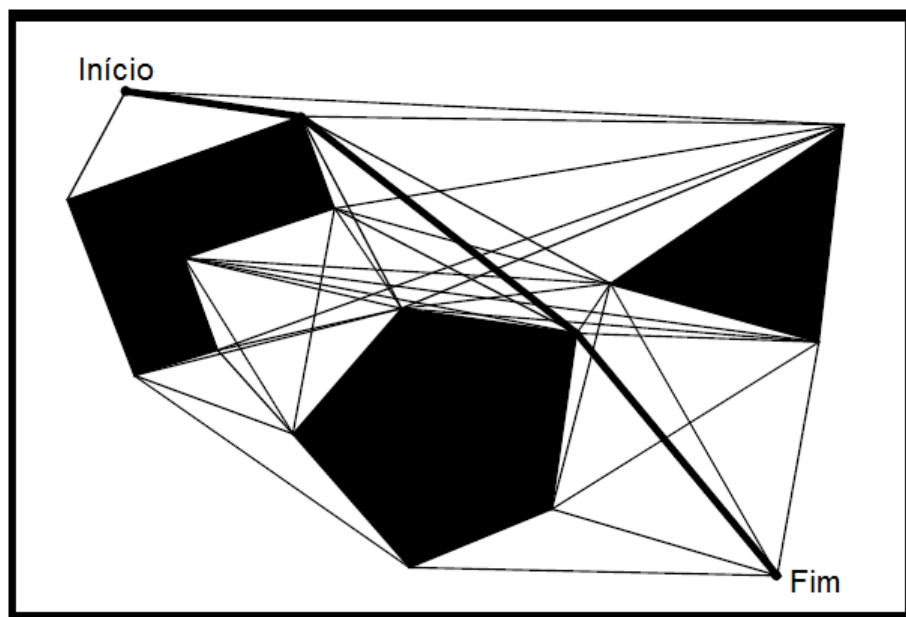


Figura 1.1: Representação gráfica do mapa de rotas para o planejamento de trajetórias.

No método baseado na *Decomposição por Células* determinam-se as regiões ocupadas pelos obstáculos do meio, bem como as livres nas quais o veículo pode trafegar sem colisões, gerando áreas denominadas de *células*, como pode ser observado na Figura 1.2.

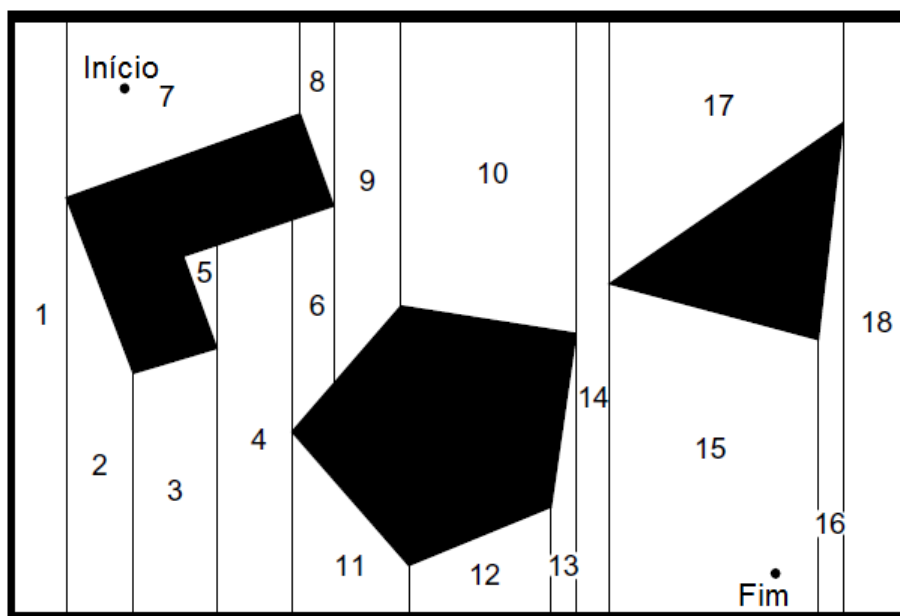


Figura 1.2: Decomposição do ambiente por células.

A partir dessa decomposição, pode-se usar um algoritmo que faça o veículo percorrer um caminho passando pelos pontos médios dessas células até chegar em uma onde encontra-se o ponto final desejado, gerando um caminho poligonal. Caso o ponto final não esteja no ponto médio da célula em que se encontra, basta percorrer uma linha reta do ponto médio de sua célula até o ponto desejado, já que dentro dessa região livre não há risco de colisões com obstáculos. Normalmente, usa-se um algoritmo que passe pelo menor número de células até chegar na célula do ponto final desejado [1]. Na Figura 1.3, tem-se, para o ambiente da Figura 1.2, a escolha da sequência com menor número de células que faz o veículo sair da posição inicial e chegar na célula do ponto final.

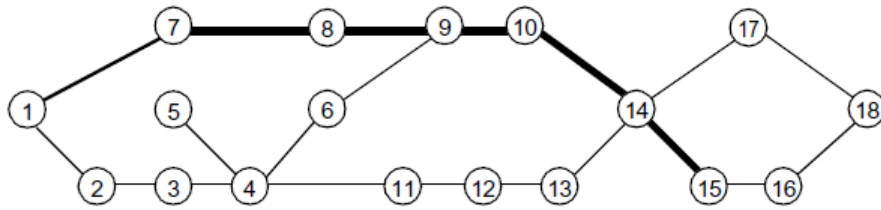


Figura 1.3: Escolha da sequência de células para atingir o ponto final desejado.

Já o método baseado em *Campos Potenciais Artificiais*, conhecido também como método APF (*Artificial Potential Fields*), visto em [6, 7, 8], utiliza-se de uma função potencial, definida matematicamente como um campo vetorial, para gerar uma trajetória que naturalmente faz o veículo atingir o ponto final desejado e desviar dos obstáculos. De acordo com [9], o APF é um método de simples implementação e baixo custo computacional, em que são planejadas trajetórias que evitam os obstáculos, através de caminhos seguros que impedem a ocorrência de colisões, ao mesmo tempo em que tende a convergir para o ponto final desejado. Na Figura 1.4 pode-se observar um exemplo de trajetória gerada pelo método APF, tomando como base o mesmo ambiente usado nas Figuras 1.1 e 1.2.

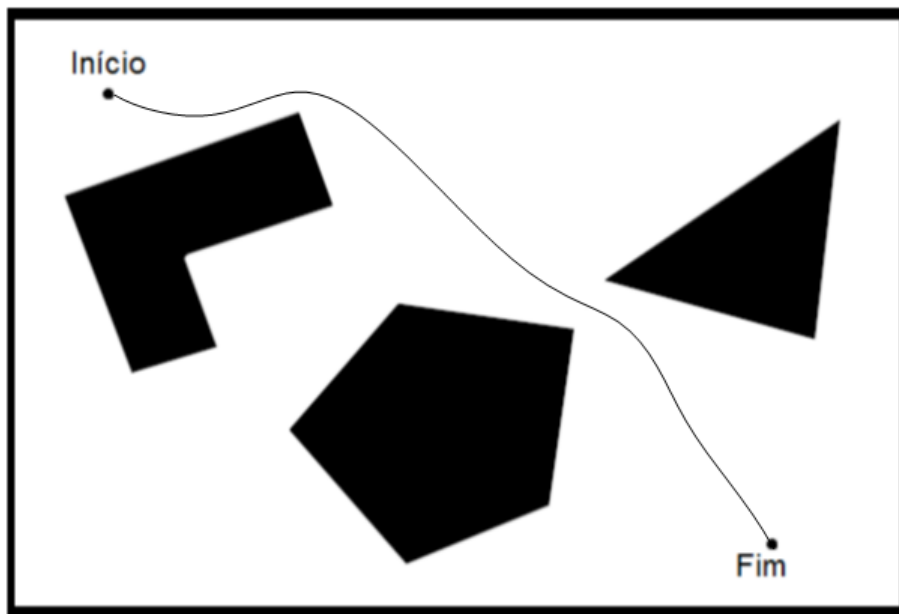


Figura 1.4: Trajetória com o método baseado em Campos Potenciais Artificiais.

Se comparado com o *Mapa de Rotas* ou com a *Decomposição por Células*, pode-se observar que o método APF planeja trajetórias mais seguras, ou seja, afasta-se mais

dos obstáculos detectados, evitando colisões e tendendo a convergir até o ponto final desejado [2, 5, 9].

Por fim, no *controle*, tem-se o acionamento dos atuadores do robô de forma que o mesmo consiga realizar o caminho previamente planejado, atentando-se para as restrições físicas que os motores possuem. Portanto, o controle é cinemático, pois controla-se a posição do robô no ambiente. Nesta etapa, é necessário a aquisição de dados sensoriais dos atuadores, baseado em medições internas através de sensores classificados em *proprioceptivos*, como os *encoders*, que medem as velocidades nos eixos dos motores.

Observa-se na literatura o uso de duas estratégias de controle cinemático aplicado a veículos diferenciais, uma usando o controlador não linear invariante no tempo (NTI) e outra fazendo a linearização por realimentação dinâmica (DFL) [10, 11, 12]. Ambas essas técnicas são de simples implementação e apresentam resultados satisfatórios relativos ao controle de percurso do veículo. Para essas estratégias, com vistas a facilitar a discussão, não são considerados os efeitos dinâmicos resultantes do acionamento dos motores do veículo, visto que tais efeitos apresentam uma dinâmica muito mais rápida que a de variação da posição do veículo no tempo [12]. Dessa forma, os efeitos dinâmicos dos motores podem ser considerados como perturbações do sistema, sendo implementados em simulação e observados experimentalmente para analisar seu efeito nos resultados do controle.

O foco deste trabalho está principalmente na etapa de planejamento, sendo que a principal contribuição está na implementação de uma modificação no algoritmo do método APF, visando superar limitações. Na etapa de controle optou-se pela implementação de dois tipos de controladores cinemáticos (DFL e NTI). A seguir é exposto o objetivo central deste trabalho.

1.1 Objetivo

O objetivo deste trabalho é realizar o desenvolvimento de um sistema de navegação, com planejamento da trajetória e controle cinemático, para que um veículo diferencial seja capaz de desviar dos obstáculos previamente detectados no ambiente. Na Figura 1.5, pode-se observar o planejamento e controle de um veículo diferencial com desvio de obstáculos.

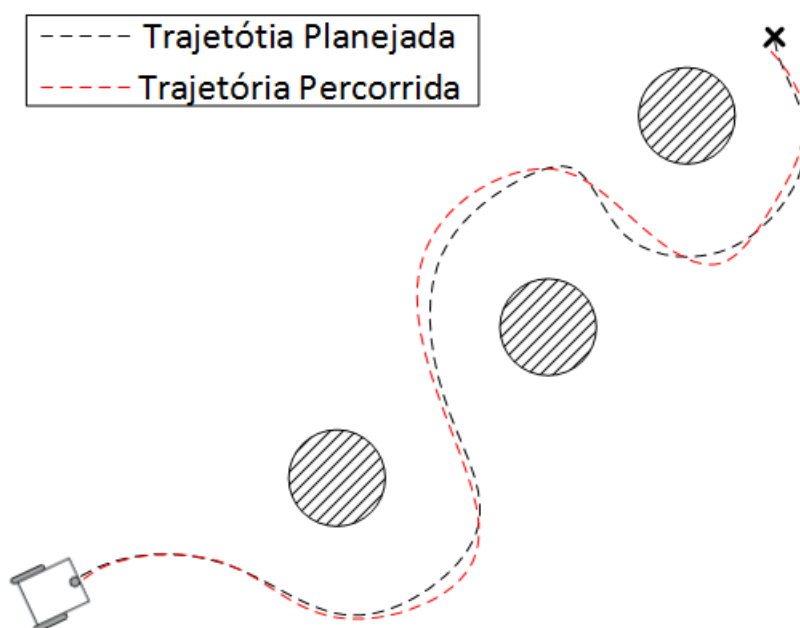


Figura 1.5: Planejamento e Controle da Trajetória para desvio de obstáculos.

Experimentalmente, para testar o sistema de controle do veículo em laboratório, toma-se como conhecido todo o ambiente no qual o robô irá se deslocar, ou seja, não é realizada a etapa de mapeamento do meio através de sensores reais, como ultra-som ou infravermelho.

1.2 Motivação

Foi observado em outros trabalhos [4, 13] que a etapa de planejamento nem sempre gera trajetórias com caminhos seguros para o desvio de obstáculos e convergência ao ponto final desejado. Se comparado a outros métodos, o método de planejamento baseado em campos potenciais artificiais é o que planeja as trajetórias mais seguras, que tendem a convergir para o ponto final desejado com um maior afastamento entre o veículo e os obstáculos detectados, evitando as colisões [9]. No entanto, sabe-se que esse método possui limitações decorrentes, principalmente, do problema dos mínimos locais, que consiste basicamente no fato de que, dependendo das configurações iniciais do robô e da disposição dos obstáculos no meio, a trajetória planejada pode não convergir até a referência, mesmo que exista algum caminho factível [1, 5]. A Figura 1.6 ilustra graficamente um ponto de mínimo local devido a um obstáculo

côncavo. Caso o robô chegue a este ponto de mínimo local ou “vale”, o mesmo não conseguirá sair, pois a função potencial não irá mais variar, fazendo com que não seja atingido o ponto de referência desejado [8].

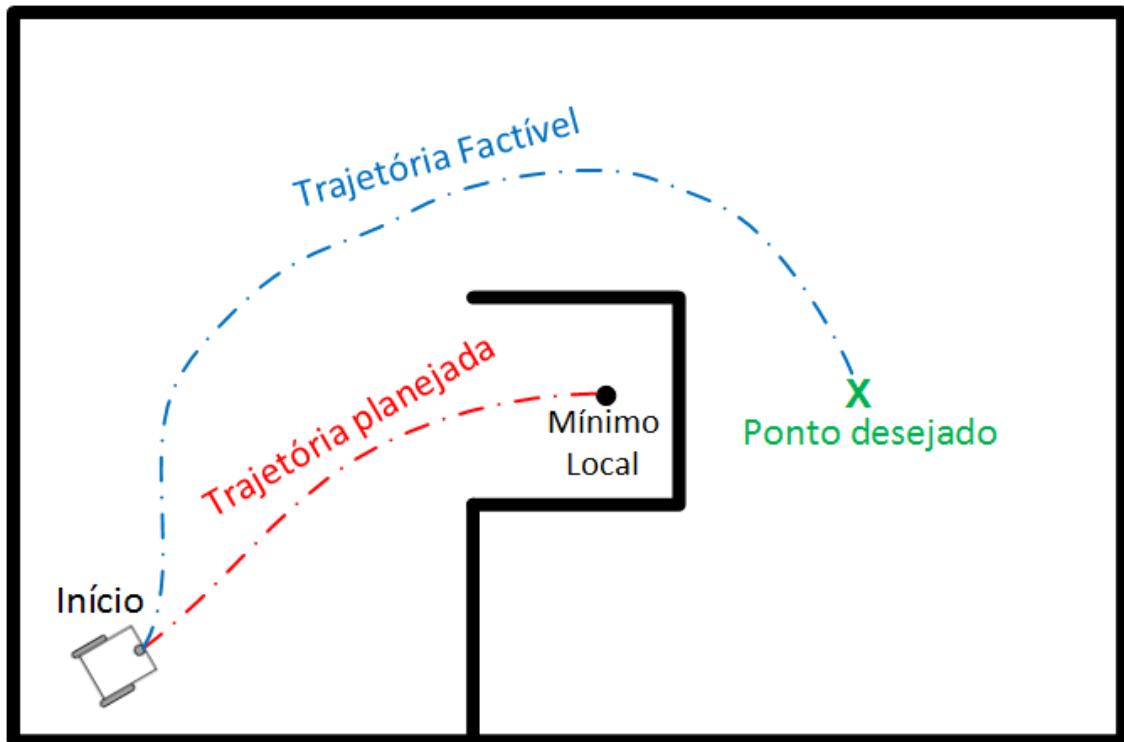


Figura 1.6: Mínimo local devido a um obstáculo côncavo.

Portanto, a principal motivação deste trabalho está na implementação de uma nova abordagem no planejamento de trajetórias baseado no método APF, de modo que seja superado o problema dos mínimos locais, ou seja, planejando trajetórias seguras que se afastam dos obstáculos detectados e evitam as colisões, ao mesmo tempo em que converge até o ponto final desejado, caso haja algum caminho factível.

Outra motivação está no fato de que esse desenvolvimento poderá ser aplicado futuramente no contexto de uma cadeira de rodas motorizada, um tipo de veículo diferencial [14], visando proporcionar maior mobilidade e segurança aos usuários, já que em alguns casos, de pessoas com limitações físicas mais severas, há uma clara dificuldade em controlar precisamente a cadeira de rodas motorizada através de um *joystick*. Dessa forma, um sistema autônomo de navegação que leve o cadeirante até um determinado ponto desejado diminuirá significativamente a ocorrência de colisões que podem gerar graves acidentes.

Além disso, este trabalho também apresenta estudos e desenvolvimentos que possuem aplicação direta em pesquisa e novos projetos na área da robótica móvel, utilizando sistemas robóticos similares.

1.3 Organização da Dissertação

No Capítulo 2, é apresentada a fundamentação teórica do trabalho, necessária para a compreensão e desenvolvimento do sistema de navegação proposto. São apresentados os modelos cinemático e dinâmico para um veículo diferencial de duas rodas genérico e as características da trajetória não-holonômica que esse tipo de veículo é capaz de percorrer. Além disso, são mostrados os tipos de controle cinemático e o método APF para planejamento de trajetória.

No Capítulo 3, tem-se o desenvolvimento do sistema de navegação, incluindo a modelagem cinemática e dinâmica do robô diferencial Lego EV3, usado como modelo de veículo diferencial, em que são implementadas as duas estratégias de controle cinemático (DFL e NTI), bem como o método APF para planejamento de trajetória. Além disso, é apresentado o desenvolvimento do algoritmo com as modificações propostas para o método APF clássico, visando superar o problema dos mínimos locais.

No Capítulo 4, são apresentados os resultados de simulação e experimentais utilizando o robô Lego EV3, em que são feitas análises e comparações destes resultados.

Por fim, no Capítulo 5, são apresentadas as considerações finais do trabalho, mostrando o que foi alcançado como contribuição e as perspectivas de desdobramentos futuros em torno do que foi desenvolvido.

Capítulo 2

FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão apresentados os conceitos teóricos requeridos para o desenvolvimento do trabalho proposto, tais como modelagem cinemática e dinâmica de veículos com acionamento diferencial, estratégias de controle cinemático e o método para planejamento de trajetórias baseado em campos potenciais artificiais.

2.1 Arquitetura de Navegação para Robôs Móveis

Sistemas robóticos podem ser classificados em robôs manipuladores e móveis, que por sua vez são classificados em terrestres, aquáticos ou aéreos [2]. Neste trabalho, serão feitos estudos e desenvolvimentos para os robôs móveis terrestres.

Sabe-se que um robô móvel terrestre apresenta uma arquitetura de navegação composta de um sistema de *hardware* e *software* que garantem a execução das mais diversas tarefas [15]. Existem basicamente cinco níveis hierárquicos da arquitetura de navegação, ordenados do nível mais alto para o mais baixo, que são [12, 13]:

- Percepção;
- Decisão;
- Planejamento do Caminho;
- Geração de Trajetória e
- Sistema de Controle.

A *Percepção* corresponde ao sensoriameto do robô, o qual, neste trabalho, é realizado apenas em simulação, com reconhecimento gradativo do ambiente durante o movimento do veículo, detectando novos obstáculos localizados a uma delimitada distância de sua atual posição, denominada de raio de detecção.

O nível de *Decisão* é o responsável por fornecer os dados de entrada para realizar o planejamento da trajetória, saindo de um ponto inicial conhecido e atingindo um ponto final desejado. Estas decisões variam de acordo com o tipo de tarefa a ser executada.

No *Planejamento do Caminho* tem-se efetivamente o algoritmo responsável por fornecer a sequência de coordenadas necessárias para que o robô atinga um determinado ponto de referência sem que ocorra a colisão com os obstáculos do meio. Nesse nível, busca-se usar métodos que planejam caminhos levando-se em conta o objetivo desejado.

A *Geração de Trajetória* consiste em, a partir de um planejamento prévio, realizar um ajuste na curva de trajetória, a fim de suavizá-la e facilitar o posterior controle cinemático.

Por fim, o *Sistema de Controle* é o nível que realiza o acionamento efetivo dos atuadores do robô, responsáveis por fazê-lo percorrer a trajetória previamente planejada e ajustada.

Para a implementação do sistema de planejamento e geração de trajetórias, em conjunto com a aplicação do sistema de controle, é necessário conhecer o modelo do robô móvel para saber como aplicar os sinais necessários a fim de que seja percorrida a trajetória previamente planejada. A seguir serão abordadas as características dos robôs móveis com acionamento diferencial.

2.2 Robôs Móveis com Acionamento Diferencial

Os robôs móveis com acionamento diferencial podem controlar independentemente suas duas rodas para realizar um determinado movimento, já que apresentam um motor para cada roda. Esses robôs realizam trajetórias caracterizadas como não-holonômicas, sendo também conhecidos como um tipo de sistema não-holonômico [13]. Os sistemas não-holonômicos são definidos como sendo de dimensão finita, ou

seja, onde ocorre uma ou mais restrições impostas em um ou mais estados do sistema [10]. Dessa forma, um robô diferencial não é capaz de se movimentar em todas as direções¹, como por exemplo um movimento lateral perpendicular à direção na qual o robô esteja.

Na Figura 2.1, tem-se um exemplo clássico, ilustrando o estacionamento de um robô com restrições não-holonômicas, quando o mesmo está numa determinada posição e orientação inicial perpendicular à trajetória holonômica.

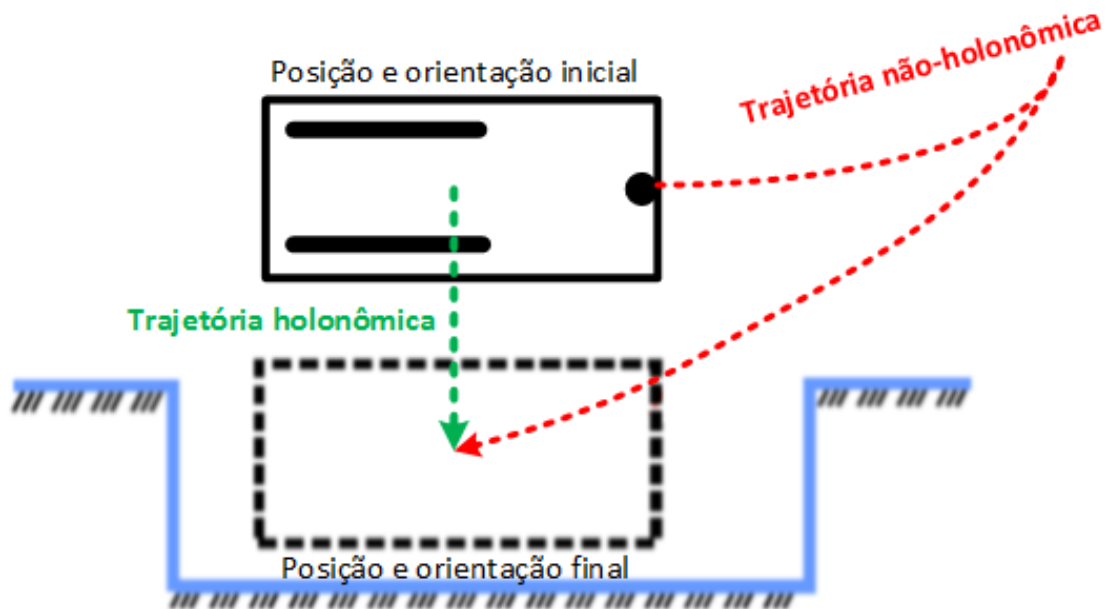


Figura 2.1: Estacionamento de um robô móvel com restrições não-holonômicas.

Pode-se notar que, neste caso, para o robô atingir a posição e orientação final desejadas, é necessário realizar uma trajetória diferente da holonômica, que seria a mais simples, visto que não é possível que o robô execute um movimento perpendicular à sua orientação inicial. Portanto, o robô realiza uma trajetória não-holonômica para atingir a posição e orientação finais desejadas.

A geometria básica dos robôs móveis diferenciais consiste em um raio r das rodas e em uma distância L entre os eixos das rodas. Com isso, é possível estimar as distâncias percorridas a partir da aplicação das velocidades nas rodas.

¹Considera-se também que não há derrapagem da roda no movimento do robô e nem que o mesmo gire em torno de seu próprio eixo sem se deslocar no plano.

Na Figura 2.2, pode-se observar as medidas necessárias para formular o modelo cinemático de um robô móvel com tração diferencial.

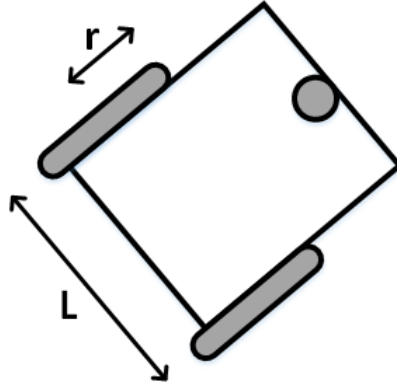


Figura 2.2: Geometria básica de um robô móvel com tração diferencial.

A seguir serão obtidos os modelos cinemático e dinâmico desse tipo de robô. De posse desses modelos, pode-se realizar o controle cinemático para que o robô execute as trajetórias desejadas, como será visto posteriormente na Seção 2.3.

2.2.1 Modelo Cinemático

A partir das velocidades de rotação medidas nas rodas direita e esquerda, pode-se determinar a velocidade linear v e a velocidade angular w do robô [15]:

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} r/2 & r/2 \\ r/L & -r/L \end{bmatrix} \cdot \begin{bmatrix} \omega_{r_D} \\ \omega_{r_E} \end{bmatrix}, \quad (2.1)$$

sendo ω_{r_D} e ω_{r_E} as velocidades de rotação nas rodas direita e esquerda, respectivamente.

Supõe-se que em um dado instante de tempo discreto k o robô encontra-se na posição $(x(k), y(k))$, com ângulo de rotação $\theta(k)$, em relação ao sistema de coordenadas global, como evidenciado na Figura 2.3.

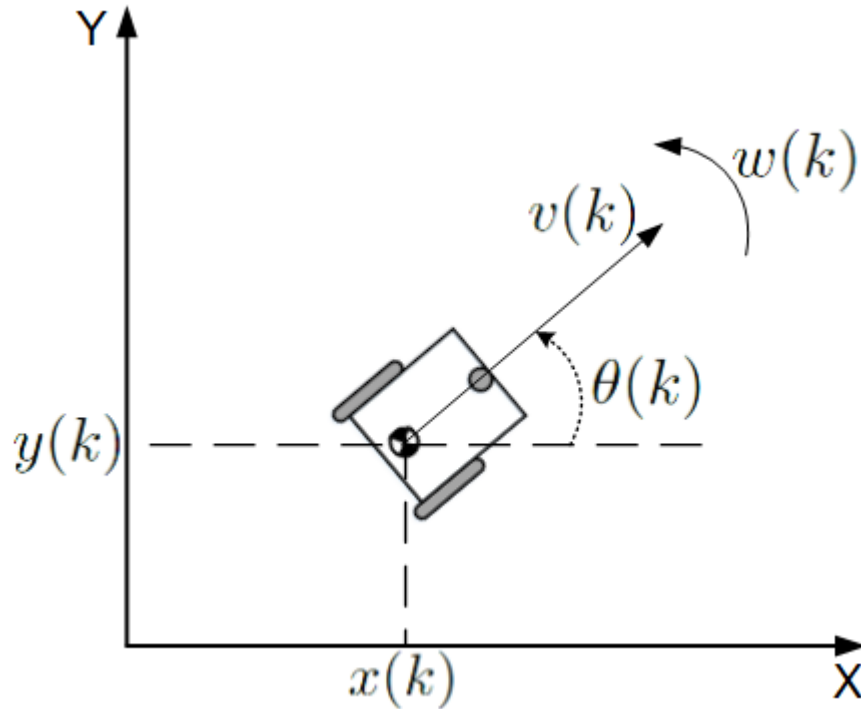


Figura 2.3: O robô no sistema de coordenadas global.

Após serem aplicados os sinais de controle nos motores das rodas, o robô passa para o próximo estado, de acordo com a equação [2, 15]:

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} + R \cdot \begin{bmatrix} v(k) \\ w(k) \end{bmatrix} \cdot t_s, \quad (2.2)$$

sendo t_s o período de amostragem e R a matriz de rotação, dada por:

$$R = \begin{bmatrix} \cos(\theta(k)) & 0 \\ \sin(\theta(k)) & 0 \\ 0 & 1 \end{bmatrix}. \quad (2.3)$$

Será evidenciado a seguir o modelo dinâmico, referente ao sistema de motorização do robô, ou seja, serão observadas as restrições dos respectivos atuadores, tais como aceleração e velocidade máxima que o robô é capaz de atingir.

2.2.2 Modelo Dinâmico

O modelo dinâmico do robô está associado à dinâmica dos motores de suas rodas. O sistema de motorização do robô diferencial, utilizado como objeto de estudo neste trabalho, possui duas rodas com motores elétricos de pequeno porte, que podem ser modelados como sistemas de primeira ordem [16]:

$$\frac{\omega_r(z)}{u_r(z)} = \frac{A}{z - B}, \quad (2.4)$$

sendo ω_r a velocidade de rotação da roda (direita ou esquerda) e u_r o sinal de controle aplicado, que neste caso também é um sinal de velocidade de rotação da respectiva roda.

Pode-se obter os valores de A e B através do método de modelagem do tipo caixa preta por meio da resposta ao degrau [17], sendo necessário realizar ensaios nos motores, fazendo o robô sair do repouso até atingir sua velocidade linear máxima v_{max} . A partir dos ensaios, tem-se a aceleração linear máxima a_{max} do robô, usada para determinar o valor de A , de acordo com a seguinte equação:

$$A = \frac{a_{max} \cdot t_s}{v_{max}}. \quad (2.5)$$

O sinal de controle u_r deve ser a velocidade de referência, de modo que o ganho estático para a Equação (2.4) seja unitário devido ao controlador interno assegurar seguimento de referência de velocidade. Portanto, pode-se determinar B através da equação a seguir:

$$B = 1 - A. \quad (2.6)$$

Por fim, tem-se a equação que representa o modelo dinâmico do robô no tempo k discreto:

$$\omega_r(k + 1) = B \cdot \omega_r(k) + A \cdot u_r(k). \quad (2.7)$$

Também é possível obter os valores de A e B sem necessariamente fazer os ensaios, através de outros métodos de modelagem, desde que se tenha as informações técnicas referente aos motores elétricos de corrente contínua que são usados nas rodas [17].

2.3 Controle Cinemático

Na literatura são encontradas algumas técnicas já consagradas que realizam o controle cinemático para percorrer uma dada trajetória, visto em [10, 11, 12]. No entanto, tais estratégias são baseadas na simplificação de que o robô responde quase que instantaneamente ao sinal de controle aplicado, ou seja, a dinâmica resultante da mudança de velocidade no movimento robótico é considerada desprezível [12]. Com essa simplificação, o modelo tomado como base para o controle é apenas o cinemático, que foi evidenciado na Seção 2.2.1.

No entanto, é importante conhecer o modelo dinâmico para que o mesmo seja implementado em simulação, de modo a observar os efeitos da dinâmica resultante da mudança de velocidade na etapa de controle e constatar se a estratégia desenvolvida consegue fazer o veículo se deslocar dentro da trajetória desejada.

O objetivo da etapa de controle é, então, fazer com que o robô passe pelas posições (x_d, y_d) relativas a uma trajetória previamente planejada. Os sinais de controle são as velocidades linear v_s e angular w_s aplicadas no robô. Portanto, controla-se a posição aplicando sinais de velocidades, como ilustrado através do diagrama de blocos da Figura 2.4.

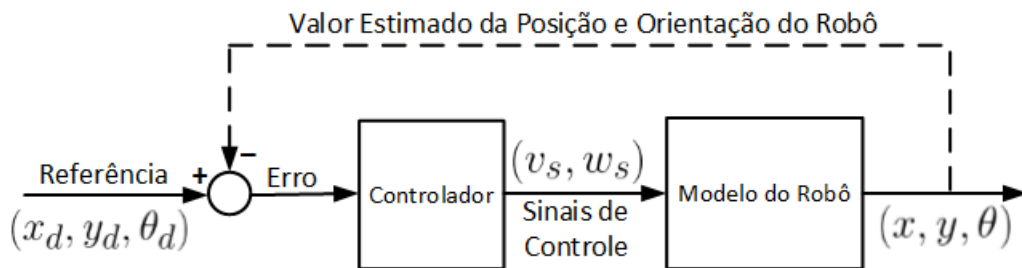


Figura 2.4: Diagrama de blocos do sistema de controle.

Para o robô percorrer todo o caminho, as velocidades linear v_d e angular w_d de referência ao longo da trajetória são calculadas de acordo com a equação [10]:

$$\begin{aligned} v_d &= \sqrt{(\Delta x_d)^2 + (\Delta y_d)^2} \\ w_d &= \frac{\Delta^2 y_d \cdot \Delta x_d - \Delta^2 x_d \cdot \Delta y_d}{(\Delta x_d)^2 + (\Delta y_d)^2}, \end{aligned} \quad (2.8)$$

sendo os operadores Δ e Δ^2 definidos por:

$$\begin{aligned}\Delta\xi(k) &= \frac{\xi(k) - \xi(k-1)}{t_s} \\ \Delta^2\xi(k) &= \frac{\Delta\xi(k) - \Delta\xi(k-1)}{t_s} \\ \xi &= \{x_d, y_d\}\end{aligned}\tag{2.9}$$

A seguir serão abordados dois tipos de controladores: usando a *Linearização por Realimentação Dinâmica* e o controlador *Não-Linear Invariante no Tempo*.

2.3.1 Linearização por Realimentação Dinâmica - DFL

A Linearização por Realimentação Dinâmica (DFL - *Dynamic Feedback Linearization*) é uma estratégia de controle bem conhecida no contexto cinemático do ramo da robótica móvel. O objetivo é atingir uma linearização do modelo cinemático do robô a cada vez em que ocorra uma mudança de estado [18]. Dessa forma, a todo tempo é feita uma nova linearização, visando o melhor seguimento da trajetória. Na equação a seguir tem-se a lei de controle DFL [12, 18]:

$$\begin{aligned}v_s(k) &= v_o + a(k).t_s \\ w_s(k) &= -(u_1(k)/v_o).sin(\theta(k)) + (u_2(k)/v_o).cos(\theta(k)),\end{aligned}\tag{2.10}$$

sendo v_o a velocidade linear do robô no instante anterior, que deve ser diferente de zero para que não haja singularidade. As variáveis u_1 e u_2 são análogas à decomposição da aceleração linear horizontal (eixo x) e vertical (eixo y), resultando na aceleração linear resultante a . Pode-se calcular u_1 , u_2 e a através das equações evidenciadas a seguir [18]:

$$\begin{aligned}u_1(k) &= \Delta^2x_d(k) + K_{p1}.(x_d(k) - x(k)) + K_{d1}.(\Delta x_d(k) - \Delta x(k)) \\ u_2(k) &= \Delta^2y_d(k) + K_{p2}.(y_d(k) - y(k)) + K_{d2}.(\Delta y_d(k) - \Delta y(k)). \\ a(k) &= u_1(k).cos(\theta(k)) + u_2(k).sin(\theta(k))\end{aligned}\tag{2.11}$$

Com isso, tem-se a lei de controle para a estratégia da Linearização por Realimentação Dinâmica, em que K_{p1} , K_{p2} , K_{d1} e K_{d2} são os parâmetros de controle, escolhidos de modo a tentar obter pequeno erro de posição e o pouco esforço de controle dos motores.

2.3.2 Não-Linear Invariante no Tempo - NTI

O controle não-linear invariante no tempo, também conhecido como NTI (*Nonlinear Time-Invariant control*) é resultado de uma abordagem do projeto de controladores em sistemas não-lineares de Lyapunov. Como pode ser visto em [19], esta estratégia busca uma lei de controle capaz de provar que a função candidata é função de Lyapunov. Na equação evidenciada a seguir, pode-se observar a lei de controle para o controlador NTI [19]:

$$\begin{aligned} v_s(k) &= v_d(k) \cdot \cos(\theta_d(k) - \theta(k)) + k_1 \cdot M_1(k) \\ w_s(k) &= w_d(k) - k_2 \cdot v_d(k) \cdot S_\theta(k) \cdot M_2(k) + k_3 \cdot (\theta_d(k) - \theta(k)), \end{aligned} \quad (2.12)$$

sendo:

$$\begin{aligned} \theta_d(k) &= \tan^{-1}(\Delta y_d(k)/\Delta x_d(k)) \\ \theta(k) &= \tan^{-1}(\Delta y(k)/\Delta x(k)) \\ M_1(k) &= (x_d(k) - x(k)) \cdot \cos(\theta(k)) + (y_d(k) - y(k)) \cdot \sin(\theta(k)) \\ M_2(k) &= (y_d(k) - y(k)) \cdot \cos(\theta(k)) - (x_d(k) - x(k)) \cdot \sin(\theta(k)) \\ S_\theta(k) &= \frac{\sin(\theta_d(k) - \theta(k))}{\theta_d(k) - \theta(k)} \end{aligned} \quad (2.13)$$

Os coeficientes k_1 , k_2 e k_3 são obtidos fazendo:

$$\begin{aligned} k_1 = k_3 &= 2 \cdot \zeta \cdot \sqrt{w_d(k)^2 + b \cdot v_d(k)^2} \\ k_2 &= b \end{aligned} \quad (2.14)$$

Portanto, tem-se liberdade na escolha de dois parâmetros de controle: b (análogo à agressividade do controlador) e ζ (alusivo ao fator de amortecimento), sendo $b > 0$ e $0 < \zeta \leq 1$. Tais parâmetros interferem diretamente nos valores de k_1 , k_2 e k_3 , sendo escolhidos de modo a obter o menor erro de posição e menor esforço de controle por parte dos atuadores do sistema.

Os sinais de controle da Equação (2.12) são aplicados para que o robô percorra a trajetória definida pelos pontos (x_d, y_d) . Esta trajetória deverá ser planejada de modo que o robô desvie dos obstáculos dispostos no ambiente, como será discutido a seguir.

2.4 Planejamento da Trajetória

Na etapa de planejamento são determinados os pontos (x_d, y_d) para que o robô atinja uma dada referência e desvie dos obstáculos. Existem diversas técnicas de planejamento de trajetórias, como visto em [1, 2], todavia este trabalho aborda e utiliza um método de planejamento baseado em campos potenciais artificiais, conhecido como método APF. Este método entende o robô, analogamente, como uma partícula de carga positiva que se desloca para o ponto de referência, representado como uma partícula fixa com carga negativa, mas se afasta dos obstáculos que têm carga positiva [2], seguindo um princípio básico da eletricidade: *cargas iguais se repelem e as opostas se atraem*. A Figura 2.5 ilustra o planejamento pelo método APF seguindo essa analogia física.

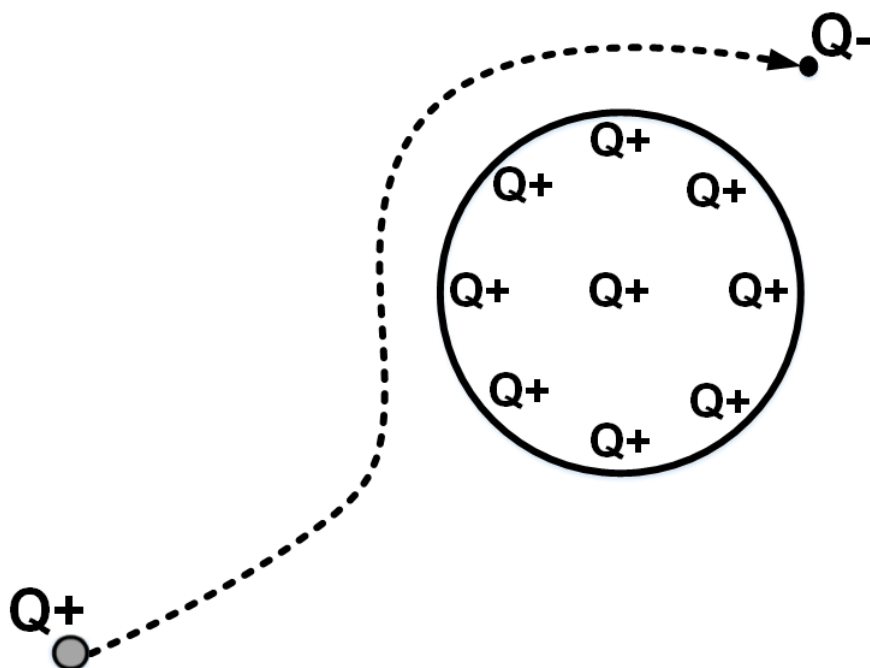


Figura 2.5: Trajetória obtida através do método de campos potenciais.

2.4.1 Funções Potenciais

De acordo com [2, 5], o APF é um método de simples implementação, cuja modelagem matemática consiste em um campo potencial, descrito como sendo uma função potencial, real e diferenciável, definida por U . Através de uma aproximação

de tempo discreto, pode-se calcular o gradiente da função potencial de acordo com a equação dada por:

$$\left(\frac{\Delta x_d}{t_s}, \frac{\Delta y_d}{t_s} \right) = -\vec{\nabla}U(x_d, y_d), \quad (2.15)$$

sendo que o potencial $U(x_d, y_d)$ é a soma do potencial $U_{att}(x_d, y_d)$ de atração para o ponto de referência com o potencial $U_{rep}(x_d, y_d)$ de repulsão, referente aos obstáculos detectados. Logo:

$$U(x_d, y_d) = U_{att}(x_d, y_d) + U_{rep}(x_d, y_d) \quad (2.16)$$

Sabe-se que não existe uma única escolha para as funções potenciais, no entanto é necessário que o potencial $U_{att}(x_d, y_d)$ seja uma função que garanta mínimo global no ponto de referência (x_{ref}, y_{ref}) e que $U_{rep}(x_d, y_d)$ seja monotonicamente crescente em relação ao inverso da distância entre robô e obstáculo. Além disso, o potencial repulsivo não tem alcance infinito, de forma que para distâncias maiores que D_{max} o valor $U_{rep}(x_d, y_d)$ é considerado nulo, ou seja, o valor de D_{max} é escolhido com base na proximidade máxima que se deseja chegar dos obstáculos [2, 8].

Com essas condições, pode-se escolher as funções potenciais de atração e repulsão, definidas por:

$$U_{att}(x_d, y_d) = \frac{\beta}{2} \cdot ((x_{ref} - x_d)^2 + (y_{ref} - y_d)^2), \quad (2.17)$$

$$U_{rep_i}(x_d, y_d) = \begin{cases} \frac{\eta}{2} \left(\frac{1}{d_{b_i}} - \frac{1}{D_{max}} \right)^2, & d_{b_i} < D_{max} \\ 0, & d_{b_i} \geq D_{max} \end{cases}, \quad (2.18)$$

$$U_{rep}(x_d, y_d) = \sum_{i=1}^j U_{rep_i}(x_d, y_d), \quad (2.19)$$

sendo β e η as escalas de gradiente atrativo e repulsivo, respectivamente, j o número total de obstáculos detectados no ambiente e (x_{b_i}, y_{b_i}) o ponto do obstáculo i com a menor distância d_{b_i} para o robô, portanto:

$$d_{b_i} = \sqrt{(x_d - x_{b_i})^2 + (y_d - y_{b_i})^2}. \quad (2.20)$$

Na Figura 2.6, tem-se a representação gráfica de d_{b_i} em um dado instante em que se detectou um obstáculo no meio.

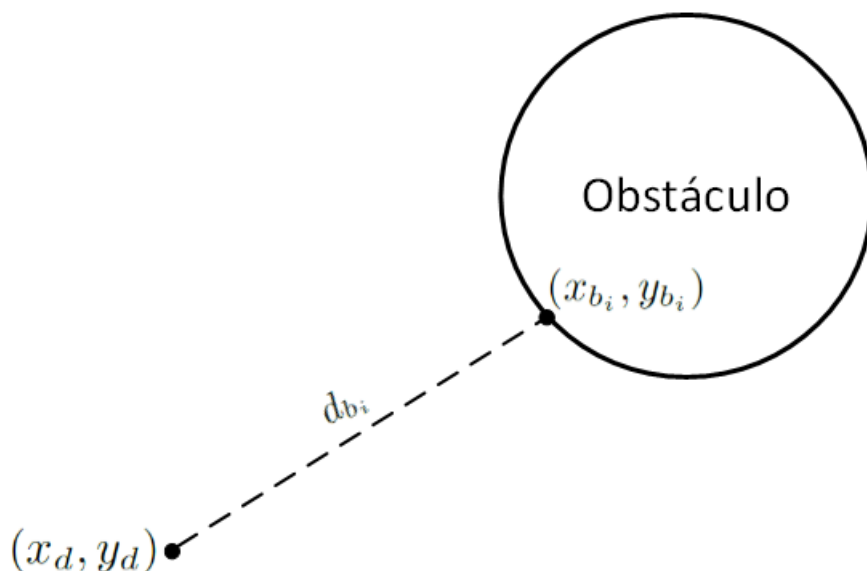


Figura 2.6: Menor distância para um obstáculo.

A seguir será visto um método para adaptação da trajetória, a fim de que seja exequível, exigindo menos dos atuadores, para um robô com tração diferencial de duas rodas.

2.4.2 Adaptação da Trajetória com Interpolação Cúbica

Com o uso das funções potenciais, pode-se encontrar os pontos (x_d, y_d) a cada nova iteração e assim planejar a trajetória que levará o robô ao mínimo local da função potencial, visando atingir a referência e desviar dos obstáculos [2].

Para adaptar um melhor caminho, utiliza-se um método de interpolação cúbica, denominado também de função *spline*, em que é gerada uma trajetória mais suave e compatível com a característica de trajetórias não-holonômicas que os robôs com tração diferencial de duas rodas podem realizar, além de serem adaptadas para possuir um número de pontos de tal forma que a velocidade do robô não varie muito da média desejada, o que acaba exigindo menos do sistema de motorização.

Essa trajetória adaptada é obtida através da aproximação de um conjunto qualquer definido de pontos para uma função parametrizada não linear. A Figura 2.7

ilustra um exemplo para geração de uma trajetória adaptada, obtida a partir de um planejamento prévio, através da interpolação cúbica com função *spline* [13].

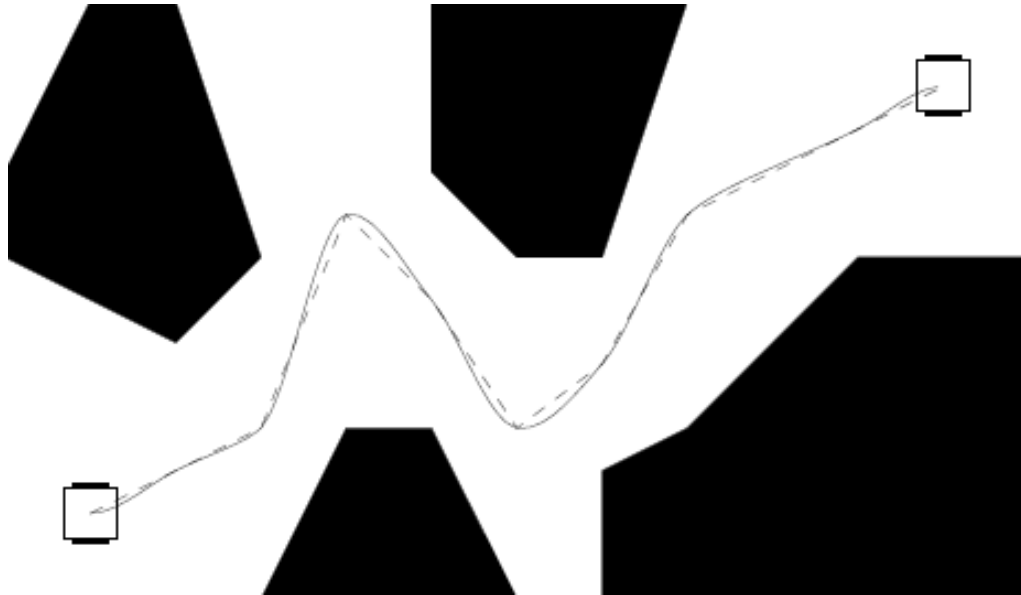


Figura 2.7: Exemplo de geração de trajetória usando interpolação cúbica.

Com o caminho planejado, adaptado e suavizado, pode-se obter uma sequência de pontos (x_d, y_d) que deverão ser colocados na etapa de controle para que o robô execute sua tarefa com eficácia e menor erro de posição.

2.4.3 O Problema dos Mínimos Locais

A trajetória planejada pelo método baseado em campos potenciais sempre levará o robô para um mínimo da função potencial, mas esse ponto não necessariamente será o mínimo global. Desta forma, tem-se uma limitação inerente ao método APF, conhecido como problema dos mínimos locais, que consiste basicamente no fato de que, dependendo das configurações iniciais do robô, dos parâmetros de atração e repulsão escolhidos e da disposição de obstáculos no meio, a trajetória planejada pode não conseguir chegar à referência, mesmo que exista algum caminho factível [1, 5].

A Figura 2.8 ilustra um caso em que o robô não consegue chegar no ponto desejado devido ao fato de ter chegado em um “vale” ou mínimo local.

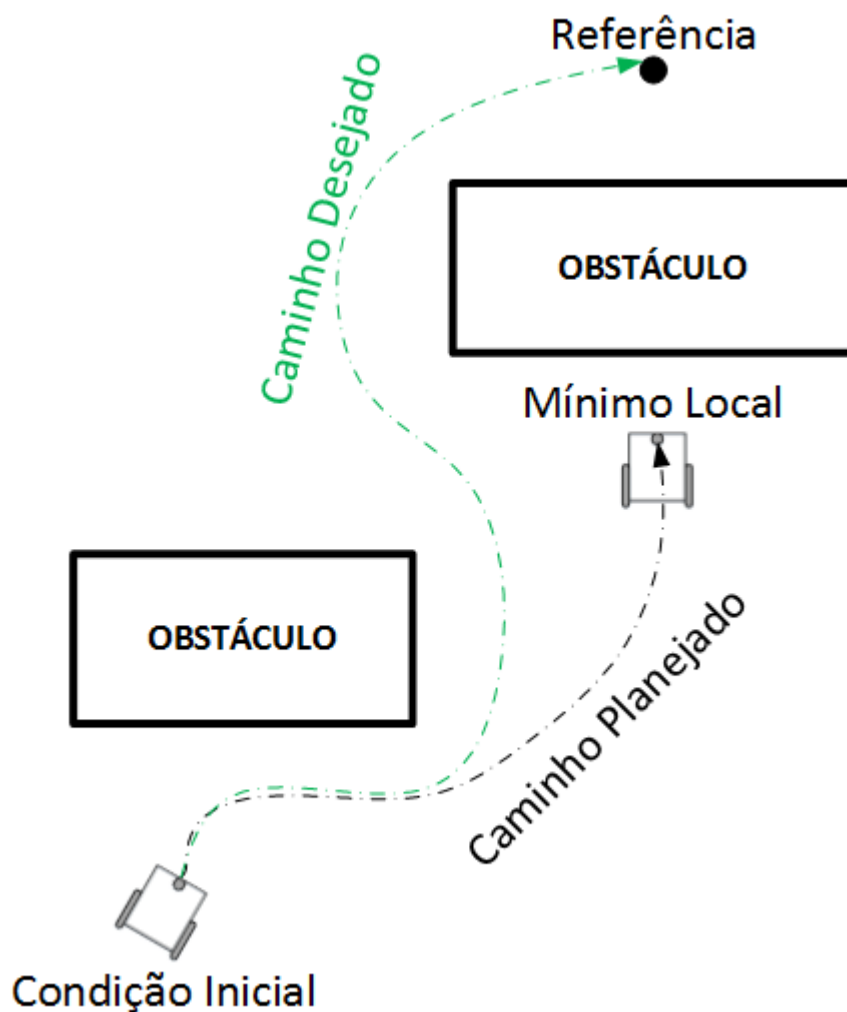


Figura 2.8: Ocorrência do Problema dos Mínimos Locais.

Nota-se que o robô não conseguiu atingir o ponto final desejado, já que as configurações iniciais de intensidade de atração e repulsão foram de tal forma que induziu o robô a chegar em um ponto onde a atração e repulsão se igualam e anulam o deslocamento, ou seja, um ponto de mínimo local da função potencial.

O problema dos mínimos locais pode ocorrer em diversas situações, como quando há obstáculo côncavo, na passagem entre dois obstáculos ou quando o ponto final desejado está próximo de um obstáculo [8].

2.5 Conclusão

Neste Capítulo, foi possível conhecer os aspectos teóricos necessários para o entendimento do trabalho. Viu-se como é o modelo dinâmico e cinemático de um robô diferencial de duas rodas. A partir do modelo cinemático é traçada a estratégia de controle para fazer com que o veículo permaneça em uma trajetória pré-estabelecida. Por fim, é visto o método APF utilizado para obter trajetórias que visem atingir um determinado ponto desejado e desviar dos obstáculos dispostos no ambiente. No método APF viu-se que através de funções potenciais são obtidos os pontos de uma trajetória que deve ser suavizada posteriormente através de um método de interpolação cúbica, de modo que sejam geradas trajetórias não-holonômicas e mais fáceis de serem seguidas na etapa de controle subsequente.

Observou-se também que o método APF apresenta uma limitação inerente, de nem sempre conseguir atingir o ponto final desejado, podendo parar a sua trajetória em um ponto de mínimo local, onde as intensidades de atração para a referência e de repulsão dos obstáculos se igualam, deixando o robô preso neste ponto de mínimo local da função potencial.

No Capítulo 3 a seguir, será discutido, dentre outros aspectos relativos ao desenvolvimento do sistema de navegação, uma modificação para o método APF, a fim de superar a limitação decorrente do problema dos mínimos locais, fazendo com que sempre seja planejado um caminho até o ponto de referência quando houver uma trajetória factível.

Capítulo 3

SISTEMA DE NAVEGAÇÃO

Neste capítulo, serão apresentados os detalhes sobre o desenvolvimento do sistema de navegação, englobando o controle cinemático para seguir uma determinada trajetória, bem como o planejamento de trajetórias para desvio de obstáculos, em que são evidenciados os algoritmos implementados, incluindo o algoritmo proposto para modificar o método APF, que visa superar o problema dos mínimos locais.

3.1 O Robô Diferencial Lego EV3

Existem *kits* de robótica comercializados por determinadas empresas com o intuito de facilitar o desenvolvimento e experimentos de projetos na área. Um desses *kits* é o *kit Lego Mindstorm EV3*, da empresa dinamarquesa *Lego*, que em sua atual geração (EV3) contém pacotes de *software* e uma estrutura de *hardware* que permitem a criação de diversos tipos de robôs customizados e programados [20].

Neste trabalho, é usado o *kit Lego Mindstorm EV3* para fazer a montagem de um robô móvel com duas rodas e acionamento diferencial. Esse robô móvel, vide Figura 3.1, será a base de testes do sistema de navegação desenvolvido.

Geometricamente, o robô tem o valor do raio de suas rodas r e da distância entre os eixos das rodas L iguais a $2,8\text{cm}$ e 12cm , respectivamente.

Na seção a seguir serão observados os ensaios de acionamento do robô diferencial Lego EV3, para que seja realizada posteriormente a modelagem dinâmica de seu movimento.

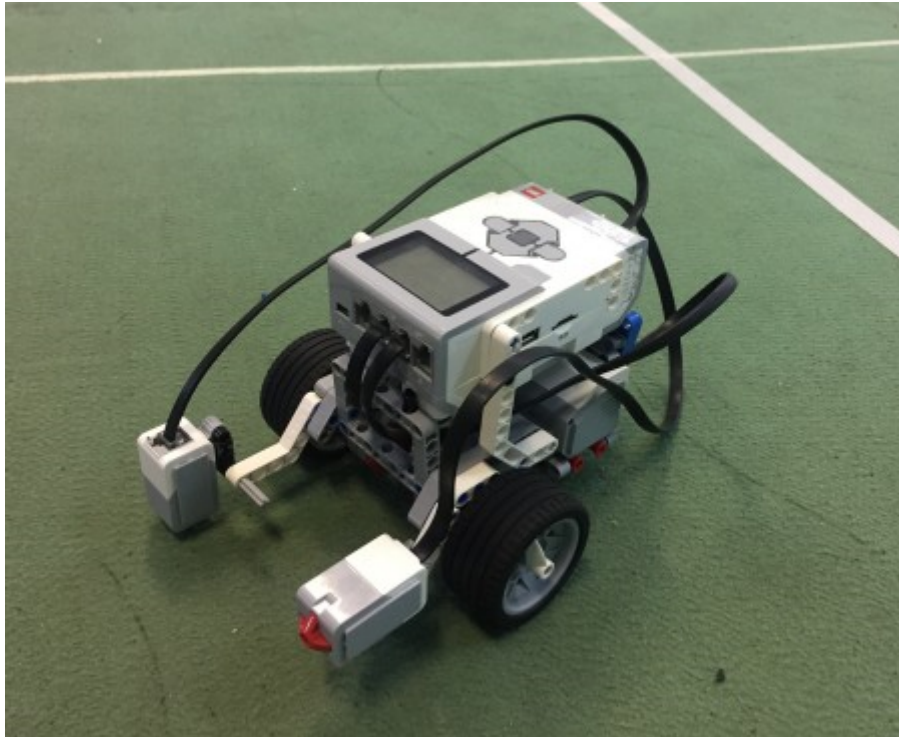


Figura 3.1: Robô móvel com tração diferencial montado com o kit Lego EV3.

3.1.1 Ensaio de Acionamento

Como não há informações sobre a potência dos motores e outros dados técnicos do robô Lego, fez-se necessário realizar um ensaio em laboratório, visando obter uma resposta em malha aberta para que fosse possível encontrar o seu modelo dinâmico usando o método de modelagem com a identificação do tipo caixa preta por meio da resposta ao degrau [17].

O ensaio consiste em aplicar sinais nos motores de forma que o robô saía do repouso e atinja a máxima velocidade linear, com velocidade angular nula. Durante esse processo é possível medir as rotações das rodas, obtendo os valores de velocidade e aceleração linear para que, conseqüentemente, seja encontrado o modelo dinâmico, como já discutido na Seção 2.2.2.

Na Tabela 3.1 a seguir, tem-se os valores obtidos para as medidas das velocidades de rotação nas rodas do robô durante o ensaio, sendo adotado um período de amostragem t_s igual a $100ms$.

Tabela 3.1: Rotação das rodas medidas em cada instante do ensaio.

Tempo (s)	Roda Direita (rad/s)	Roda Esquerda (rad/s)
0	0	0
0,1	0,8727	0,6981
0,2	12,6536	12,7409
0,3	17,7151	17,9769
0,4	18,0642	18,3260
0,5	17,9769	18,9368
0,6	18,5878	18,8496
0,7	18,1514	18,5878
0,8	17,7151	18,2387
0,9	17,7151	18,2386
1,0	17,9769	18,3260
1,1	18,0642	18,6750
1,2	17,6278	18,3260
1,3	17,8896	18,2387
1,4	17,8024	18,1514
1,5	17,6278	18,5878
1,6	17,8896	18,4132
1,7	18,0642	18,5005
1,8	17,7151	18,4132
1,9	18,4132	19,0241
2,0	18,1514	18,5878
2,1	17,8024	18,1514
2,2	17,8024	18,5878
2,3	17,8896	18,5005
2,4	17,9769	18,3260
2,5	17,7151	18,3260
2,6	18,3260	19,0241

Observa-se que existe uma diferença entre as velocidades de rotação nas rodas direita e esquerda, o que implica em um movimento que não foi totalmente linear durante o ensaio. Isso se deve às perturbações do sistema, que fazem os motores de cada roda atuarem de forma um pouco diferente, apesar de possuírem as mesmas características elétricas e mecânicas.

De acordo com a Tabela 3.1, pode-se observar que a máxima diferença entre as medidas das velocidades de rotação das rodas é igual a $0,9599rad/s$, instante em que as velocidades de rotação nas rodas esquerda e direita são iguais a $18,5878rad/s$

e $17,6278\text{rad/s}$, respectivamente. Portanto, constata-se que as perturbações do sistema geram erros menores que 10% para as velocidades aplicadas no robô. Tais erros são levados em consideração nas simulações que serão realizadas posteriormente.

Na Figura 3.2 tem-se graficamente os dados já evidenciados na Tabela 3.1, em que se observa uma resposta característica para um sistema com dinâmica de primeira ordem.

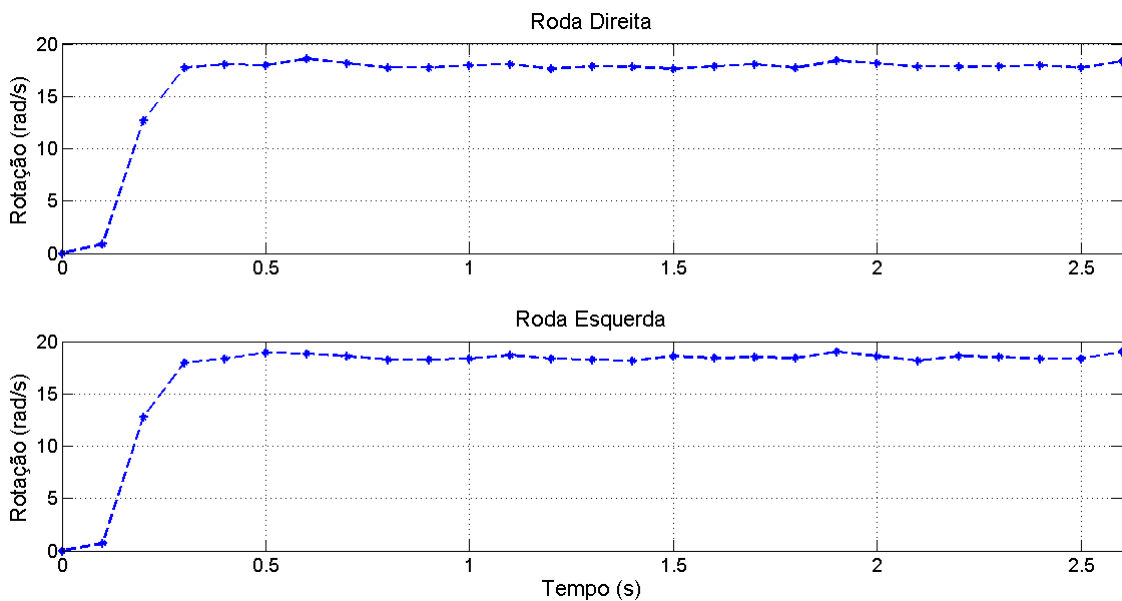


Figura 3.2: Resposta em malha aberta para o acionamento do robô.

A partir desses dados, pode-se obter o modelo do robô, como será discutido na seção a seguir.

3.1.2 Modelo do Robô Lego

De acordo com a Tabela 3.1, pode-se obter o valor da velocidade linear máxima igual a $0,5241\text{m/s}$ e aceleração linear máxima igual a $3,0018\text{m/s}^2$. Com isso, como visto na Seção 2.2.2, são utilizadas as Equações (2.5), (2.6) e (2.7) para obter o seguinte modelo dinâmico de primeira ordem para o robô Lego EV3:

$$\frac{\omega_r(z)}{u_r(z)} = \frac{0,5727}{z - 0,4273}. \quad (3.1)$$

E no domínio do tempo, tem-se:

$$\omega_r(k+1) = 0,4273.\omega_r(k) + 0,5727.u_r(k). \quad (3.2)$$

Na Figura 3.3, tem-se a comparação entre o modelo calculado de acordo com a Equação (3.2), com a dinâmica representada na Figura 3.2, observada no ensaio realizado em laboratório.

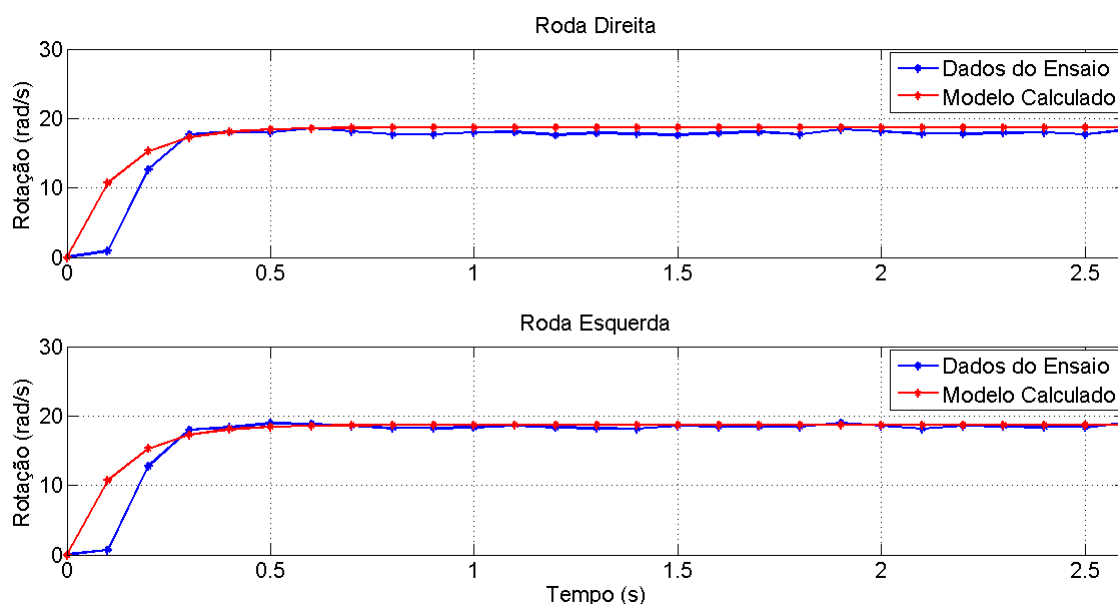


Figura 3.3: Comparação entre a dinâmica calculada com a obtida no ensaio em laboratório.

Apesar da diferença visual observada com os dados do ensaio, o modelo dinâmico calculado foi suficiente para ser utilizado como base para as simulações do acionamento dos motores no movimento robótico.

A modelagem cinemática, como já discutido anteriormente na Seção 2.2.1, depende apenas da geometria básica do robô, ou seja, do valor do raio das rodas e da distância entre os eixos das rodas. Com isso, usando as equações discutidas na Seção 2.2.1, pode-se obter o modelo cinemático do robô Lego EV3.

De posse do modelo cinemático do robô Lego EV3, pode-se desenvolver os algoritmos para implementar o controle cinemático para seguimento de trajetória, como será discutido a seguir.

3.2 Implementação do Controle

A partir do modelo cinemático para o movimento do robô pode-se implementar um controle de modo que o veículo percorra uma dada trajetória definida pelos pontos (x_d, y_d) , conforme já discutido na Seção 2.3. As variáveis de controle v_s e w_s estão em função da posição (x_d, y_d) e da orientação θ_d de referência, bem como da posição (x_e, y_e) e orientação θ_e estimadas, calculadas de acordo com o modelo cinemático do robô. Na Figura 3.4 tem-se um diagrama mais detalhado do sistema de controle, em complemento ao que está ilustrado na Figura 2.4.

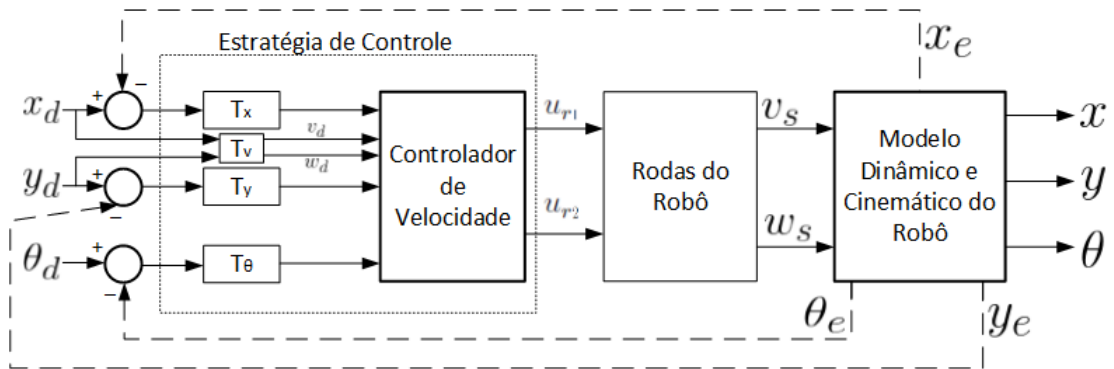


Figura 3.4: Diagrama de blocos detalhado para o sistema de controle.

Os sub-sistemas T_x , T_y e T_θ estão de acordo com a estratégia de controle adotada e podem ser não-lineares. Já o sub-sistema não-linear T_v está de acordo com o elucidado na Equação (2.8), em que são encontrados os valores de velocidades linear e angular necessárias para que o robô atinga determinada posição no tempo, fazendo com que seja executada a trajetória desejada.

Na estratégia de controle, os efeitos oriundos da dinâmica dos atuadores são desprezíveis, visto que os motores variam a velocidade muito mais rapidamente do que ocorre a variação da posição do veículo no plano. Dessa forma, o controle é basicamente cinemático, já que é desejado controlar a posição do robô ao longo do tempo, de forma que ele percorra a trajetória previamente planejada.

Neste trabalho, serão utilizadas duas estratégias diferentes, visando a comparação entre as mesmas. Na primeira estratégia de controle tem-se a utilização de um controlador linear cuja linearização varia de acordo com o estado em que o robô está, conhecido como controlador DFL. Na segunda estratégia de controle utiliza-se

um controlador não-linear cuja lei de controle não varia no tempo, conhecido como controlador NTI. Em ambos os casos o principal desafio é a escolha dos parâmetros de controle para que se tenha o mínimo erro de posição, juntamente com pequeno esforço por parte dos atuadores. Não há uma fórmula precisa que forneça os valores ótimos para os parâmetros de controle, portanto para determiná-los é necessário realizar simulações e verificar em qual combinação de parâmetros se observa o melhor resultado de controle.

3.2.1 O Algoritmo de Controle

O algoritmo de controle consiste em realizar o cálculo da próxima posição desejada para o robô atingir, seguindo a cada período de amostragem a lei de controle DFL da Equação (2.10) ou a lei de controle NTI da Equação (2.12). Em cada iteração, são utilizados dois pontos consecutivos da trajetória planejada, de forma que o robô realize uma linha reta ligando esses pontos.

É necessário calcular o número total de pontos para a trajetória planejada, de forma que a velocidade média fique em torno de 30% da velocidade máxima, o que não exigirá muito do sistema de motorização do robô. O número total de pontos será a razão entre o tempo para percorrer o caminho e o período de amostragem adotado (100ms). O tempo para percorrer o caminho é obtido fazendo a divisão entre a distância total da trajetória planejada e a velocidade média desejada (30% da velocidade máxima), ambos dados conhecidos. Dessa forma, a trajetória é adaptada através da interpolação cúbica, em que é utilizada a função *spline* para suavização da trajetória, como já elucidado na Seção 2.4.2.

Dessa forma, o número de iterações acabará sendo igual ao número total de pontos da trajetória planejada e adaptada. Toda vez que o veículo termina uma iteração, o processo é refeito para o próximo ponto, culminando numa nova trajetória reta para o veículo, até que todo o caminho desejado seja completamente percorrido. A Figura 3.5 ilustra o algoritmo de controle simplificado.

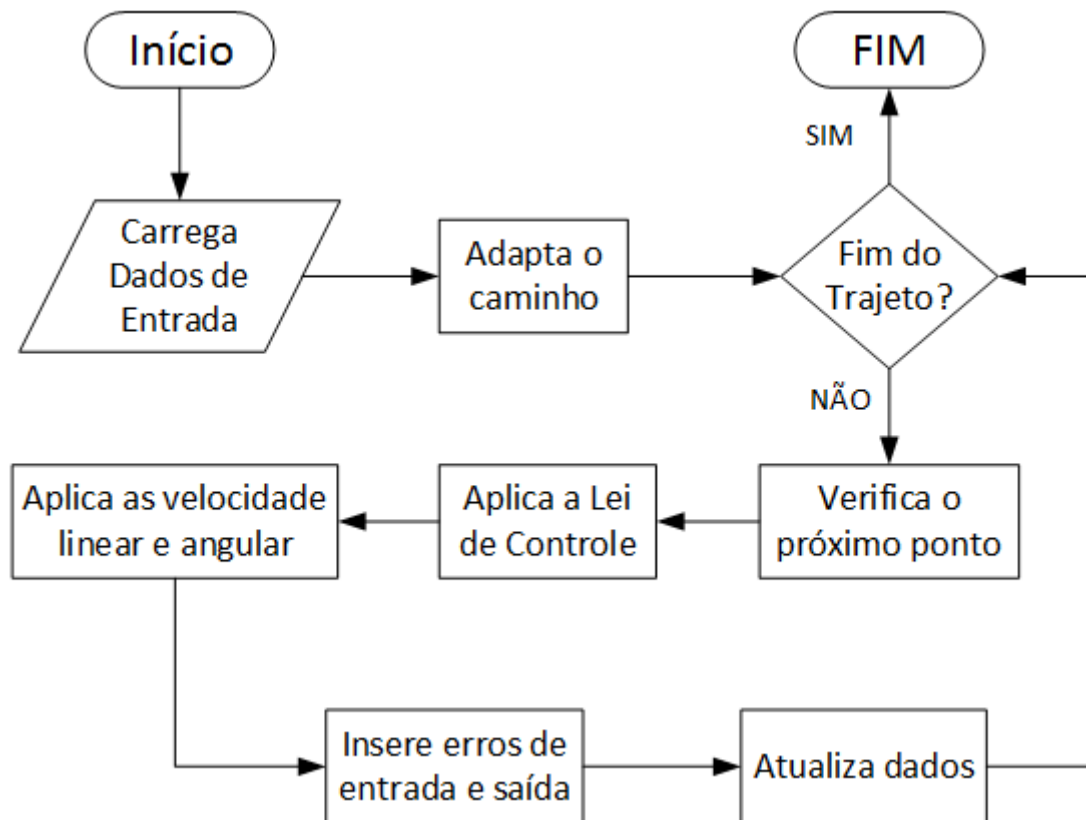


Figura 3.5: Algoritmo de controle simplificado.

Utilizando-se o algoritmo evidenciado, pode-se realizar o controle do seguimento de trajetória, tanto com o controlador DFL quanto com o controlador NTI, variando apenas a lei de controle característica. A seguir, serão observados alguns resultados para o controle cinemático do robô nos dois tipos de controlador, destacando também o esforço de controle através da análise das variações das velocidades linear e angular do robô ao longo do caminho percorrido.

3.2.2 Controlador DFL

Para o controlador DFL é necessário escolher os parâmetros K_{p1} , K_{p2} , K_{d1} e K_{d2} , que serão aplicados na lei de controle DFL, de acordo com o que já foi mostrado na Equação (2.11).

Para encontrar os valores dos parâmetros, são realizadas simulações usando o algoritmo evidenciado na Figura 3.5, em que são obtidos diferentes resultados que foram comparados, de modo que a seguinte escolha, evidenciada na Tabela 3.2, obteve um

resultado considerado visualmente satisfatório referente ao erro de seguimento de trajetória e também referente ao esforço de controle, como será visto posteriormente nos resultados.

Tabela 3.2: Parâmetros escolhidos para o controlador DFL.

Parâmetro	Valor Escolhido
K_{p1}	2
K_{p2}	2
K_{d1}	1
K_{d2}	1

As simulações que são realizadas utilizam os modelos dinâmico e cinemático obtidos para o robô diferencial Lego EV3, sendo incorporado um erro aleatório de no máximo 10% para a medição das velocidades nas rodas (ruído na saída) e um erro aleatório de no máximo 10% na aplicação das velocidades nas rodas do robô (ruído da entrada). Tais erros são alusivos às perturbações abordadas na Seção 3.1.1, referente ao que foi observado no ensaio realizado em laboratório.

Além disso, as simulações são feitas levando-se em consideração as restrições dos motores do robô, ou seja, as máximas velocidades de rotação das rodas, observadas na Tabela 3.1, que implicam numa velocidade linear máxima de $0,5241m/s$, velocidade angular máxima de $8,7350rad/s$, aceleração linear máxima de $3,0018m/s^2$ e aceleração angular máxima de $5,0030rad/s^2$.

O robô está inicialmente em repouso e a trajetória desejada para o percurso é a representada pela função parametrizada dada por:

$$\begin{aligned} x_d(t) &= 0,5.(4.\cos(0,3.t) - \cos(0,5.t)) \\ y_d(t) &= 0,5.(4.\sen(0,3.t) - \sen(1,2.t)) \end{aligned}, \quad t \in [\pi/6, 19.\pi/3]. \quad (3.3)$$

A Figura 3.6 ilustra a simulação para análise do percurso da trajetória, representada na Equação (3.3), pelo robô Lego EV3, em que foram utilizados os parâmetros escolhidos da Tabela 3.2.

Nota-se que a escolha adotada gera resultados satisfatórios no seguimento da trajetória, em que é observada semelhança entre o caminho percorrido e o desejado.

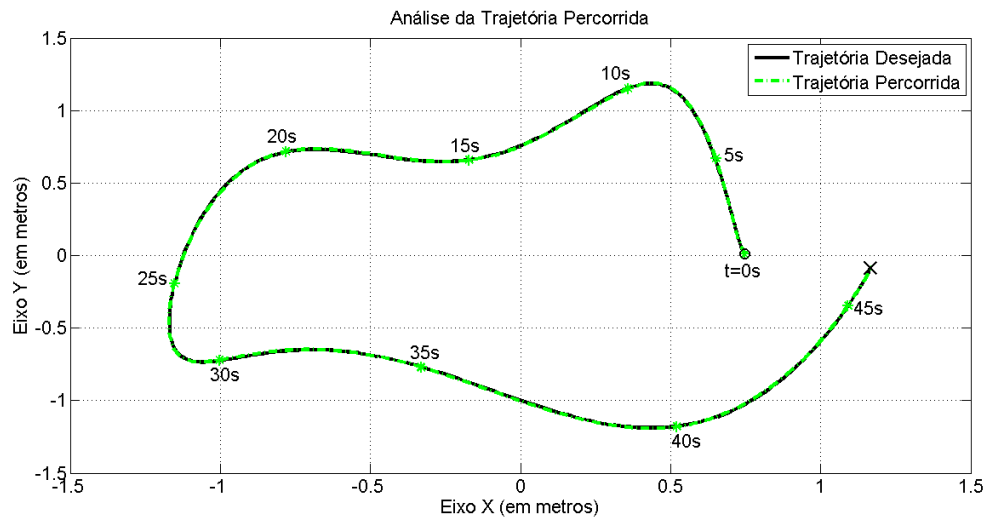


Figura 3.6: Resultado para análise da trajetória percorrida com controlador DFL.

Na Figura 3.7 vê-se o esforço de controle durante o percurso da mesma trajetória da Figura 3.6.

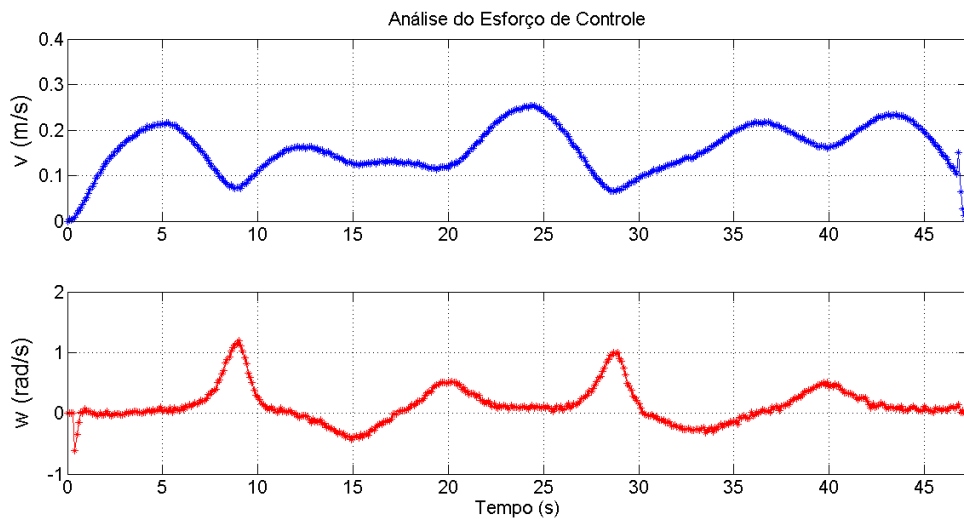


Figura 3.7: Resultados para análise do esforço de controle no percurso da trajetória com controlador DFL.

Com relação ao esforço de controle, observa-se que não há alterações bruscas de velocidade linear, além de que a média fica em torno de 30% da máxima. Já para a velocidade angular, em alguns instantes observam-se variações acentuadas, no entanto essas acelerações angulares aplicadas no robô, mesmo altas, respeitam as restrições dos motores, ficando sempre abaixo do valor máximo de $5,0030 \text{ rad/s}^2$.

3.2.3 Controlador NTI

Analogamente ao que já foi mostrado para o caso do controlador DFL, o controle NTI apresenta dois parâmetros para serem escolhidos, que são b e ζ , cujos valores estão evidenciados na Tabela 3.3 a seguir, em que foram obtidos os resultados visualmente satisfatórios referente ao erro de seguimento de trajetória e ao esforço de controle, como será observado posteriormente nos resultados.

Tabela 3.3: Parâmetros escolhidos para o controlador NTI.

Parâmetro	Valor Escolhido
b	40
ζ	1

De forma similar ao que foi mostrado na Seção 3.2.2, são realizadas simulações, sendo adotadas as mesmas intensidades de perturbação, ou seja, um ruído máximo de 10% dos valores de rotação aplicados nos motores do robô.

A Figura 3.8 ilustra a simulação para análise do percurso da trajetória, representada na Equação (3.3), pelo robô Lego EV3.

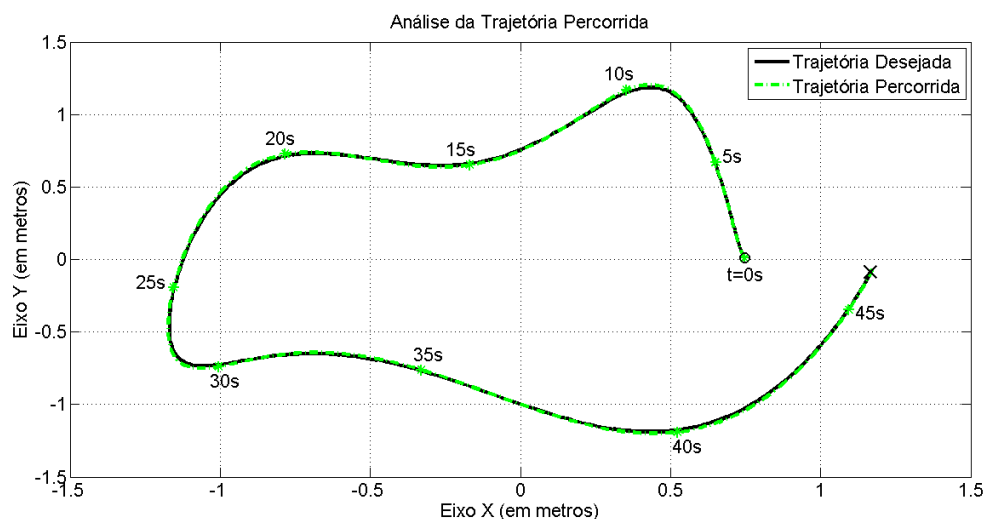


Figura 3.8: Resultado para análise da trajetória percorrida com controlador NTI.

Na Figura 3.9 vê-se o esforço de controle para a mesma trajetória da Figura 3.8.

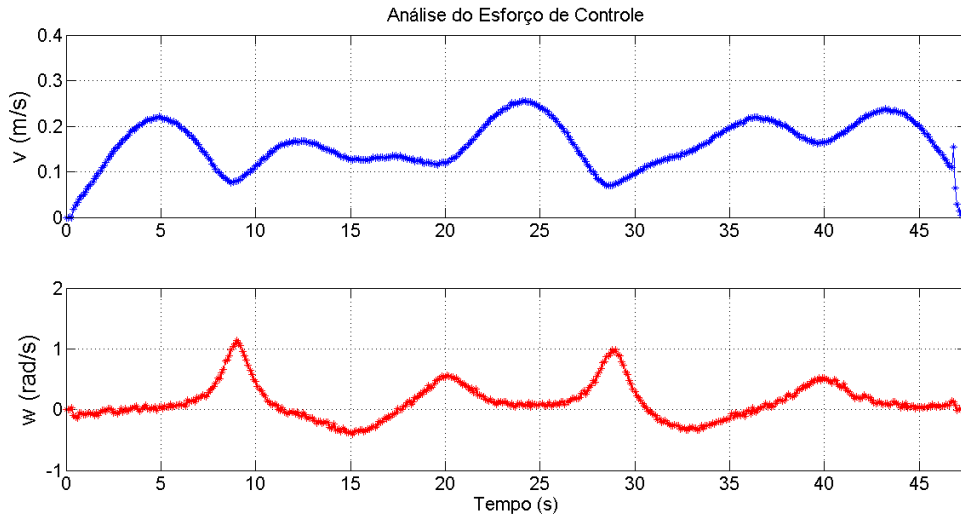


Figura 3.9: Resultados para análise do esforço de controle no percurso da trajetória com controlador NTI.

Nota-se que, também para o controlador NTI, a escolha adotada obteve resultados satisfatórios no seguimento da trajetória e no esforço de controle com o robô Lego EV3. Adicionalmente, observa-se resultados semelhantes no seguimento de trajetória e no esforço de controle, se comparado com o controlador DFL.

A seguir, serão abordados os aspectos referentes à implementação do planejamento, que irão construir as trajetórias nas quais o sistema de controle implementado irá seguir de forma a evitar as colisões com os obstáculos do meio.

3.3 Implementação do Planejamento

Como visto na Seção 2.4, o planejamento da trajetória consiste em desenvolver um algoritmo que, a partir das informações de posicionamento dos obstáculos no ambiente, é capaz de determinar os pontos (x_d, y_d) para que o robô atinja uma dada referência desejada, sem que haja colisões com nenhum obstáculo.

Todo o planejamento é feito antes que o robô realize o próximo movimento. Portanto, toma-se como base a posição inicial do robô num dado tempo discreto, que será a posição inicial da trajetória a ser planejada. A cada novo obstáculo que é

detectado em um determinado raio de detecção¹, um novo planejamento deve ser executado, em que são fornecidos novos dados de entrada do robô e do ambiente para o algoritmo de planejamento. Sendo assim, de forma geral, tem-se, vide Figura 3.10, o algoritmo para planejamento de trajetória e controle cinemático na medida em que o robô se desloca pelo ambiente.

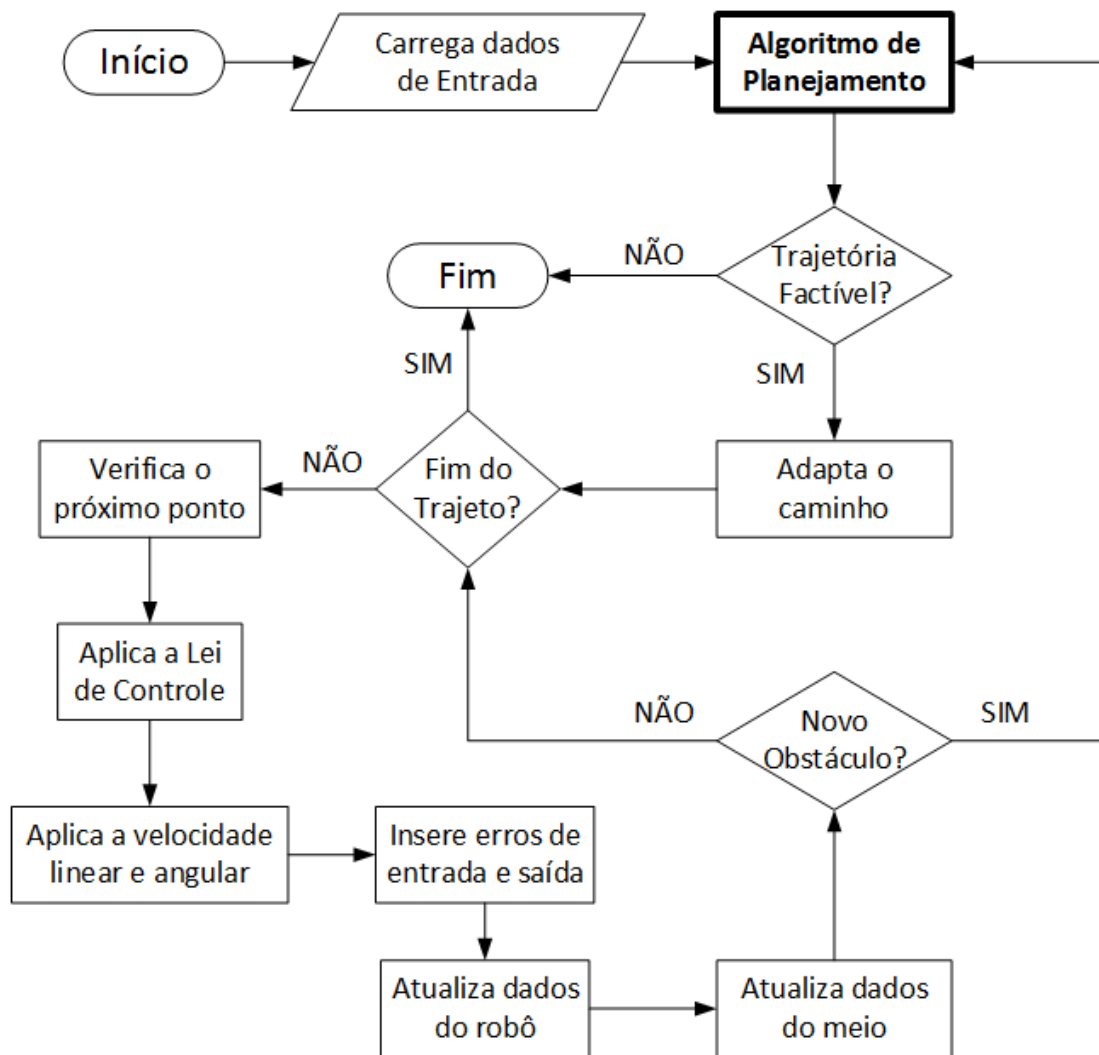


Figura 3.10: Algoritmo geral para o planejamento de trajetória e controle cinemático.

A seguir será elucidado como é desenvolvido o algoritmo para o método APF. No entanto, como já visto na Seção 2.4.3 esse método, em determinadas situações, não converge para o ponto final desejado, por conta do problema dos mínimos locais.

¹Vale destacar que o mapeamento é simulado e não através de dados reais de sensores.

Dessa forma, para contornar esse problema é também desenvolvido uma modificação no algoritmo do APF clássico, que será visto posteriormente.

3.3.1 Algoritmo do Método APF

Para desenvolver o algoritmo do método APF são necessários, basicamente, três passos:

- carregar dados do ambiente;
- gerar pontos da trajetória e
- testar se o caminho planejado é factível.

Primeiramente, são carregadas as informações do ambiente, ou seja, são detectados os obstáculos em um raio de detecção de $2,0m$ da posição atual, simulando o sistema de sensoriamento. Dessa forma, tem-se as coordenadas no sistema plano para as coordenadas (x_{b_i}, y_{b_i}) das bordas dos obstáculos detectados e, com isso, podem ser calculadas as mínimas distâncias da borda de cada obstáculo para a borda do robô, de acordo com o que foi visto na Equação (2.20).

Após o mapeamento do ambiente e cálculo das distâncias d_{b_i} , pode-se gerar os pontos da trajetória que consiga desviar dos obstáculos detectados. Nesta etapa, deve-se fazer uma rotina para serem obtidos os pontos da trajetória. Para isso, são calculados, a cada iteração, os potenciais de atração U_{att} e de repulsão U_{rep} , de acordo com as Equações (2.17), (2.18) e (2.19) já elucidadas. Portanto, a cada iteração pode ser calculada a próxima posição $(x_d(k+1), y_d(k+1))$ da trajetória, de acordo com as equações:

$$\begin{aligned} x_d(k+1) &= (\eta \cdot \frac{t_s}{100} \cdot K - \beta \cdot \frac{t_s}{100} + 1) \cdot x_d(k) + \beta \cdot \frac{t_s}{100} \cdot x_{ref} - \eta \cdot \frac{t_s}{100} \cdot K_x \\ y_d(k+1) &= (\eta \cdot \frac{t_s}{100} \cdot K - \beta \cdot \frac{t_s}{100} + 1) \cdot y_d(k) + \beta \cdot \frac{t_s}{100} \cdot y_{ref} - \eta \cdot \frac{t_s}{100} \cdot K_y \end{aligned} \quad (3.4)$$

sendo:

$$\begin{aligned} K &= \sum_{i=1}^j \left(\frac{1}{D_{max}} - \frac{1}{d_{b_i}} \right)^2 \\ K_x &= \sum_{i=1}^j \left(x_{b_i} \cdot \left(\frac{1}{D_{max}} - \frac{1}{d_{b_i}} \right) \right)^2, \quad \forall d_{b_i} < D_{max} \\ K_y &= \sum_{i=1}^j \left(y_{b_i} \cdot \left(\frac{1}{D_{max}} - \frac{1}{d_{b_i}} \right) \right)^2 \end{aligned} \quad (3.5)$$

Adota-se que o valor de D_{max} é igual a $36cm$, em função das dimensões do robô diferencial Lego EV3. Os parâmetros de atração β e repulsão η são previamente escolhidos para serem utilizados na Equação (3.4). Não há uma maneira formal de determinar a melhor escolha para os parâmetros β e η no método APF, apenas se sabe que a atração deve ser maior que a repulsão [5]. Neste trabalho, foram feitos testes em simulações para fazer a escolha dos valores dos parâmetros. Após os testes, foram escolhidos os seguintes valores para β e η , evidenciados na Tabela 3.4 a seguir, que gerou resultados satisfatórios no planejamento de trajetórias.

Tabela 3.4: Escolha dos parâmetros de atração e repulsão do método APF.

Parâmetro	Valor do Parâmetro
β	10
η	0,5

A rotina só é finalizada quando se obtém um ponto suficientemente próximo, a uma distância de no máximo $L/2$ da posição na qual se quer chegar, sendo L a distância entre os eixos das rodas do robô. Outro critério de parada para a rotina de planejamento ocorre quando se atinge um ponto de mínimo local da função potencial, como já elucidado na Seção 2.4.3. Pode-se saber se foi atingido um ponto de mínimo local, quando não ocorrem significativas variações para as próximas posições planejadas no trajeto, mesmo após muitas iterações para obtenção dos novos pontos (x_d, y_d) .

A Figura 3.11 sintetiza de forma simplificada o algoritmo utilizado para o planejamento da trajetória através do método APF.

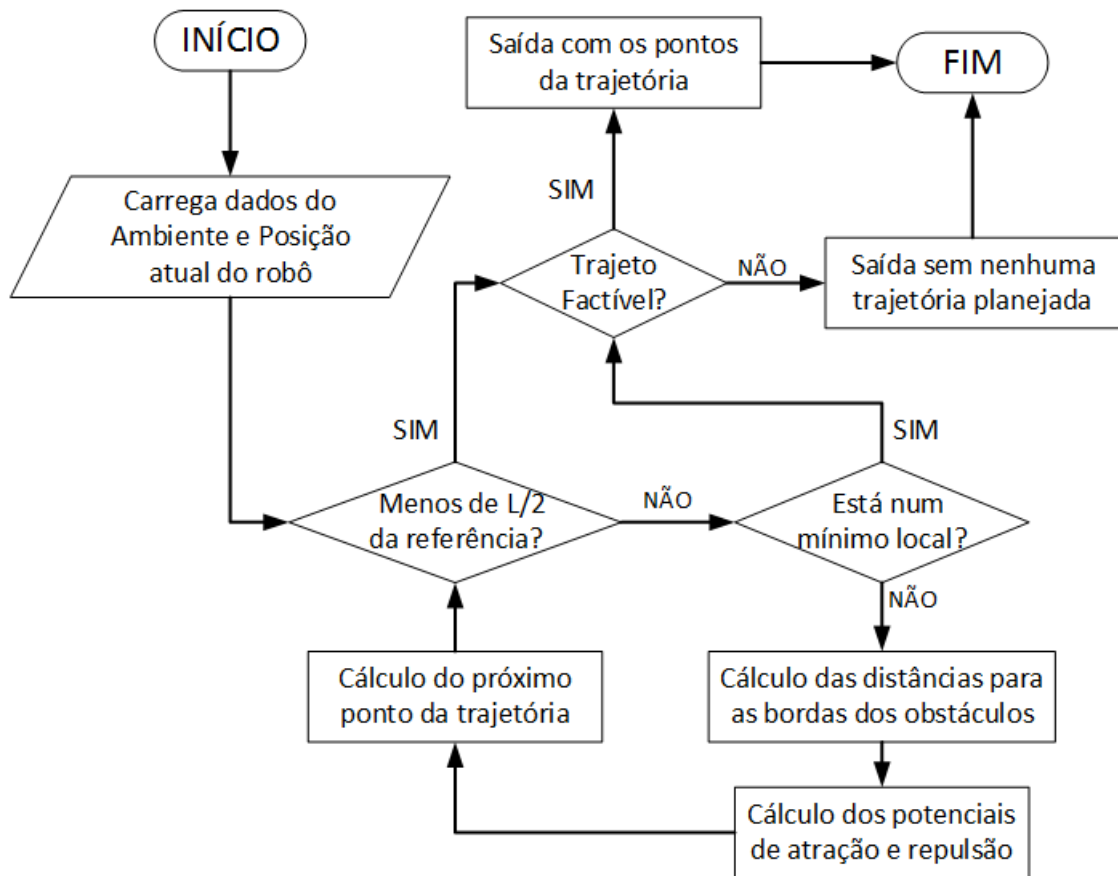


Figura 3.11: Algoritmo para o planejamento de trajetória.

3.3.2 Modificação do Método APF

Como explicado na Seção 2.4.3, no planejamento com o método APF nem sempre será atingido o ponto de referência, mesmo que se tenha uma trajetória factível, visto que há a possibilidade do planejamento chegar em um ponto de mínimo local da função potencial, a depender das configurações iniciais escolhidas para os parâmetros de atração à referência e repulsão dos obstáculos, do ponto inicial do veículo e da disposição dos obstáculos no meio. Dessa forma, para superar esse problema e garantir que haja convergência para o ponto de referência desejado sempre que houver uma trajetória factível, são feitas algumas modificações no algoritmo original do método. Esse método modificado será chamado de *APF-2*.

Após a obtenção dos dados na etapa de detecção dos obstáculos², são calculados

²Essa obtenção é feita apenas em simulação, pois não há nenhum sistema de sensoriamento real.

todos os n pontos (x_c, y_c) pertencentes a uma circunferência \mathcal{C}_k de raio $d_r = L$, cujo centro está na posição atual da trajetória. O próximo ponto da trajetória planejada será um dos pontos (x_c, y_c) que apresenta o menor valor da função potencial. Portanto, para cada nova iteração é obtido um novo ponto para a trajetória planejada, que corresponde a um ponto em uma distância d_r da posição anterior e que apresenta o menor valor de potencial comparado aos n pontos (x_c, y_c) pertencentes à circunferência \mathcal{C}_k .

A Figura 3.12 mostra graficamente como ocorre cada iteração para obter os pontos da trajetória no decorrer do planejamento com o método APF-2.

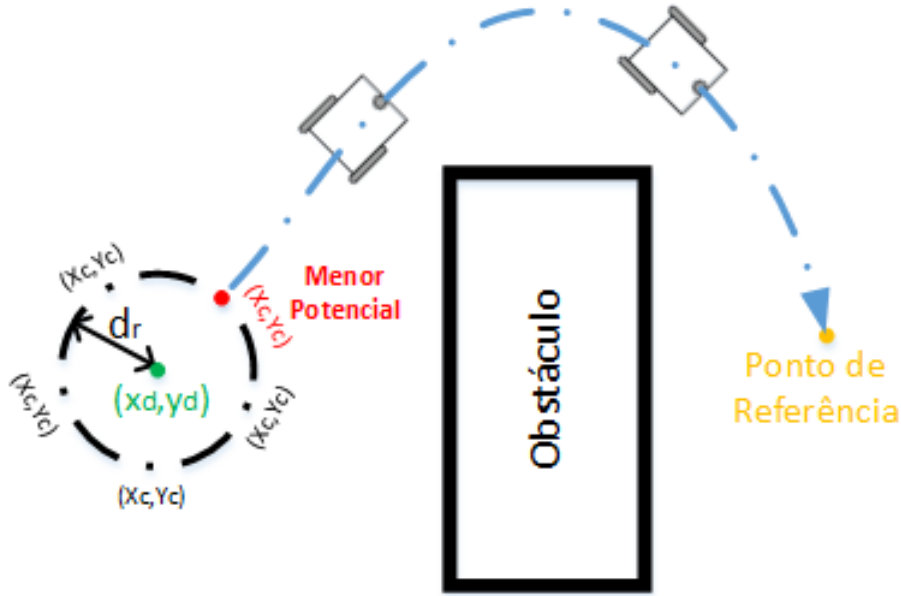


Figura 3.12: Esquema gráfico do planejamento APF-2 em cada iteração.

Os potenciais de atração P_{att} e repulsão P_{rep} a cada iteração e em cada ponto (x_c, y_c) , no método APF-2, são calculados de acordo com a equação:

$$P_{att} = \frac{q_{ref}}{(x_c - x_{ref})^2 + (y_c - y_{ref})^2}$$

$$P_{rep} = \sum_{i=1}^j \left(\frac{q_b}{(x_c - x_{b_i})^2 + (y_c - y_{b_i})^2} \right), \quad (3.6)$$

sendo q_{ref} a carga do ponto de referência e q_b a carga associada ao obstáculo i na posição (x_{b_i}, y_{b_i}) .

O próximo ponto da trajetória deverá ser aquele em que o potencial resultante

ψ , calculado de acordo com a Equação (3.7), que relaciona o potencial de repulsão com o de atração, seja o menor entre todos os pontos de \mathcal{C}_k :

$$\psi = P_{rep} - P_{att}. \quad (3.7)$$

Visando convergir a trajetória planejada para um ponto final suficientemente próximo do ponto de referência desejado, são feitas variações incrementais dos parâmetros q_b e q_{ref} no decorrer do algoritmo de planejamento.

O valor q_b , da carga de cada objeto i , é decrementado na medida em que o ponto planejado da trajetória se aproxima do ponto de referência, para que sejam diminuídos os efeitos de repulsão quando o veículo estiver próximo do ponto de referência, facilitando a convergência para a posição final desejada.

Caso a trajetória planejada não atinja o ponto de referência desejado, ou seja, caso ocorra um mínimo local, incrementa-se o valor de q_{ref} , reinicia-se o valor de q_b , volta ao ponto inicial e realiza-se nova tentativa de convergência para levar a trajetória até o ponto de referência, visto que na tentativa anterior não houve suficiente potencial de atração. Assim, o valor de q_{ref} é incrementado até que se atinga o ponto final desejado, superando a possibilidade de ocorrência do problema dos mínimos locais, observado no método APF clássico.

Portanto, a principal modificação no método consiste em alterações incrementais e heurísticas nos parâmetros das funções potenciais no decorrer do planejamento, visando sempre convergir a trajetória para o ponto final desejado e eliminar a ocorrência do problema de mínimos locais.

A Figura 3.13 ilustra o fluxograma simplificado para o algoritmo do método APF-2, relativo à modificação do método APF clássico representado anteriormente no fluxograma da Figura 3.11.

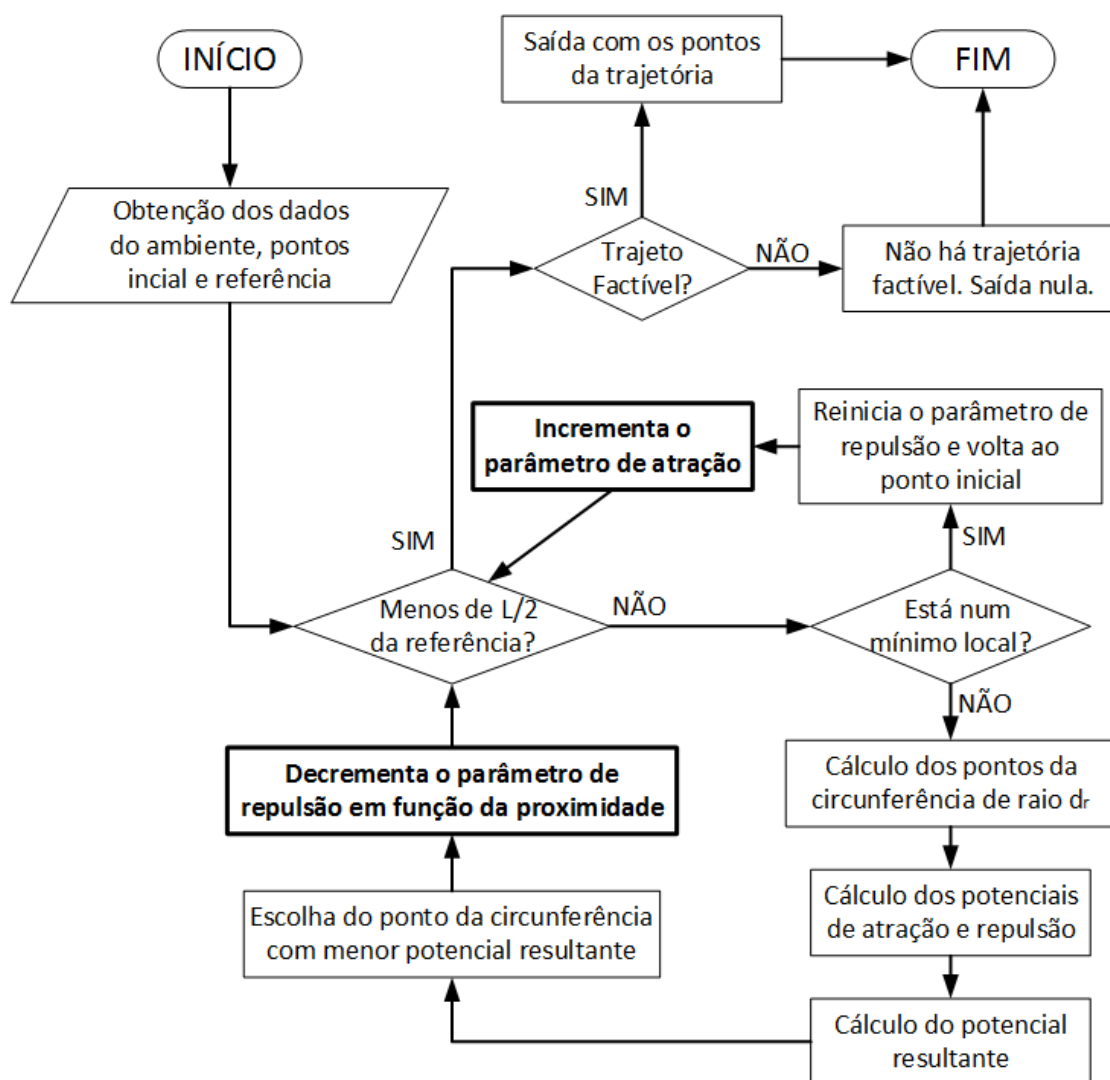


Figura 3.13: Fluxograma simplificado do método APF-2 de planejamento de trajetória.

Destacam-se também outros métodos heurísticos de planejamento, que não apresentam o problema dos mínimos locais, como observado em [1, 2, 5]. No entanto, o método APF é o que possui a característica de planejar trajetórias que mais se aproximam do caminho mais seguro, que leva o robô até a referência evitando ao máximo os obstáculos [9]. Levando-se em conta essa vantagem do APF, a modificação proposta, além de contornar o problema dos mínimos locais, visa planejar trajetórias similares ao APF, como será visto nos resultados que serão elucidados no Capítulo a seguir.

3.4 Conclusão

Neste capítulo, foram apresentados os aspectos relativos ao desenvolvimento propriamente dito para o sistema de controle e planejamento de trajetórias.

Primeiramente, foi mostrada a modelagem do sistema físico, que neste trabalho corresponde ao veículo representado pelo robô diferencial Lego EV3, usado como base de testes. O modelo dinâmico depende do sistema para acionamento dos motores das rodas do veículo, enquanto que o modelo cinemático depende apenas da geometria do veículo, levando em consideração as equações apresentadas na Seção 2.3.

Em seguida, foram mostradas as estratégias de controle cinemático para seguir uma determinada trajetória. Os efeitos dinâmicos não são considerados na implementação do controle para seguimento de trajetória, visto que a dinâmica para variação da posição é muito mais lenta que na variação da velocidade do veículo. A partir de um algoritmo genérico do sistema de controle, foram implementadas estratégias utilizando as leis características dos controladores DFL e NTI, visando comparar os resultados de ambos. Dessa forma, observou-se que os resultados para o controle DFL e NTI foram significativamente semelhantes, tanto em precisão no seguimento da trajetória quanto no esforço de controle exigido pelos motores do robô.

Por fim, foi evidenciada a implementação do sistema de planejamento de trajetórias para desvio de obstáculos, observando o algoritmo para o método APF, em que, a depender das configurações iniciais referentes aos valores escolhidos para os parâmetros de atração e repulsão da função potencial, poderá ocorrer o problema dos mínimos locais, fazendo com que a trajetória planejada não atinga o ponto final desejado.

Visando superar a limitação decorrente do problema dos mínimos locais, é desenvolvida uma modificação do método APF clássico. Nessa modificação, chamada de método APF-2, são desenvolvidas estratégias heurísticas com variações incrementais nos parâmetros de atração e repulsão das funções potenciais para que os potenciais se ajustem dinamicamente, no decorrer do próprio planejamento, de maneira a evitar ocorrência de mínimos locais, garantindo a convergência para o ponto final desejado através de uma trajetória baseada em campos, que naturalmente planeja um cami-

nho seguro, no sentido de levar o veículo até o ponto de referência, afastando-se dos obstáculos detectados para evitar as colisões.

Sendo assim, visando mostrar a eficácia do sistema desenvolvido, no próximo Capítulo serão observados os resultados referentes ao sistema de planejamento e controle implementado, em que serão colocadas diferentes situações e configurações iniciais usando o robô Lego EV3 como modelo. Também serão elucidados resultados experimentais obtidos a partir de testes realizados em laboratório.

Capítulo 4

RESULTADOS

Neste Capítulo, serão mostrados os resultados de simulação para o sistema de navegação, com planejamento de trajetória e controle cinemático, aplicado no modelo do robô diferencial Lego EV3. Para o planejamento são feitas as simulações para diferentes ambientes e em condições iniciais diversas, usando tanto o método APF clássico quanto a nova abordagem proposta, que visa evitar a ocorrência do problema de mínimos locais. No controle serão observados os resultados de simulação para o seguimento da trajetória planejada, com vistas à precisão obtida pelo controlador, bem como os sinais de controle aplicados para as estratégias DFL e NTI utilizadas. É feita uma análise dos resultados simulados, de forma a comparar as estratégias de controle e os métodos de planejamento, concluindo se nestes casos foi atingido o objetivo proposto para o sistema: chegar em um ponto de referência e desviar dos obstáculos detectados que estão dispostos no meio.

Além das simulações do sistema de navegação, serão realizados experimentos em laboratório usando o robô diferencial Lego EV3. Em tais experimentos é observado o resultado do caminho real que foi percorrido pelo robô em comparação com o trajeto que foi planejado previamente a partir de um ambiente totalmente conhecido.

4.1 Simulações do Sistema

Todas as simulações para os testes do sistema de navegação foram feitas usando o *software* MATLAB®¹, em que são programadas rotinas próprias, sem a necessidade da utilização de *toolboxes* para robótica ou controle. Foram usadas quatro diferentes configurações de ambiente para simular o sistema de navegação desenvolvido. Em todas essas configurações utiliza-se um campo retangular de dimensões $2,5m \times 2,5m$ ¹.

O *Ambiente 1* apresenta três obstáculos dispostos de acordo com o que está evidenciado na Figura 4.1.

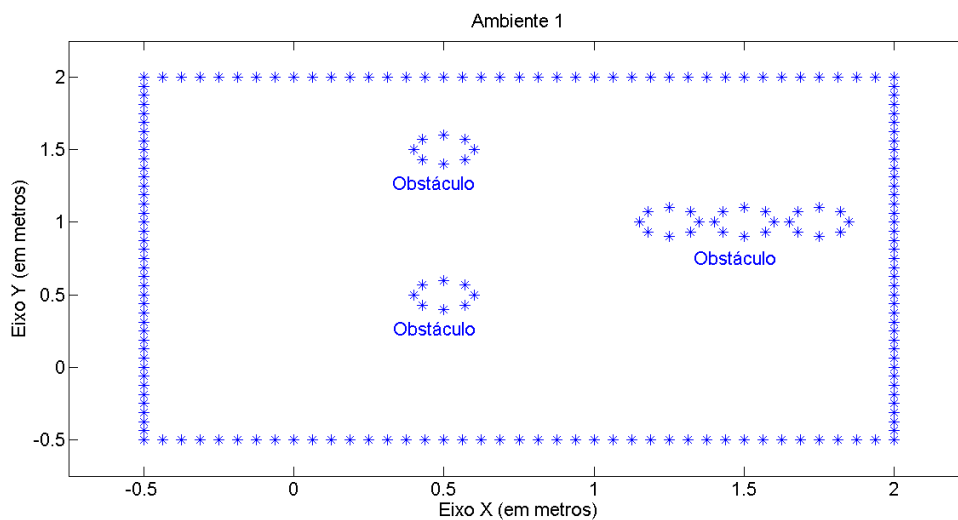


Figura 4.1: Ambiente 1.

Para este primeiro ambiente é escolhido o ponto (x_{ref}, y_{ref}) sendo $(1,5; 1,5)$ e são feitos testes para três diferentes posições iniciais do robô: em $(-0,25; -0,25)$, em $(-0,25; 1,75)$ e em $(1,75; -0,25)$.

O *Ambiente 2* apresenta um obstáculo côncavo no meio do campo, como observado na Figura 4.2, visando ilustrar o caso em que há o problema dos mínimos locais com o método APF clássico, devido a esse tipo específico de obstáculo.

¹Essas são as dimensões do campo disponível no laboratório de robótica

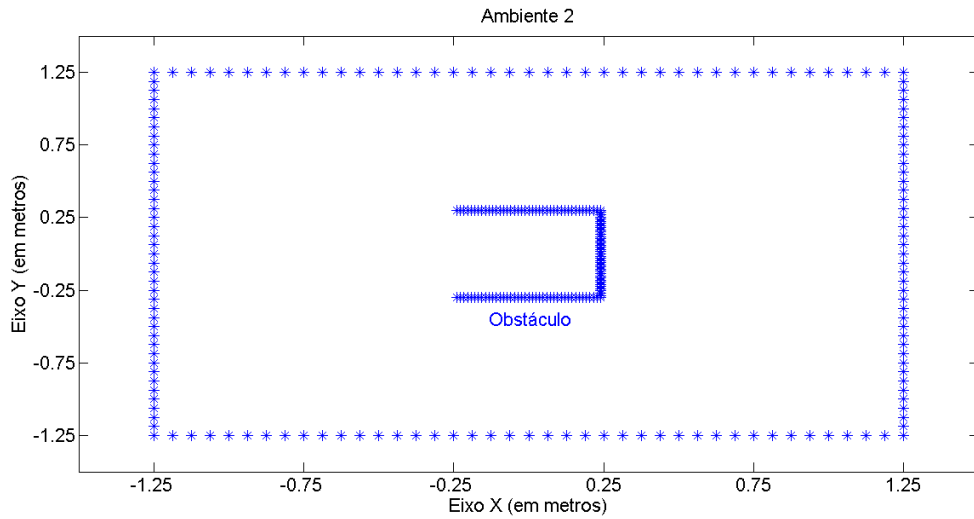


Figura 4.2: Ambiente 2.

Neste segundo ambiente, usa-se o ponto de referência em $(0, 75; 0)$ e duas diferentes posições iniciais: em $(-1; -1)$ e em $(-1; 0)$.

Para dificultar o planejamento, coloca-se mais um obstáculo côncavo para o *Ambiente 3*, como pode-se observar na Figura 4.3.

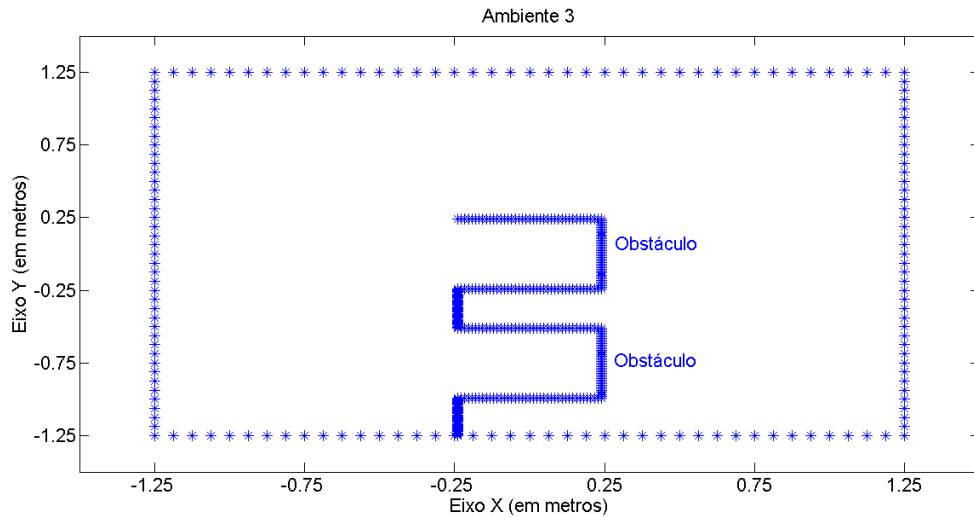


Figura 4.3: Ambiente 3.

No *Ambiente 3*, escolhe-se o mesmo ponto de referência $(0, 75; 0)$ do *Ambiente 2* e as mesmas duas posições iniciais em $(-1; 0)$ e em $(-1; -1)$.

Por fim, para ilustrar um caso em que não há trajetória factível, usa-se o *Ambiente 4*, ilustrado na Figura 4.4.

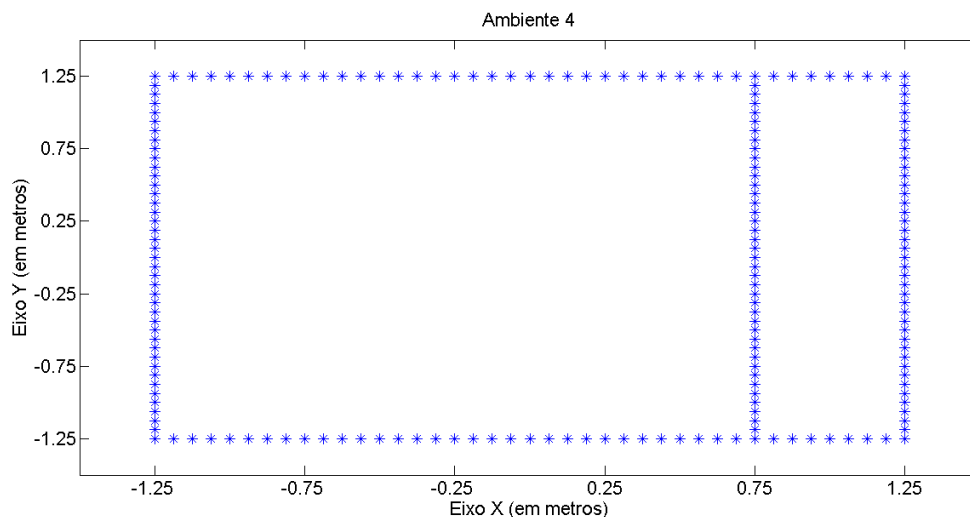


Figura 4.4: Ambiente 4.

Para esse último ambiente, o ponto inicial é em $(-1; -1)$ e o ponto final é em $(1; 1)$, de modo que não haverá nenhum caminho que ligue esses pontos sem que o robô passe por algum obstáculo do ambiente, neste caso uma parede fechando a passagem. Portanto, não há trajetória factível para o *Ambiente 4*.

Como já discutido na Seção 3.3, para o mapeamento do ambiente não é desenvolvido um processamento do sinal a partir de sensores reais, como os de ultra-som ou infravermelho. Ao invés disso, é feita uma simulação do mapeamento durante o movimento do robô, sendo detectado novos obstáculos localizados a uma distância menor ou igual a $2,0m$, que é o raio de detecção². Portanto, é feita uma simulação do mapeamento que detecta obstáculos a distâncias de no máximo $2,0m$ da posição atualizada do robô em cada instante.

As simulações mostram os resultados para o funcionamento do sistema de navegação usando o modelo do robô diferencial Lego EV3, destacando a trajetória planejada e percorrida para os quatro diferentes ambientes e posições iniciais escolhidas, utilizando os dois métodos de planejamento (APF e APF-2) e as duas diferentes estratégias de controle cinemático (DFL e NTI). Para uma dessas tra-

²Esse é um valor escolhido com base nas distâncias médias que podem ser detectadas por sensores de ultra-som ou infravermelho

jetórias, são mostrados também os sinais de controle (velocidade linear e angular) aplicadas no robô durante o percurso, a fim de verificar o esforço de controle.

A Tabela 4.1 mostra as inicializações para as 16 diferentes simulações que são executadas nos 4 ambientes distintos usando o robô Lego EV3 como modelo, de modo a testar o sistema de navegação.

Tabela 4.1: Simulações executadas para o sistema de navegação desenvolvido.

	Ambiente	Posição Inicial	Método de Planejamento
Simulação 1	<i>Ambiente 1</i>	$(-0, 25; -0, 25)$	APF
Simulação 2	<i>Ambiente 1</i>	$(-0, 25; -0, 25)$	APF-2
Simulação 3	<i>Ambiente 1</i>	$(-0, 25; 1, 75)$	APF
Simulação 4	<i>Ambiente 1</i>	$(-0, 25; 1, 75)$	APF-2
Simulação 5	<i>Ambiente 1</i>	$(1, 75; -0, 25)$	APF
Simulação 6	<i>Ambiente 1</i>	$(1, 75; -0, 25)$	APF-2
Simulação 7	<i>Ambiente 2</i>	$(-1; -1)$	APF
Simulação 8	<i>Ambiente 2</i>	$(-1; -1)$	APF-2
Simulação 9	<i>Ambiente 2</i>	$(-1; 0)$	APF
Simulação 10	<i>Ambiente 2</i>	$(-1; 0)$	APF-2
Simulação 11	<i>Ambiente 3</i>	$(-1; 0)$	APF
Simulação 12	<i>Ambiente 3</i>	$(-1; 0)$	APF-2
Simulação 13	<i>Ambiente 3</i>	$(-1; -1)$	APF
Simulação 14	<i>Ambiente 3</i>	$(-1; -1)$	APF-2
Simulação 15	<i>Ambiente 4</i>	$(-1; -1)$	APF
Simulação 16	<i>Ambiente 4</i>	$(-1; -1)$	APF-2

4.1.1 Resultados da Trajetória Planejada e Percorrida

Nesta seção, serão mostrados apenas os resultados dos métodos de planejamento de trajetória e controle cinemático do robô para os diferentes casos mostrados na Tabela 4.1, ilustrando a trajetória percorrida e planejada. Posteriormente, na Seção 4.1.4, será feita uma análise referente a todos esses resultados.

Nas Figuras 4.5 a 4.10 são mostrados os resultados de simulação, destacando a análise da trajetória planejada e percorrida no *Ambiente 1*.

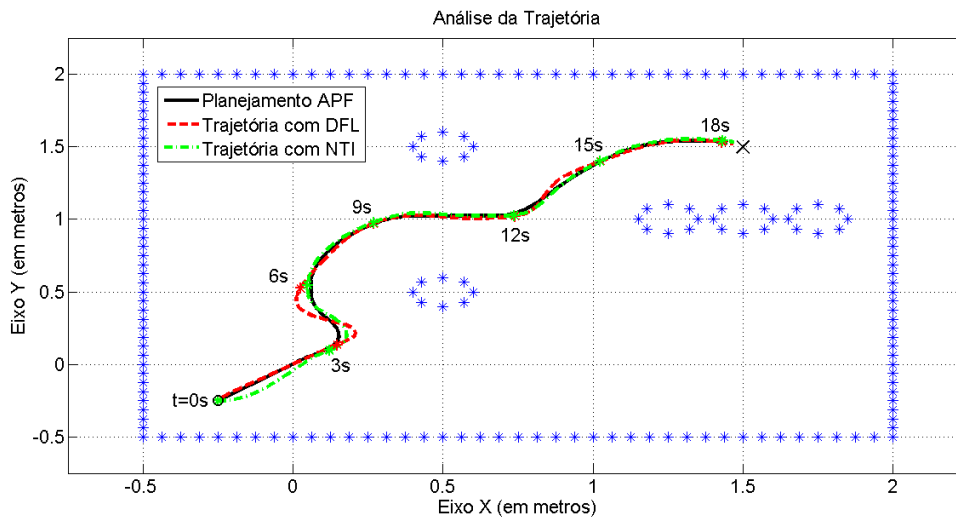


Figura 4.5: Simulação 1.

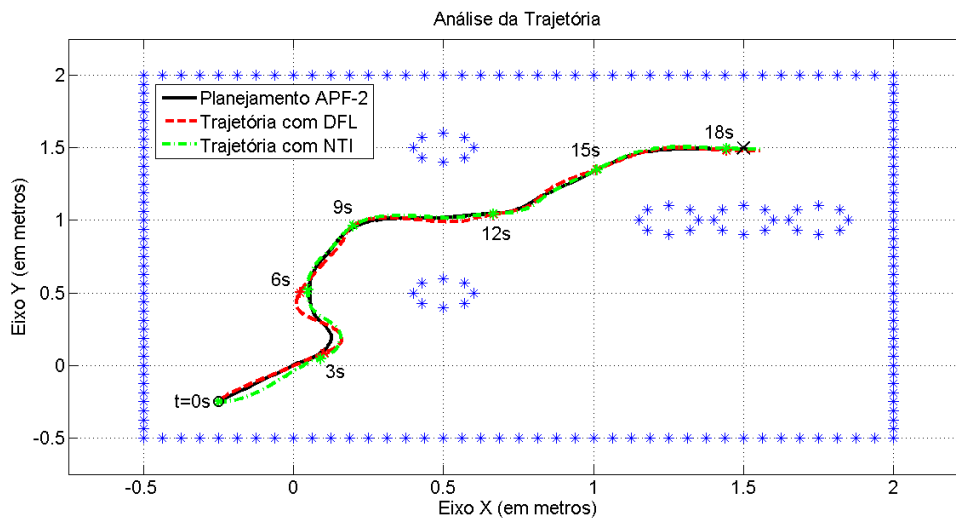


Figura 4.6: Simulação 2.

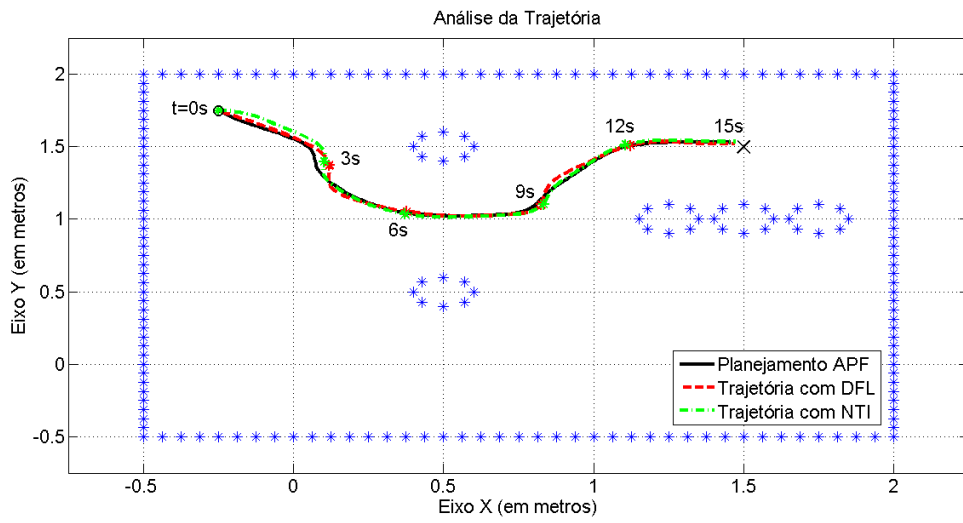


Figura 4.7: Simulação 3.

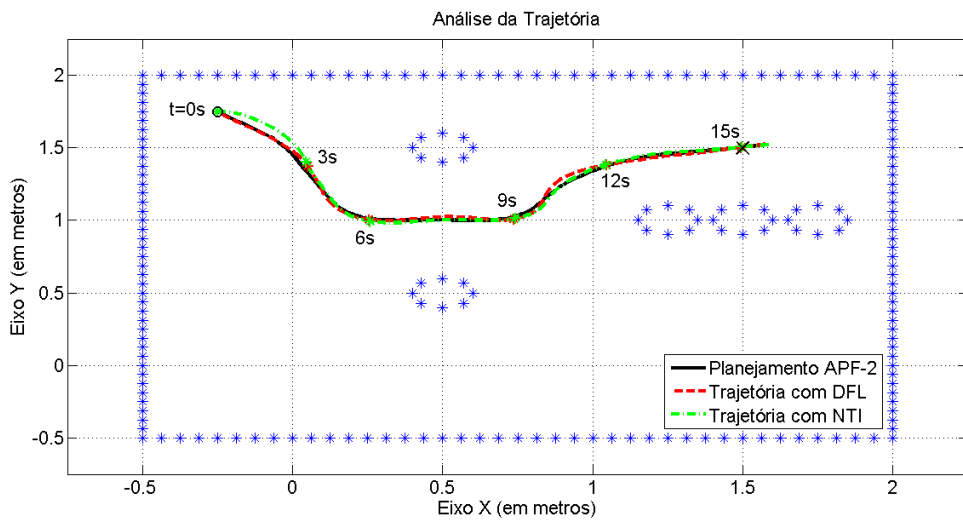


Figura 4.8: Simulação 4.

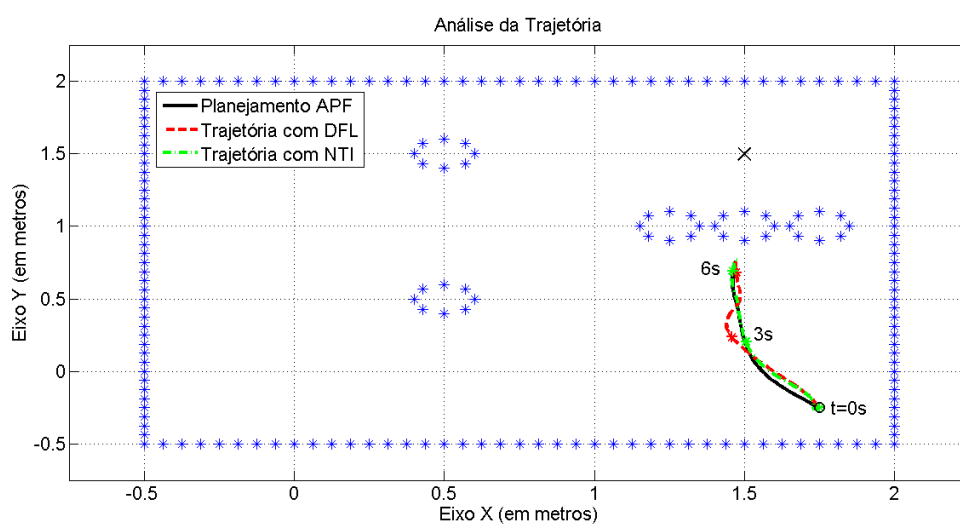


Figura 4.9: Simulação 5.

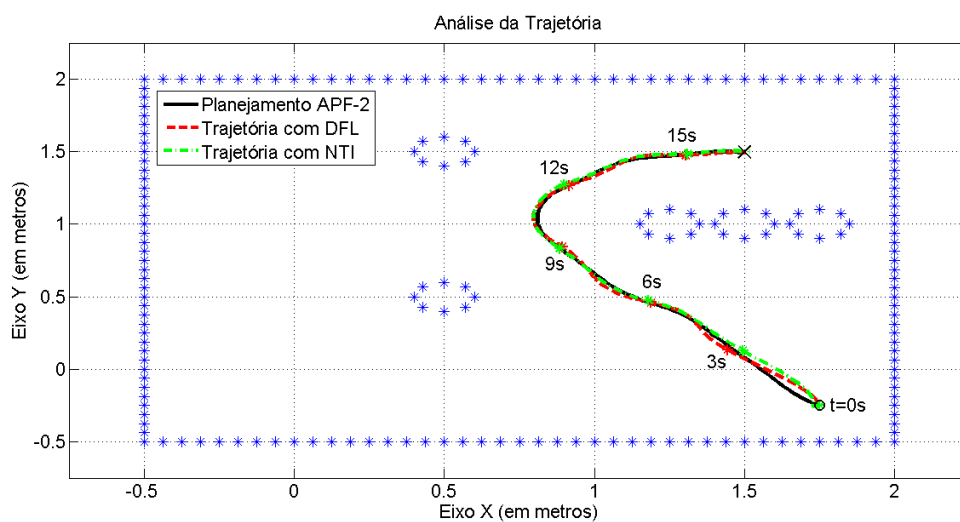


Figura 4.10: Simulação 6.

Nas Figuras 4.11 a 4.14 são mostrados os resultados de simulação para as trajetórias no *Ambiente 2*.

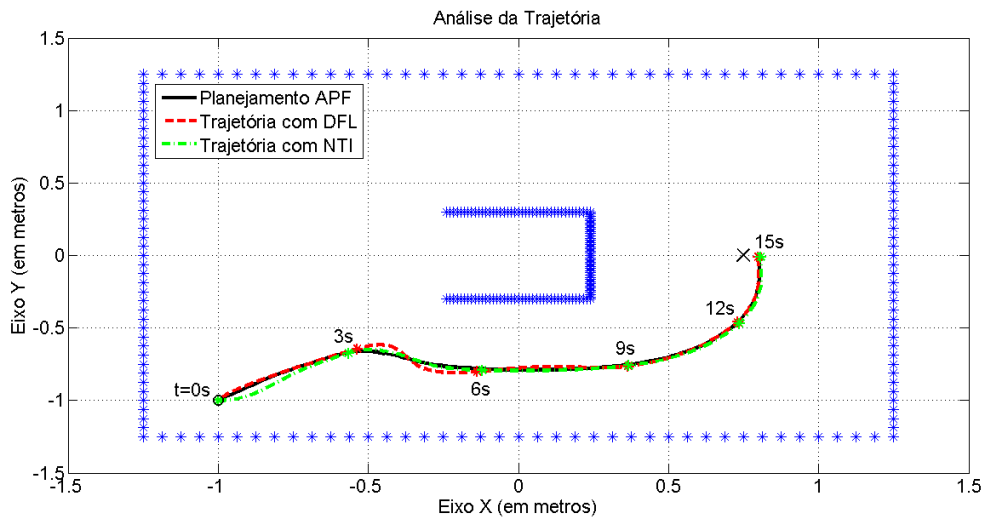


Figura 4.11: Simulação 7.

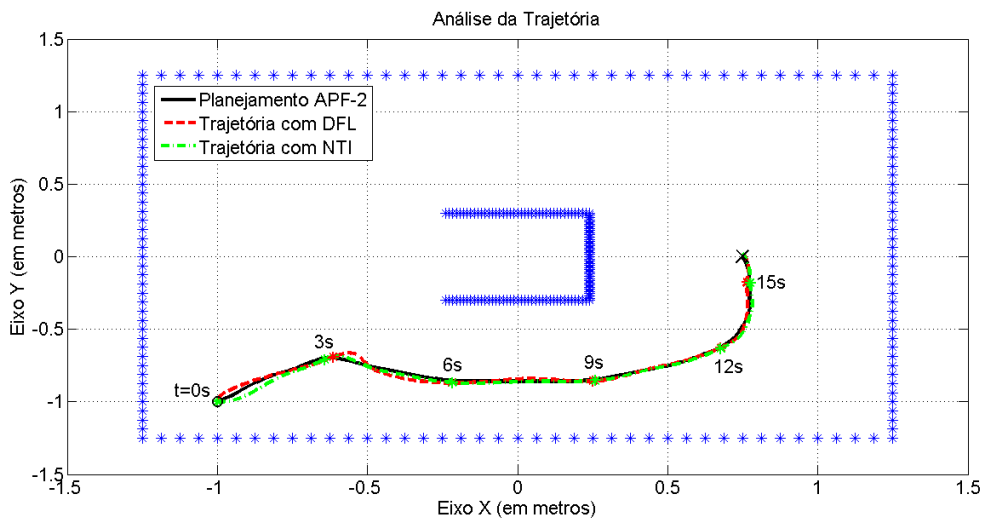


Figura 4.12: Simulação 8.

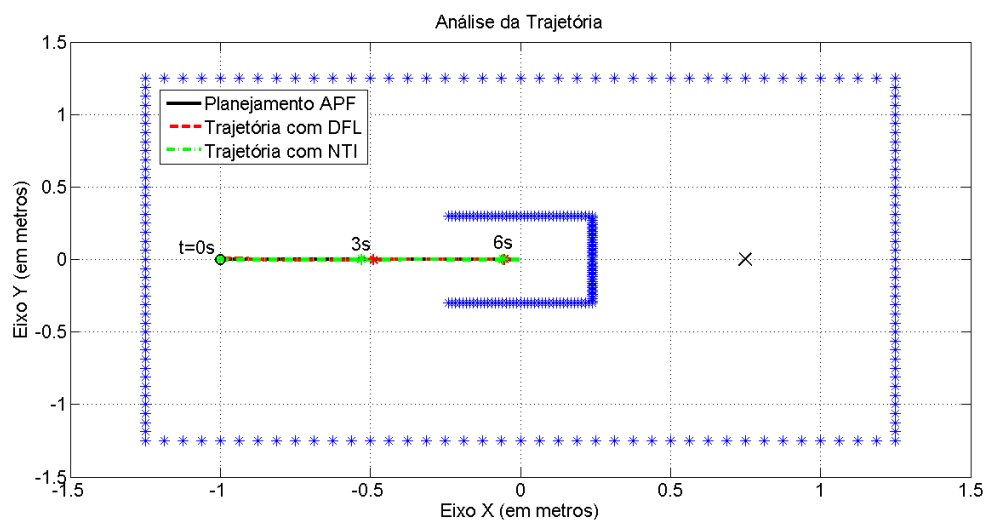


Figura 4.13: Simulação 9.

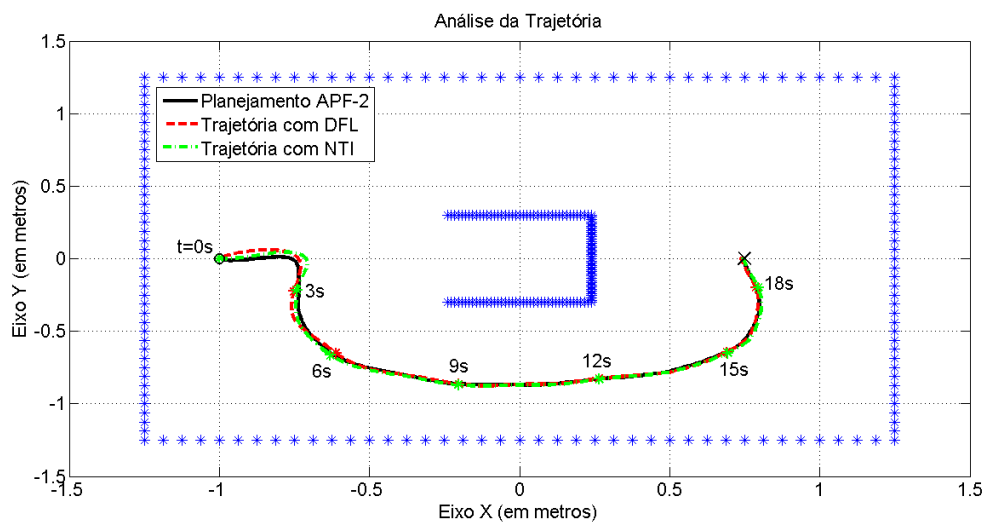


Figura 4.14: Simulação 10.

Nas Figuras 4.15 a 4.18 são mostrados os resultados de simulação para as trajetórias no *Ambiente 3*.

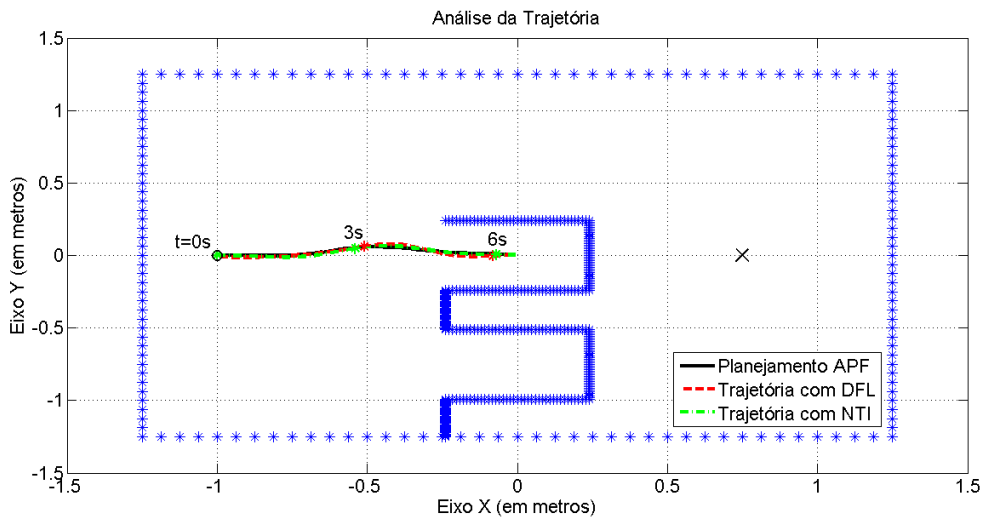


Figura 4.15: Simulação 11.

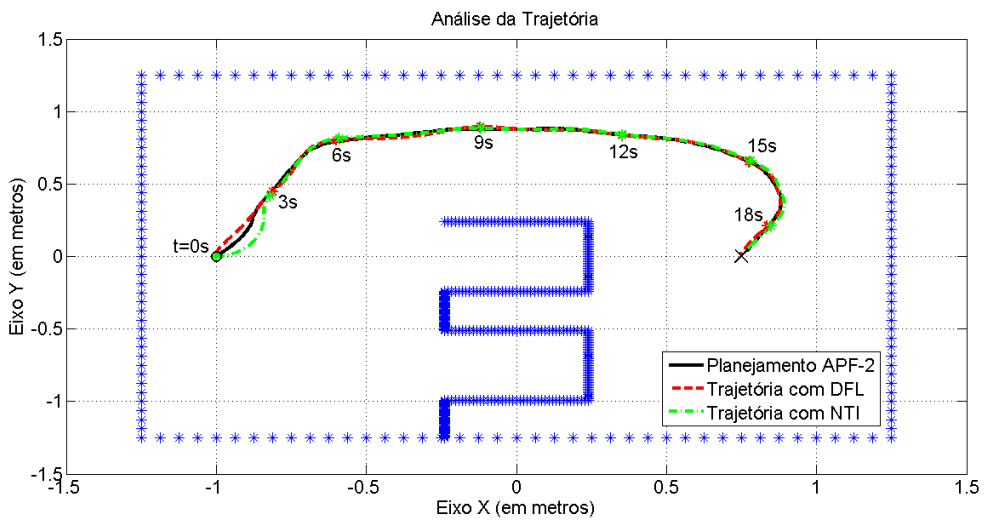


Figura 4.16: Simulação 12.

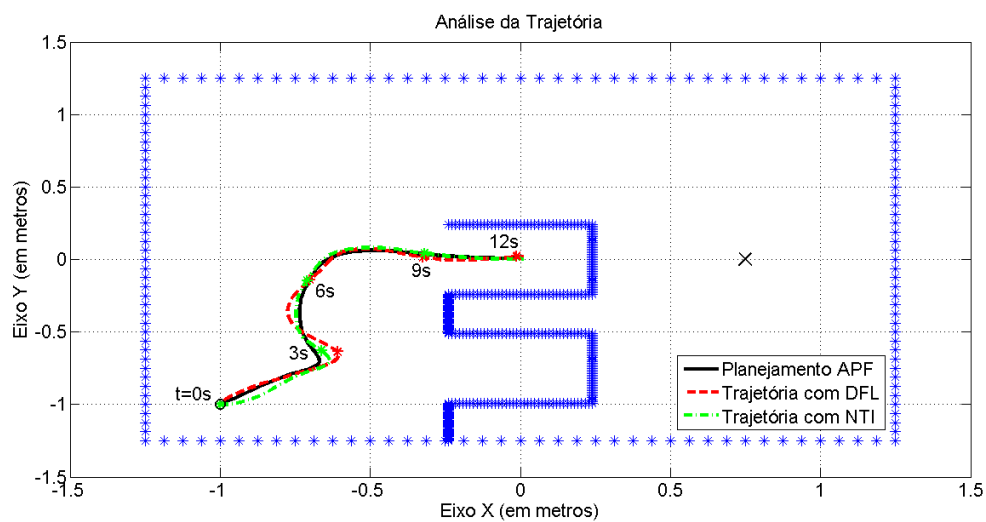


Figura 4.17: Simulação 13.

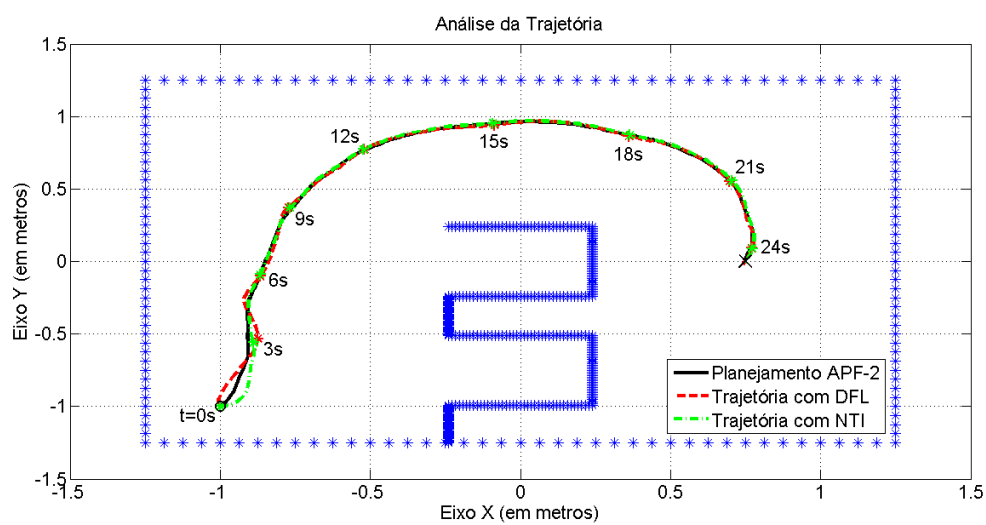


Figura 4.18: Simulação 14.

Nas Figuras 4.19 a 4.20 são mostrados os resultados de simulação para as trajetórias no *Ambiente 4*.

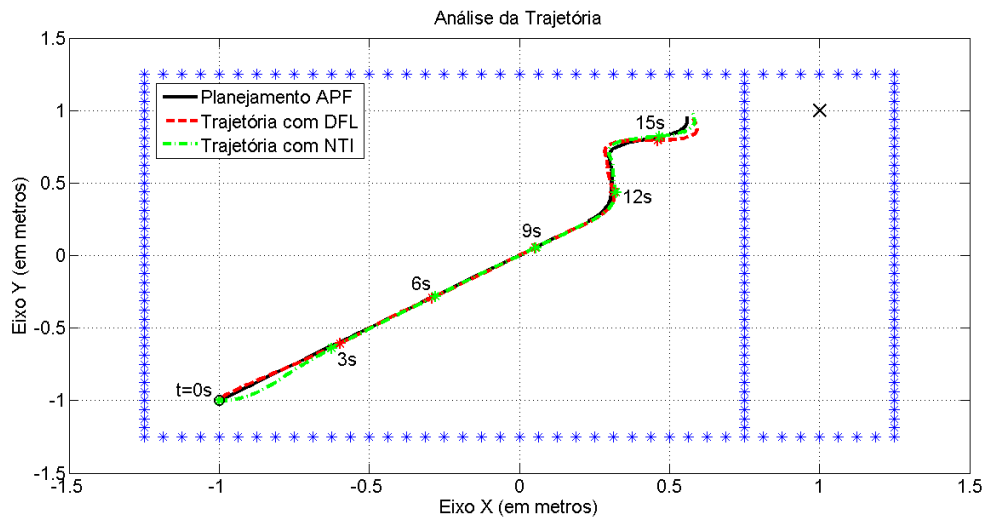


Figura 4.19: Simulação 15.

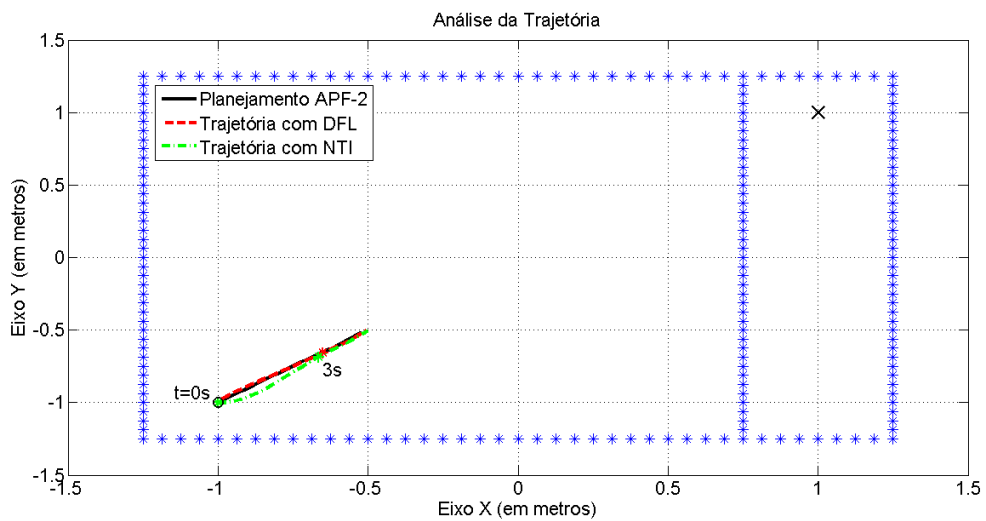


Figura 4.20: Simulação 16.

4.1.2 Resultado do Esforço de Controle

Para ilustrar o esforço de controle, na Figura 4.21 tem-se a variação no tempo das velocidades linear e angular, que são os sinais de controle aplicados no robô, durante o caminho percorrido na *Simulação 2*, usando as estratégias DFL e NTI. As demais simulações apresentam resultados que levam a conclusões semelhantes referentes ao esforço de controle, que serão discutidas na Seção 4.1.4 a seguir.

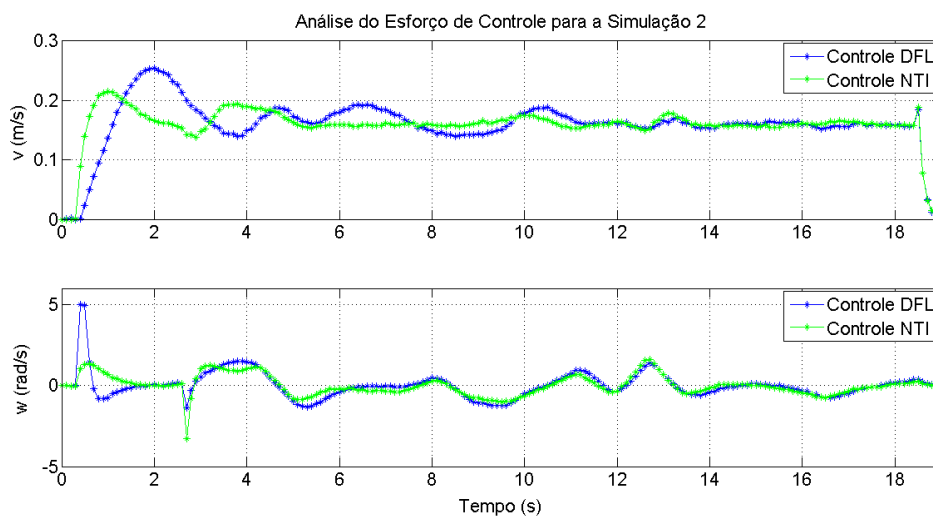


Figura 4.21: Sinais de Controle aplicados na *Simulação 2*, com o APF-2.

4.1.3 Variação dos Parâmetros no Método APF-2

É importante observar também como ocorrem as variações dos parâmetros de atração e repulsão quando é utilizado o método APF-2. Como já explicado, o APF-2 aplica variações incrementais nos parâmetros durante o processo de planejamento da trajetória, visando sempre conseguir convergir para o ponto final desejado e desviar dos obstáculos detectados. Na Figura 4.22, tem-se um exemplo da variação dos parâmetros durante o planejamento da *Simulação 2*.

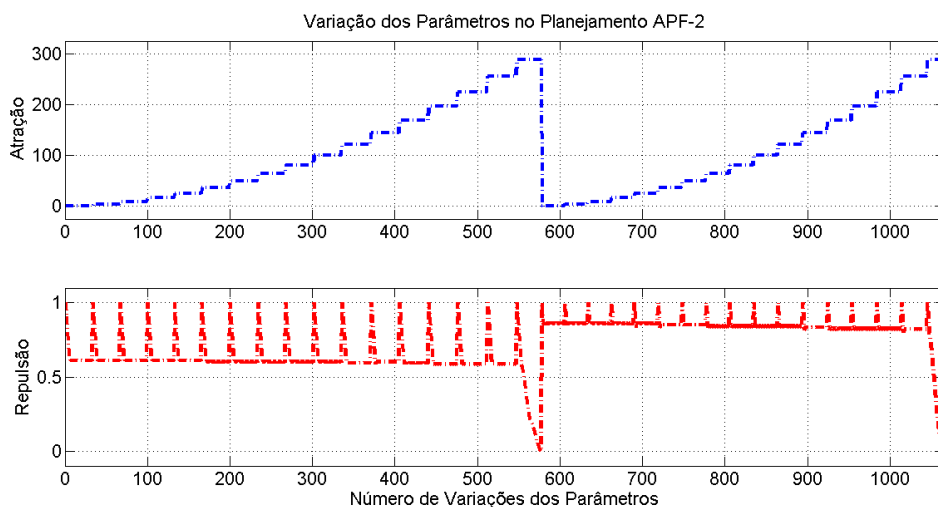


Figura 4.22: Variação dos Parâmetros de Planejamento durante a *Simulação 2*, com o APF-2.

Nas outras simulações, ocorreram variações que levam a conclusões semelhantes, que serão discutidas na Seção 4.1.4 a seguir.

4.1.4 Análise dos Resultados

Comparando-se os resultados ilustrados nas Figuras 4.5 e 4.6, nas Figuras 4.7 e 4.8 ou nas Figuras 4.11 e 4.12, observa-se que, em todos esses casos, as trajetórias planejadas com o método APF clássico e com o método APF-2 foram aproximadamente as mesmas. As trajetórias com o método APF clássico apresentam a característica de evitar colisões através do afastamento dos obstáculos detectados, tendendo a convergir para um ponto de mínimo potencial³, o que constitui uma trajetória segura em face ao objetivo do sistema de desviar dos obstáculos. Portanto, pode-se dizer que o método APF-2 também gerou trajetórias seguras na relação desvio e convergência à posição desejada, já que foram similares às trajetórias obtidas pelo método clássico quando não houve o problema dos mínimos locais.

No entanto, comparando-se os resultados mostrados nas Figuras 4.9 e 4.10, nas Figuras 4.13 e 4.14, nas Figuras 4.15 e 4.16 ou nas Figuras 4.17 e 4.18, observa-se que nestes casos ocorreram o problema dos mínimos locais no planejamento usando o método APF clássico, que fez com que o caminho planejado não levasse o robô para o ponto final desejado, parando em um ponto de mínimo local da função potencial. Já com o método APF-2, planejaram-se trajetórias que atingiram o ponto final desejado, superando o problema dos mínimos locais.

No caso em que não há trajetória factível, vide Figuras 4.19 e 4.20, tanto o método APF quanto o método APF-2 não chegaram ao ponto desejado, visto que não havia trajetória factível até o ponto final desejado. Neste caso, o método clássico interrompe o movimento novamente em um ponto de mínimo local da função potencial. Já o método APF-2 interrompe previamente o movimento, pois já verifica a impossibilidade de atingir o ponto final desejado. Portanto, o método APF-2 é mais rápido em verificar quando não há trajetória factível.

Com relação ao controle cinemático para o seguimento da trajetória planejada, observa-se em todas as simulações que o robô Lego EV3 percorre um caminho bem próximo ao planejado, o que atesta um sistema de controle com funcionamento

³O ponto de referência será inerentemente um ponto de mínimo potencial.

satisfatório. Comparando-se as estratégias DFL e NTI, pode-se dizer que com ambos os controladores têm-se resultados muito similares relativos à precisão no seguimento da trajetória planejada. Além disso, pode-se observar na Figura 4.21 que o esforço de controle é considerado pequeno durante o percurso do caminho pelo robô, ou seja, considera-se que o sistema de controle (DFL ou NTI) não exige muito do sistema de motorização, já que visualmente não ocorre um excesso de variações bruscas na velocidade linear ou angular.

Por fim, pode ser feita uma análise quanto a variação dos parâmetros de planejamento com o APF-2, vide Figura 4.22, em que observa-se um aumento quadrático no valor da atração. Cada novo valor na atração implica em uma tentativa de planejamento que não obteve sucesso, bem como reinicia o valor de repulsão. Além disso, o parâmetro de atração reinicia o seu valor, voltando para 1, quando é detectado um novo obstáculo que faz a trajetória anteriormente planejada ser ineficaz, ou seja, todo o planejamento deve ser realizado novamente. Quanto ao parâmetro de repulsão, vê-se que na medida em que o planejamento atinge o ponto final desejado, a repulsão diminui até chegar a zero, visando facilitar a convergência.

4.2 Resultados Experimentais

Para obter resultados experimentais do sistema de navegação desenvolvido, utiliza-se o robô Lego EV3 no *Ambiente 1*, cuja foto real em laboratório pode ser observada na Figura 4.23.

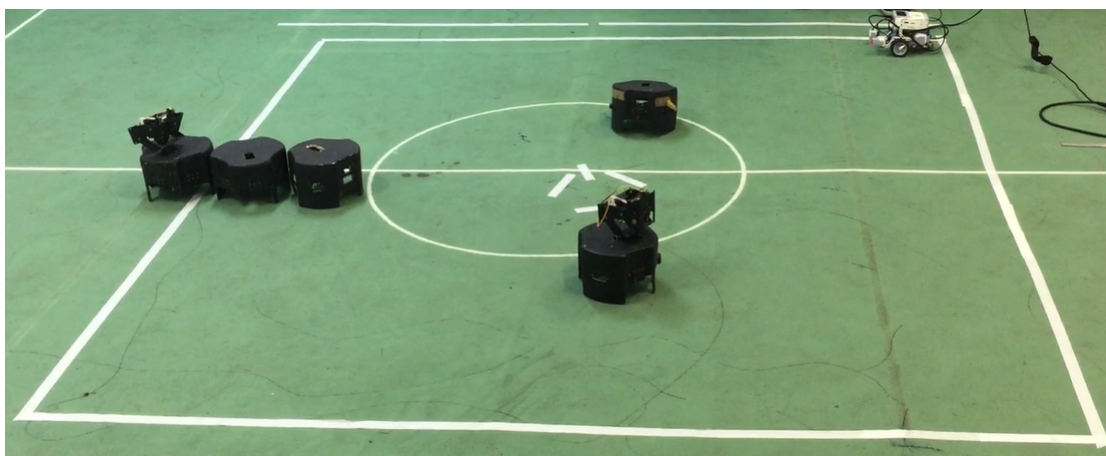


Figura 4.23: Ambiente 1 Real.

O MATLAB® tem suporte para controlar os movimentos e velocidade do robô Lego EV3 em tempo real, através da atribuição dos valores de rotação das rodas durante todo o caminho percorrido. Além disso, o *software* pode fornecer dados de leitura das velocidades de rotação das rodas a cada período de amostragem e durante todo o caminho percorrido [21].

Levando-se em conta essas facilidades de implementação, utiliza-se o MATLAB® como *software* para a realização dos experimentos em laboratório com o robô diferencial Lego EV3.

Nos experimentos são aplicados os sinais de velocidade linear e angular no robô, via *software*, de forma análoga ao que foi feito na obtenção dos resultados de simulação, visando atingir o ponto de referência e desviar dos obstáculos. No entanto, com relação à etapa de mapeamento, é importante saber que não se utiliza sensores ultra-som, infravermelho ou outro tipo de sensoriamento, sendo então necessário que todo o ambiente seja previamente conhecido nos experimentos, ou seja, tem-se todas as coordenadas dos obstáculos antes de implementar o sistema de planejamento e controle proposto.

São realizados quatro diferentes experimentos, com as configurações que estão evidenciadas na Tabela 4.2, em que é utilizado apenas o planejamento APF-2.

Tabela 4.2: Experimentos realizados no *Ambiente 1 Real* com o robô Lego EV3.

—————	Ponto Inicial	Planejamento	Controle
Experimento 1	(-0, 25; -0, 25)	APF-2	DFL
Experimento 2	(-0, 25; -0, 25)	APF-2	NTI
Experimento 3	(1, 75; -0, 25)	APF-2	DFL
Experimento 4	(1, 75; -0, 25)	APF-2	NTI

Nesses dois experimentos são obtidos os resultados para a trajetória percorrida e os sinais de controle aplicados ao longo do caminho. Adicionalmente, são incorporados os dados de saída de simulações com mesmas configurações iniciais dos respectivos experimentos, a fim de realizar uma comparação entre os resultados experimentais e os simulados. Vale ressaltar que as simulações usadas em comparação com os experimentos foram obtidas também a partir de um ambiente totalmente conhecido, portanto os caminhos planejados não são idênticos aos que foram observados nas Figuras 4.6 e 4.10.

Na Figura 4.24, tem-se o resultado da trajetória percorrida pelo robô no *Experimento 1*. Já na Figura 4.25, observam-se os sinais de controle aplicados ao longo do percurso.

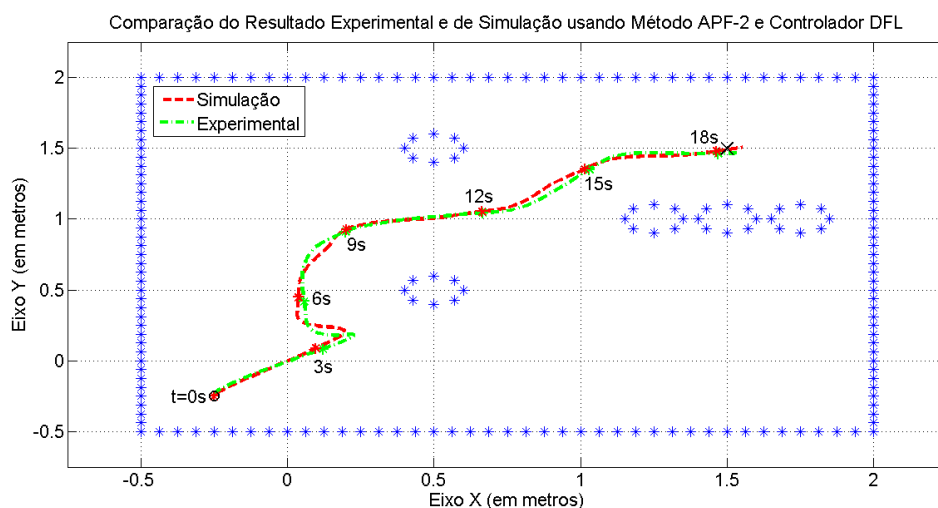


Figura 4.24: Resultado da trajetória percorrida no *Experimento 1*.

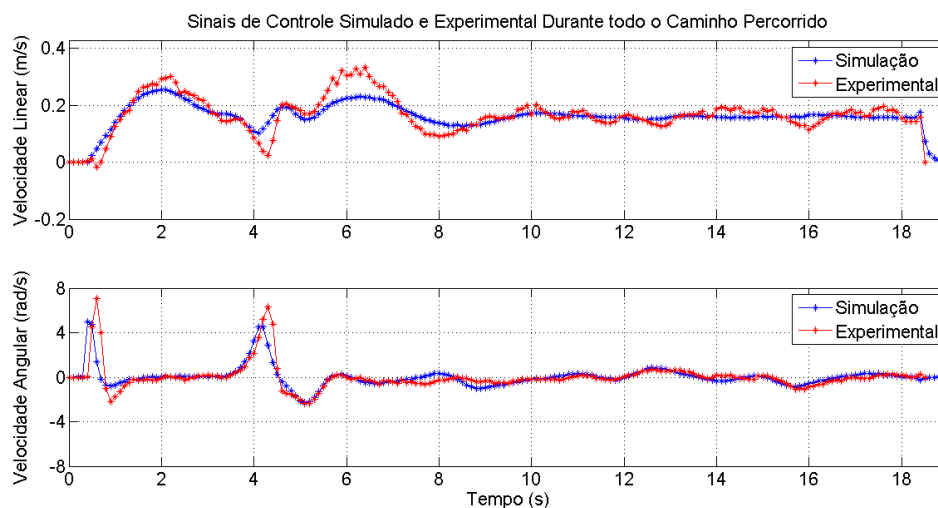


Figura 4.25: Sinais de controle aplicados no *Experimento 1*.

Para o *Experimento 2*, na Figura 4.26 tem-se o resultado da trajetória percorrida pelo robô e na Figura 4.27 observam-se os sinais de controle aplicados ao longo do caminho.

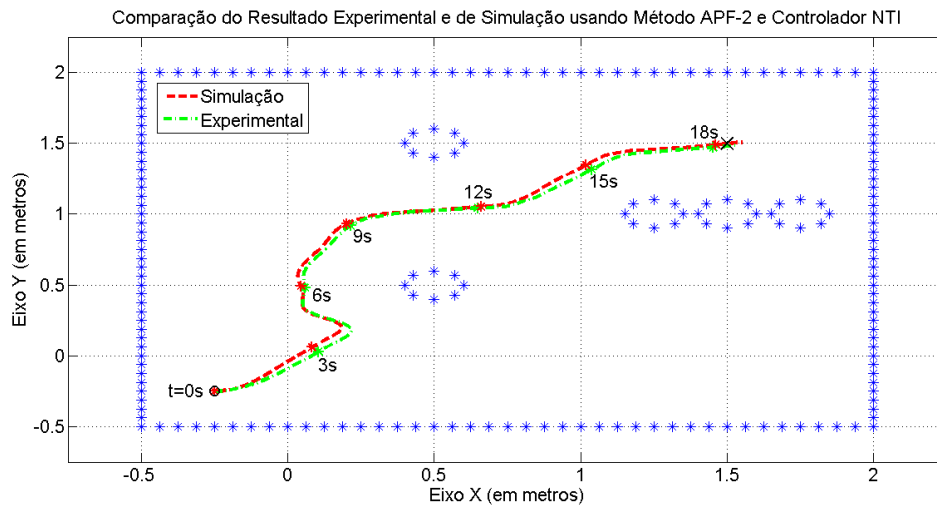


Figura 4.26: Resultado da trajetória percorrida no *Experimento 2*.

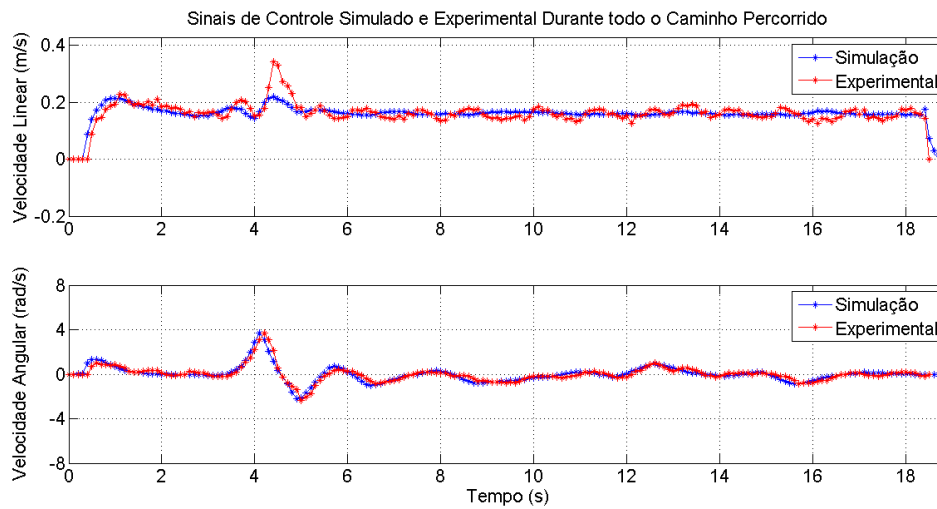


Figura 4.27: Sinais de controle aplicados no *Experimento 2*.

Para o *Experimento 3*, na Figura 4.28 tem-se o resultado da trajetória percorrida pelo robô e na Figura 4.29 observam-se os sinais de controle aplicados ao longo do caminho.

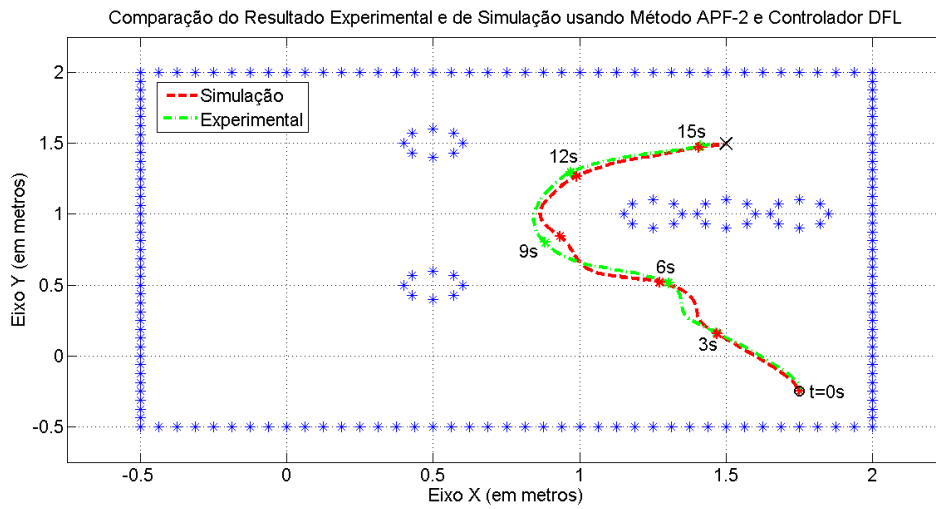


Figura 4.28: Resultado da trajetória percorrida no *Experimento 3*.

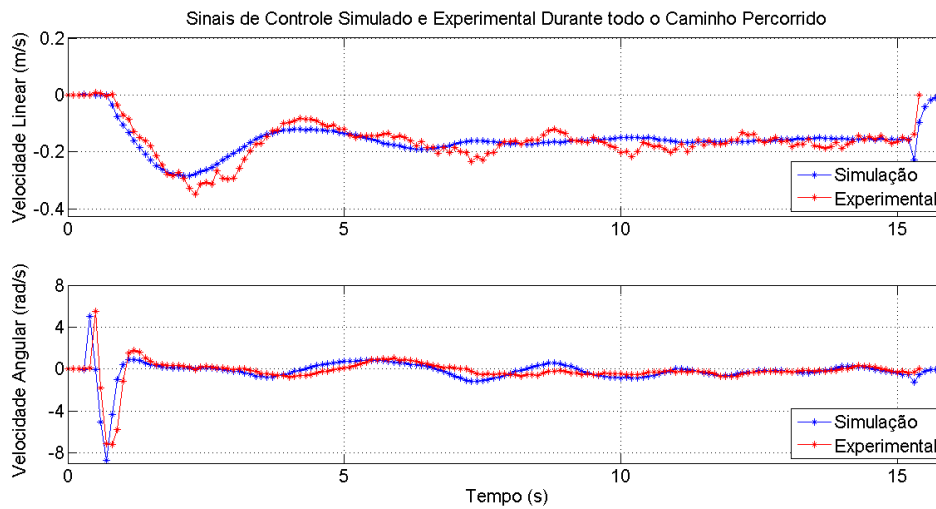


Figura 4.29: Sinais de controle aplicados no *Experimento 3*.

Por fim, para o *Experimento 4*, na Figura 4.30 tem-se o resultado da trajetória percorrida pelo robô e na Figura 4.31 observam-se os sinais de controle aplicados ao longo do caminho.

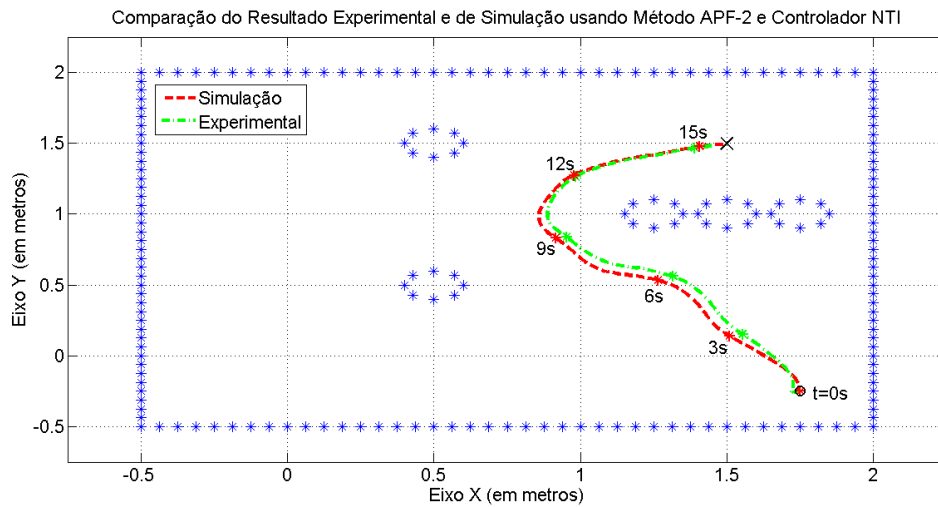


Figura 4.30: Resultado da trajetória percorrida no *Experimento 4*.

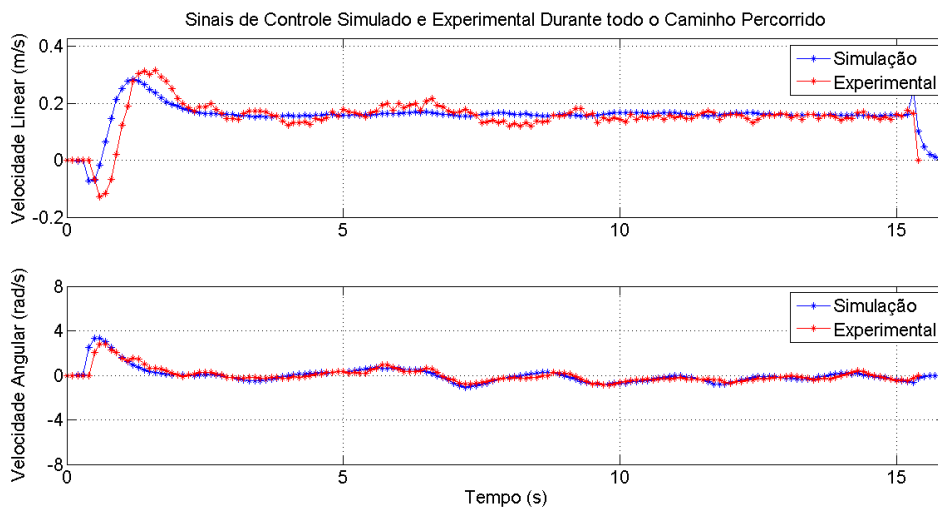


Figura 4.31: Sinais de controle aplicados no *Experimento 4*.

Nota-se que os resultados experimentais foram visualmente parecidos com os que foram obtidos em simulação, tanto na trajetória percorrida⁴ quanto nos sinais de controle aplicados no robô.

⁴A trajetória percorrida obtida no experimento é estimada com base nas velocidades medidas nas rodas do robô, levando-se em consideração que não há derrapagem durante o movimento.

4.3 Conclusão

Sendo assim, foi possível observar nas simulações e nos experimentos que o sistema de navegação implementado atende satisfatoriamente o seu propósito de atingir um ponto desejado e desviar dos obstáculos do ambiente.

Nas simulações e experimentos constata-se o que os controladores DFL e NTI têm visualmente resultados semelhantes quanto ao seguimento de trajetória e esforço de controle. Com relação à precisão para o seguimento de trajetória, nota-se que nas duas estratégias utilizadas o caminho planejado e o percorrido são visualmente semelhantes e suficientemente precisos para o objetivo desejado de desviar dos obstáculos e chegar ao ponto final. Já referente ao esforço de controle, ambas as estratégias apresentam variações de sinais de controle (velocidade linear e angular) que não são visualmente muito bruscas, ou seja, o esforço de controle é considerado satisfatório, não sendo exigido muito dos motores durante a etapa de acionamento dos mesmos.

Os métodos APF e APF-2 tem planejamentos semelhantes apenas quando não há mínimos locais na trajetória do método clássico, pois nos casos em que há o APF-2 consegue superar o problema e atingir o ponto final desejado sem colidir com os obstáculos. Portanto, pode-se constatar que o novo método proposto é mais eficaz que o método clássico.

Nos experimentos realizados, além de serem vistos resultados que ratificam a eficácia do sistema desenvolvido, observa-se também significativa semelhança com os resultados de simulação.

Capítulo 5

CONSIDERAÇÕES FINAIS

Este trabalho realizou o desenvolvimento de um sistema de navegação, com planejamento de trajetórias e controle cinemático, aplicado para veículos móveis terrestres de duas rodas com acionamento diferencial.

O planejamento consiste em construir trajetórias que atinjam um ponto de referência desejado e ao mesmo tempo desvie dos obstáculos dispostos e detectados no meio. Para isso, pode-se utilizar o método APF, que é baseado em campos potenciais artificiais. Este método consegue planejar trajetórias que tendem a convergir para um mínimo da função potencial, evitando os obstáculos detectados. Comparado a outros métodos de planejamento, como o Mapa de Rotas ou a Decomposição por células, o método APF planeja caminhos mais seguros, que evitam as colisões com os obstáculos detectados, convergindo até o ponto final desejado.

No entanto, observa-se uma limitação inerente ao método APF, conhecida com problema dos mínimos locais, que faz com que o planejamento convirja para um ponto de mínimo local da função potencial que nem sempre será o ponto de referência desejado para o final do trajeto.

Visando superar o problema dos mínimos locais, são desenvolvidas modificações no algoritmo do método APF clássico. Assim, desenvolve-se um novo método de planejamento, denominado APF-2, em que é desenvolvida uma estratégia que implementa variações incrementais e heurísticas nos parâmetros de atração e repulsão da função potencial. Portanto, são feitos ajustes nos parâmetros no decorrer do planejamento do percurso, de modo a evitar que se atinja um ponto de mínimo local que seja diferente do ponto final desejado. Esta é, então, a principal contribuição

deste trabalho: um melhoramento no algoritmo do método para planejamento de trajetórias baseado em campos potenciais artificiais.

Para a etapa de mapeamento, não há um sistema de sensoriamento real implementado, de forma que os resultados obtidos para o sistema de planejamento são vistos apenas em simulação. Nessas simulações, observa-se que o planejamento usando o método APF-2 supera o problema dos mínimos locais para situações em que ocorreria usando o método APF clássico. Devido a isso, constata-se sensível melhoramento para a nova proposta do sistema de planejamento. Adicionalmente, o método APF-2 desenvolvido apresenta resultados de planejamentos semelhantes aos do método APF clássico, quando não ocorre o problema dos mínimos locais, ou seja, continua planejando trajetórias seguras, no sentido de evitar as colisões afastando-se dos obstáculos detectados e convergindo até o ponto final desejado.

Com relação ao controle cinemático, são utilizadas duas estratégias, denominadas DFL e NTI, em que, comparativamente, observa-se que ambos os controladores apresentam resultados de simulação semelhantes no seguimento da trajetória e sinais de controle aplicados, sendo que tais resultados são significativamente satisfatórios. Além disso, foram realizados experimentos em laboratório usando um robô diferencial Lego EV3, obtendo-se resultados próximos aos previstos em simulação, constatando-se novamente a eficácia do sistema de controle cinemático desenvolvido.

Sendo assim, a partir das simulações e dos experimentos realizados constata-se que o sistema de planejamento e controle desenvolvido realiza o objetivo proposto de planejar uma trajetória que desvie dos obstáculos detectados para atingir o ponto desejado, controlando o veículo de modo que o mesmo percorra esse caminho. Além disso, esse sistema de navegação é aplicável não só no robô diferencial Lego EV3, usado como berço de testes no trabalho, mas que também pode ser estendido para outros sistemas robóticos semelhantes, cuja estrutura seja correspondente a de um veículo móvel terrestre de duas rodas com acionamento diferencial.

Vale ressaltar que a partir dos estudos realizados e dos resultados obtidos no desenvolvimento desta dissertação, publicou-se um artigo [22] que foi apresentado em um simpósio internacional.

5.1 Perspectivas de Trabalhos Futuros

Pode-se incorporar, ao sistema de navegação desenvolvido, um sensoriamento real para que possibilite a realização de experimentos e testes do planejamento da trajetória com a detecção gradativa do ambiente. Como exemplos de sensoriamento, pode-se utilizar sensores infra-vermelho ou de ultra-som, capazes de detectar os obstáculos a uma distância de onde o veículo se encontra igual ou até maior que a de $2,0m$ que foi escolhida neste trabalho. Além disso, pode-se usar câmeras *kinect* no sensoriamento para captar a imagem, processá-la e, assim, obter uma leitura 3D do ambiente ao redor do veículo.

Apesar dos testes de planejamento com o método APF-2 para inúmeros arranjos de obstáculos em diferentes tipos de ambientes, incluindo os exemplos ilustrados neste trabalho, ainda não é possível garantir que este método sempre irá superar o problema dos mínimos locais ou que não irá apresentar outras limitações. Portanto, podem ser realizados estudos e implementações de novas heurísticas no algoritmo do método APF-2, visando eliminar eventuais limitações que venham a ocorrer para configurações e ambientes não previstos.

O sistema de navegação desenvolvido pode ser aplicado futuramente no contexto de uma cadeira de rodas motorizada, que é um tipo de veículo móvel terrestre de duas rodas e acionamento diferencial. Dessa forma, essa aplicação, que faz a cadeira de rodas desviar de obstáculos automaticamente, dará maior mobilidade e segurança para os usuários, uma vez que o uso de controles manuais no movimento tem grandes chances de ocasionar colisões que possivelmente geram acidentes que, no caso de pessoas com limitações físicas extremas, podem ser fatais.

Para fazer uma comparação com os resultados já obtidos usando os controladores DFL e NTI, podem ser implementadas outras estratégias de controle avançadas, como as técnicas envolvendo controladores preditivos, robustos e adaptativos no controle cinemático de veículos diferenciais.

Além de todos esses desdobramentos futuros, o sistema de navegação desenvolvido poderá também ser adaptado para veículos aéreos ou sub-aquáticos, em ambientes tridimensionais, possibilitando a navegação autônoma e segura destes sistemas robóticos.

Referências Bibliográficas

- [1] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementation*. MIT press, 2005.
- [2] R. Siegwart and I. R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. MIT press, 2011.
- [3] T. F. Bastos-Filho, F. A. Cheein, S. M. T. Müller, W. C. Celeste, C. De La Cruz, D. C. Cavalieri, M. Sarcinelli-Filho, P. F. S. Amaral, E. Perez, C. M. Soria, and R. Carelli. Towards a new modality-independent interface for a robotic wheelchair. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 22, NO. 3, 2014.
- [4] R. F. Pinto, A. G. S. Conceição, P. C. M. A. Farias, and E. T. F. Santos. A cost effective open-source three-dimensional reconstruction system and trajectory analysis for mobile robots. *5th ISSNIP-IEEE Biosignals and Biorobotics Conference*, 2014.
- [5] J. C. Latombe. *Robot Motion Planning*. Springer Science & Business Media, 2012.
- [6] R. Mezencio. Implementação do método de campos potenciais para navegação de robôs móveis baseada em computação reconfigurável. Tese de doutorado, USP, 2002.
- [7] R. R. Dos Santos, S. F. P. Saramango, and V. Steffen Jr. Planejamento de trajetória de robô autônomo através do conceito de campo potencial. *16^o Simpósio de Pós-Graduação em Engenharia Mecânica*, 2006.

- [8] R. C. Ribeiro. Estratégia de desvio de obstáculos e planejamento de trajetória para uma cadeira de rodas autônoma. Projeto de graduação, UFRJ, 2015.
- [9] Y. Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. *International Conference on Robotics and Automation*, 1991.
- [10] M. W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Dynamics and Control*. John Wiley & Sons, 2008.
- [11] G. A. Borges, G. S. Deep, and A. M. N. Lima. Controladores cinemáticos de trajetória para robôs móveis com tração diferencial. *VI Simpósio Brasileiro de Automação Inteligente*, 2003.
- [12] F. C. Vieira. Controle dinâmico de robôs móveis com acionamento diferencial. Dissertação de mestrado, UFRN, 2006.
- [13] D. P. F. Pedrosa, P. J. Alsina, and A. A. D. Medeiros. Um método de geração de trajetória para robôs não-holonômicos com acionamento diferencial. *VI Simpósio Brasileiro de Automação Inteligente*, 2003.
- [14] T. A. Cerqueira. Plataforma de comando por voz para uma cadeira de rodas motorizada. Trabalho de conclusão do curso de especialização em automação e controle, SENAI-CIMATEC, 2014.
- [15] P. Corke. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer, 2011.
- [16] T. T. Ribeiro, J. T. Dos Santos, A. G. S. Conceição, and A. L. Da Costa. Sistema microprocessado para controle em tempo real de robôs móveis omni-direcionais. *X Simpósio Brasileiro de Automação Inteligente*, 2011.
- [17] K. Ogata. *Engenharia de Controle Moderno*. Prentice-Hall, 1990.
- [18] B. d'Andrea Novel, G. Bastin, and G. Campion. *Control of nonholonomic wheeled mobile robots by state feedback linearization*. *Journal of Robotics Research*, Vol. 14, 1995.
- [19] C. Samson. *Time-varying Feedback Stabilization of car-like wheeled mobile robots*. *Journal of Robotics Research*, Vol. 12, 1993.

- [20] A. De Oliveira and E. B. Guedes. Uma análise comparativa de kits para robótica educacional. *XXIII Workshop sobre Educação em Informática*, 2015.
- [21] MathWorks. Lego mindstorms ev3 support from matlab. Disponível em: <https://www.mathworks.com/hardware-support/lego-mindstorms-ev3-matlab.html>. Acesso em: 22 Out. 2016.
- [22] T. A. Cerqueira, T. L. M. Santos, and A. G. S. Conceição. A new approach based in potential fields with obstacles avoidance for mobile robots. *13rd Latin American Robotics Symposium*, 2016.

Apêndice: Algoritmo do Planejamento APF-2 em MATLAB

A seguir tem-se o código desenvolvido em MATLAB® para implementação do algoritmo de planejamento pelo método APF-2 estudado neste trabalho.

```
%% Função para o planejamento APF-2

function [xd,yd]=GERACAO_TRAJETORIA_APF_MODIFICADO (xd_antigo,
yd_antigo,xref,yref,xi,yi,Parede,XOB,YOB,raio_obstaculo,l,dr,
vmax,ts,raio_sensor)

%% Ambiente:
[XB,YB]=visao_ambiente (Parede,XOB,YOB,raio_obstaculo,
xi,yi,raio_sensor);

igualdade=1;

objetos=size(XB,2);

%% Teste se a trajetória anterior ainda é factível,
%% mesmo com o novo obstáculo encontrado:
nao_factivel=0;
for i=1:size(xd_antigo,2)
    for j=1:size(XB,2)
        dist_obstaculo=sqrt((xd_antigo(i)-XB(j))^2+
(yd_antigo(i)-YB(j))^2);
        %Trajetória não factível (veículo colidiu com o obstáculo)
        if (dist_obstaculo<0.75*l)
            nao_factivel=1;
        end
    end
end
end

if (nao_factivel==0)
    igualdade=1;
end

if (nao_factivel==1)
    igualdade=0;
end

if (size(xd_antigo,2)==0)
    igualdade=0;
end

xd=[];
yd=[];
```

```

if (igualdade==0)

    K=1;

    atracao=0;

    erro=sqrt((xref-xi)^2+(yref-yi)^2);

    % distância total entre o veículo e a posição de referência:
    dist=erro;

    %Carga referência do Veículo
    q1=1;

    %quantidade de tentativas de obter a trajetória
    tentativas=0;

    %% Looping
    convergencia=1/2;

    %enquanto o veículo ainda não chegou no ponto de referência
    while (erro>convergencia)

        tentativas=tentativas+1;

        Qb=1*ones(size(XB,2)); %Cargas dos objetos

        xii=xi;
        yii=yi;

        xd=xi;
        yd=yi;

        erro=sqrt((xref-xii)^2+(yref-yii)^2);

        % variação heurística incremental na atração:
        atracao=atracao+tentativas^2;

        q2=atracao; % carga do ponto de referência

        %% Gerando pontos da trajetória
        contagem=0;

        while
            (erro>convergencia)&&(contagem<=(1.5+tentativas/100)*dist/dr)
                % Obtém o próximo ponto da trajetória:
                [postx,posty] = APF_MODIFICADO(K, q1, q2, Qb, xii,
                    yii, xref, yref, objetos, XB, YB, dr);

                xii=postx;
                yii=posty;

```

```

        xd=[xd postx];
        yd=[yd posty];

        contagem=contagem+1;

        erro=sqrt((xref-xii)^2+(yref-yii)^2);

        aprox=(dist/erro)^2; %aproximação quadrada

        % variação heurística na carga dos obstáculos:
        Qb=(1/aprox)*ones(size(XB,2),1);

    end

end

if (size(xd)~=0)

    % Ajuste do caminho para facilitar a etapa do controle:
    [xd,yd]=adaptador_caminho(xd,yd,vmax,ts);

    %% Teste se a trajetória é factível:
    nao_factivel=0;
    for i=1:size(xd,2)
        for j=1:size(XB,2)
            dist_obstaculo=sqrt((xd(i)-XB(j))^2+(yd(i)-YB(j))^2);
            if (dist_obstaculo<l/2)
                nao_factivel=1; %colisão esperada
            end
        end
    end

    if (nao_factivel==1) % não há planejamento
        xd=xi;
        yd=yi;
    end

end

end

end

```