José Miguel Vilca Ventura

Estudo de Coordenação de Robôs Móveis com Obstáculos ¹

Dissertação de Mestrado apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Mestre em Ciências, Programa de Engenharia Elétrica

Área de Concentração: Sistemas Dinâmicos
Orientador: Prof. Dr. Marco Henrique Terra

São Carlos
2011

¹Trata-se da versão corrigida da dissertação. A versão original se encontra disponível na EESC/USP que aloja o Programa de Pós-Graduação de Engenharia Elétrica.

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

V695e Vilca, José Miguel Ventura.
Estudo de coordenação de robôs móveis com obstáculos.
/ José Miguel Vilca Ventura ; orientador Marco Henrique
Terra. São Carlos, 2011.

Dissertação (Mestrado - Programa de Pós-Graduação em
Engenharia Elétrica e Área de Concentração em Sistemas
Dinâmicos)-- Escola de Engenharia de São Carlos da
Universidade de São Paulo, 2011.

1. Robótica. 2. Robôs móveis. 3. Fusão de dados. 4.
Filtro de Kalman estendido. 5. Desvio de obstáculos. 6.
Controle de formação. 7. Controle Markoviano. I. Título.

FOLHA DE JULGAMENTO

Candidato: Bacharel **JOSÉ MIGUEL VILCA VENTURA**

Título da dissertação: **Estudo de coordenação de robôs móveis com obstáculos.**

Data da defesa: 15/09/2011:

Comissão Julgadora:

Prof. Associado **Marco Henrique Terra (Orientador)**
(Escola de Engenharia de São Carlos/EESC)

Prof. Dr. **Valdir Grassi Júnior**
(Escola de Engenharia de São Carlos/EESC)

Profª. Associada **Roseli Aparecida Francelin Romero**
(Instituto de Ciências Matemáticas e de Computação/ICMC/USP)

Resultado:

Aprovado

aprovado

Aprovado

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica:
Prof. Titular **Denis Vinicius Coury**

Presidente da Comissão de Pós-Graduação:
Prof. Associado **Paulo Cesar Lima Segantine**

Dedicatória

*Dedico este trabalho aos meus avós
Alejandro Ventura Fernandez, Edelmira
Velarde Rossell e Leonidas Vilca Jimenez
(in memoriam).*

Agradecimentos

Agradeço à Deus e à memória das pessoas queridas por me dar toda a força necessária para a realização deste trabalho.

Aos meus pais Leonidas Vilca e Eleuteria Ventura com amor e gratidão pelo carinho e apoio ao longo do período deste mestrado. Aos meus irmãos por sempre estarem ao meu lado quando necessário.

Ao Prof Dr. Marco Henrique Terra pela oportunidade, confiança, orientação e pelo tempo dedicado na realização deste trabalho.

À minha namorada Giovana e à todos os meus companheiros e amigos pela disposição em ajudar sempre que necessário, pelas valiosas colaborações e pela paciência durante a realização deste trabalho.

Aos professores e funcionários da Escola de Engenharia de São Carlos, pelas contribuições durante a realização deste trabalho.

Ao conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo auxílio financeiro.

*“Quanto mais aumenta nosso
conhecimento, mais evidente fica nossa
ignorância.”*

John F. Kennedy

Resumo

VILCA, J. M. V. (2011). *Estudo de coordenação de robôs móveis com obstáculos*. Dissertação (Mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2011.

Coordenação de robôs móveis é um tópico importante de pesquisa dado que existem tarefas que podem ser desenvolvidas de forma mais eficiente e com menor custo por um grupo de robôs do que por um só robô. Nesta dissertação é apresentado um estudo sobre coordenação de robôs móveis para o problema de navegação em ambientes externos. Para isso, foi desenvolvido um sistema de localização utilizando os dados de odometria e do receptor GPS, e um sistema de desvio de obstáculos para planejar a trajetória livre de obstáculos. Os movimentos coordenados foram realizados em função de um líder e qualquer robô da formação pode assumir a liderança. A liderança é assumida pelo robô que ultrapassar a distância mínima a um obstáculo. Movimentos estáveis são gerados através de uma lei de controle descentralizada baseada nas coordenadas dos robôs. Para garantir a estabilidade da formação quando há alternância de líder ou remoção de robôs, foi feito controle tolerante a falhas para um grupo de robôs móveis. O controle tolerante a falhas é baseado em controle $\mathcal{H}_\infty$ por realimentação da saída de sistemas lineares sujeitos a saltos Markovianos para garantir a estabilidade da formação quando um dos robôs é perdido durante o movimento coordenado. Os resultados do sistema de localização mostram que o uso de filtro robusto para a fusão de dados produz uma melhor estimativa da posição do robô móvel. Os resultados também mostram que o sistema de desvio de obstáculos é capaz de gerar uma trajetória livre de obstáculos em ambientes desconhecidos. E por fim, os resultados do sistema de coordenação mostram que o grupo de robôs mantém a formação desejada percorrendo a trajetória de referência na presença de distúrbios ou quando um robô sai da formação.

Palavras-Chave: *Robôs móveis, fusão de dados, filtro de Kalman estendido, desvio de obstáculos, robótica, controle de formação, controle Markoviano.*

Abstract

VILCA, J. M. V. (2011). *Study of coordination of mobile robots with obstacle avoidance*. Dissertation (Master of Science) - School of Engineering of São Carlos, University of São Paulo, São Carlos, 2011.

Coordination of mobile robots is an important topic of research because there are tasks that may be too difficult for a single robot to perform alone, these tasks can be performed more efficiently and cheaply by a group of mobile robots. This dissertation presents a study on the coordination of mobile robots to the problem of navigation in outdoor environments. To solve this problem, a localization system using data from odometry and GPS receiver, and an obstacle avoidance system to plan the collision-free trajectory, were developed. The coordinated motions are performed by the robots that follow a leader, and any robot of the formation can assume the leadership. The leadership is assumed by a robot when it exceeds the threshold distance to an obstacle. Stable motions are generated by a decentralized control law based on the robots' coordinates. To ensure the stability formation when there is alternation of leader or one of the robots is removed, we made a fault tolerant control for a group of mobile robots. The fault tolerant approach is based on output feedback $\mathcal{H}_\infty$ control of Markovian jump linear systems to ensure stability of the formation when one of the robots is lost during the coordinated motion. The results of the localization system show that the use of robust filter for data fusion produces a better estimation of the mobile robot's position. The results also show that the obstacle avoidance system is capable of generating a path free from obstacles in unknown environments. Finally, the results of the coordination system show that the group of robots maintain the desired formation along the reference trajectory in the presence of disturbance or removal of one of them.

Keywords: Mobile robots, data fusion, extended Kalman filter, obstacle avoidance, robotics,

formation control, Markovian control.

Publicações

1. VILCA, J. M. V.; TERRA, M. H.; GRASSI JR., V. (2011). Formation control with leadership alternation for obstacle avoidance. *In: 18th World Congress of International Federation of Automatic Control, IFAC*, Milano, Italy.

Lista de Figuras

2.1	Geometria do RMRD.	7
2.2	Forças de tração e resistivas do robô móvel.	11
3.1	Sistemas de coordenadas retangulares ECEF (<i>Earth Centered - Earth Fixed</i>) e sistema de coordenadas geodésicas (Plano Tangente).	20
3.2	Exemplo de quantização de um <i>encoder</i> com resolução $C = 16$ ciclos por revolução.	21
3.3	Pseudodistância da posição p ao i -ésimo satélite.	26
3.4	Esquema da fusão de dados.	29
4.1	Robô móvel de rodas deslizantes, <i>Pioneer 3AT</i>	36
4.2	Posição do robô móvel usando GPS.	37
4.3	Posição real e estimada usando fusão de dados do GPS e do <i>encoder</i>	37
4.4	Erro da estimativa da posição usando fusão de dados do GPS e do <i>encoder</i>	38
4.5	Variância dos filtros usados para a fusão de dados do GPS e do <i>encoder</i>	38
4.6	Posição real e estimada usando fusão de dados do GPS e do <i>encoder</i>	40
5.1	O LTG usando um sensor laser de varredura de 180°	44
5.2	Método para o acompanhamento da borda de um obstáculo.	46
5.3	Domínio contínuo da frente de onda.	47
6.1	Esquema do sistema de acompanhamento de trajetória.	53
6.2	Exemplo 1 da aplicação do algoritmo <i>Tangent Bug</i> para desvio de obstáculos.	54

6.3	Exemplo 2 da aplicação do algoritmo <i>Tangent Bug</i> para desvio de obstáculos.	55
6.4	Trajeto�ria gerada pelo algoritmo E^* e percorrida pelo RMRD.	56
6.5	Trajeto�ria de refer�ncia, real e estimada.	56
6.6	Trajeto�ria replanejada gerada pelo algoritmo E^* e percorrida pelo RMRD.	57
6.7	Trajeto�ria de refer�ncia, real e estimada.	57
7.1	Representa��o de uma forma��o com tr�s RMs com altern�ncia de l�der.	63
8.1	Modelo Markoviano para $n = 6$ rob�s m�veis.	69
9.1	Sistema de controle de acompanhamento de trajet�ria e sistema de controle tole- rante a falhas para os RMs.	74
9.2	D�grafos para possibilidade de falha nos rob�s.	76
9.3	Trajeto�ria da forma��o com falha no rob� 5.	77
9.4	Dist�rbios aplicados nas rodas esquerda e direita dos rob�s.	77
9.5	Velocidades das rodas esquerdas e direitas dos rob�s da forma��o.	78
9.6	Torques das rodas esquerdas e direitas dos rob�s da forma��o.	78
9.7	Grupo de 6 <i>Pioneer 3-AT</i> com falha no rob� 5.	80
9.8	Trajeto�rias do grupo de 6 <i>Pioneer 3-AT</i>	81
9.9	D�grafos para cada l�der da forma��o.	81
9.10	Trajeto�ria de refer�ncia gerada pelo algoritmo <i>Tanget Bug</i>	82
9.11	Trajeto�ria da forma��o com altern�ncia de l�der e falha no rob� 5.	82
9.12	Trajeto�ria de refer�ncia gerada pelo algoritmo E^*	83
9.13	Trajeto�ria da forma��o com altern�ncia de l�der e falha no rob� 5.	83
A.1	M�dulo receptor GPS s�rie MTK-3301.	95
A.2	Especifica��es do m�dulo receptor GPS.	96
A.3	Mensagem GGA e GSV do padr�o NMEA.	98

Lista de Tabelas

4.1	Parâmetros do RMRD <i>Pioneer 3-AT</i>	35
9.1	Índices de desempenho.	79

Lista de Abreviaturas e Siglas

RM - **R**obô **M**óvel

RMRD - **R**obô **M**óvel de **R**odas **D**eslizantes

CIR - **C**entro **I**ntantâneo de **R**otação

ECEF - ***E**arth **C**entered - **E**arth **F**ixed*

GPS - ***G**lobal **P**osition **S**ystem*

BDU - ***B**ounded **D**ata **U**ncertainties*

LPV - **L**inear com **P**arâmetros **V**ariantes

LMI - ***L**inear **M**atrix **I**nequality*

DML - ***D**esigualdades **M**atriciais **L**ineares*

SLSM - **S**istemas **L**ineares sujeitos a **S**altos **M**arkovianos

Lista de Símbolos

$\mathcal{V}$	Conjunto de vértices do grafo
$\mathcal{E}$	Conjunto de bordas do grafo
Γ	Grafo direcionado
Q	Matriz de adjacência
D	Matriz diagonal
L_Γ	Laplaciano direcionado
$M(\mathbf{q})$	Matriz de Inércia
$C(\mathbf{q}, \dot{\mathbf{q}})$	Matriz de forças de Coriolis e centrípetas
z	Vetor da saída
w, δ	Distúrbios
$\ T_{zw}\ _\infty$	Norma $\mathcal{H}_\infty$ da função de transferência entre o distúrbio w e a saída z
γ	Nível de atenuação
u	Entrada de controle
$\mathcal{L}$	Ganho
y	Erros de posição
C^1	Funções continuamente diferenciáveis

Sumário

Publicações	xiii
Lista de Figuras	xv
Lista de Tabelas	xvii
Lista de Abreviaturas e Siglas	xix
Lista de Símbolos	xxi
1 Introdução	1
1.1 Revisão bibliográfica	2
1.1.1 Localização de robôs móveis	2
1.1.2 Desvio de obstáculos	2
1.1.3 Controle de formação	3
1.1.4 Controle $\mathcal{H}_\infty$ aplicado a robôs	3
1.2 Objetivo	4
1.3 Disposição dos capítulos	5
2 Modelagem de robô móvel de rodas deslizantes	7
2.1 Modelo cinemático	8
2.2 Modelo dinâmico	11
2.3 Formulação do problema	13
2.4 Controlador baseado na cinemática	15
I Localização de robôs móveis em ambientes externos	17
3 Fusão de dados de <i>encoder</i> e GPS usando filtragem	19
3.1 <i>Dead reckoning</i>	20
3.1.1 <i>Encoder</i>	21

3.2	Modelo de erro do sistema	23
3.3	Modelo do GPS	24
3.3.1	Pseudodistância do GPS	25
3.3.2	Modelo da diferença da medida	27
3.4	Fusão de dados usando filtro de Kalman robusto	29
3.4.1	Filtro de Kalman estendido	31
3.4.2	Filtro de Kalman robusto BDU	32
4	Resultados	35
4.1	Simulação computacional da fusão de dados do <i>encoder</i> e do receptor GPS	36
4.2	Resultados experimentais da fusão de dados do <i>encoder</i> e do receptor GPS	39
II	Algoritmo para desvio de obstáculos	41
5	Algoritmo para desvio de obstáculos	43
5.1	Algoritmo <i>Tangent Bug</i>	43
5.1.1	Considerações na implementação do algoritmo <i>Tangent Bug</i>	45
5.2	Algoritmo E^*	47
5.2.1	Representação do ambiente	48
5.2.2	Propagação da frente de onda	49
5.2.3	Grafo <i>upwind</i>	49
5.2.4	<i>Kernel</i> de interpolação	49
5.2.5	O Algoritmo	50
6	Resultados	53
6.1	Simulação computacional do algoritmo <i>Tangent Bug</i>	54
6.2	Simulação computacional do algoritmo E^*	55
III	Coordenação de robôs móveis	59
7	Controle de formação	61
7.1	Modelo do sistema	61
7.2	Estratégia de alternância de líder	65
8	Modelo Markoviano dos robôs móveis	67
8.1	Controle $\mathcal{H}_\infty$ por realimentação de saída para sistemas lineares sujeitos a saltos Markovianos	71

9	Resultados	73
9.1	Simulação computacional do controlador Markoviano	75
9.1.1	Estudo comparativo	79
9.1.2	Implementação do controlador Markoviano	79
9.2	Simulação computacional do controlador Markoviano com alternância de líder . .	81
10	Conclusões	85
	Referências Bibliográficas	87
	Apêndice	93
A	Cálculo da posição ECEF utilizando GPS	95
A.1	Módulo receptor GPS	95
A.2	Transformação para o sistema de coordenadas ECEF	97
B	Teoria algébrica dos grafos	99
B.1	Conceitos básicos	99

Capítulo 1

Introdução

Nos últimos anos as pesquisas em controle e coordenação de múltiplos robôs móveis têm crescido significativamente, em virtude das aplicações em exploração [1; 2], procura e resgate [3], mapeamento de locais desconhecidos [4; 5], transporte de grandes objetos [6; 7], detecção de artefatos explosivos enterrados a pouca profundidade [8] e controle de formação [9; 10].

Para obter êxito no movimento coordenado dos robôs na realização de uma tarefa pré-estabelecida é necessário que o controle utilizado para os robôs móveis seja robusto o suficiente em relação aos distúrbios externos relacionados, tais como, desníveis no ambiente, deslizamento das rodas, colisão com obstáculos e incertezas paramétricas.

Este trabalho utiliza estratégias de controle baseadas em modelos dinâmicos, que melhor representam o comportamento real do sistema. A dinâmica não linear dos robôs pode ser representada como um sistema linear a parâmetros variantes (LPV) onde o estado está em função dos parâmetros. Em virtude disso, usaremos a terminologia quase-LPV. Para a atenuação dos distúrbios externos e incertezas paramétricas, o critério $\mathcal{H}_\infty$ é um dos mais populares procedimentos para atingir este objetivo. Para manter a robustez do sistema quando há a remoção de um dos robôs da formação, será utilizado um controlador Markoviano $\mathcal{H}_\infty$ que tem se mostrado bastante eficiente para tratar de mudanças abruptas que ocorre no sistema.

1.1 Revisão bibliográfica

1.1.1 Localização de robôs móveis

A estimação da localização do robô móvel de forma precisa é um problema crítico para aplicações de planejamento, navegação e controle em ambientes internos ou externos. Os métodos mais comuns para fazer essas estimativas baseiam-se em sistemas de navegação inerciais (INS), odometria, sistema de medida com laser (LADAR), receptores GPS e sistemas de visão artificial [11; 12; 13].

Em [14] apresentou-se um método para a localização em ambientes externos urbanos quando o receptor GPS não está disponível, o método utiliza o filtro de soma Gaussiano para o sistema de visão artificial. Em [15] foi utilizado o sistema de visão artificial para o reconhecimento de lugares, isto é, primeiro estuda-se o lugar, na vez seguinte, o robô percebe certos lugares e faz a fusão com a odometria para gerar um caminho mas suave.

Existem diversos métodos para localização em ambiente externos utilizando a fusão de dados sensores de odometria, unidades inerciais de medida (IMU), INS, sensores Laser junto com receptores GPS [16; 17; 18; 19]. Em [20] apresentou-se a fusão de dados do IMU, do receptor GPS e do sistema de detecção de distância (LIDAR) para obter uma localização de maior precisão e robustez na presença de obstáculos dinâmicos. O objetivo de usar as medidas do sistema INS ou odometria junto com o receptor GPS é para obter uma estimativa mais exata da posição do robô móvel.

1.1.2 Desvio de obstáculos

Na navegação autônoma em ambientes externos, muitas vezes não se tem informação completa do ambiente. Para realizar essa operação, o robô deve possuir sensores que ajudem a realizar a navegação. Os sensores mais comuns utilizados são sonares, lasers, câmeras, etc. Diferentes algoritmos para tratar do desvio de obstáculos em ambientes desconhecidos são encontrados na literatura, como por exemplo o Algoritmo de Stentz (D^*) [21; 22] que pode ser considerado a versão dinâmica do algoritmo de Dijkstra [22], e os algoritmos *Bug 1*, *Bug 2* e *Tangente Bug* [23; 24; 25; 26; 27] cujo planejamento está baseado nas idéias de exploração de um labirinto.

Existem métodos que utilizam informações prévias do ambiente, o método de decomposição de células constrói regiões nas áreas livres de obstáculos e as codifica as células em grafos [21;

12]. O método de campos potenciais trata ao robô como um ponto que está sob a influência de forças que atraem ele ao alvo enquanto é repellido dos obstáculos [21; 12].

1.1.3 Controle de formação

O controle de formação é uma importante estratégia para a coordenação de um grupo de veículos, através da qual é possível fazer com que os veículos se estabilizem e mantenham uma formação pré-definida no movimento coordenado. Muitas abordagens de controle foram apresentadas para resolver os problemas de controle de formação, como por exemplo, a abordagem que utiliza a estratégia líder-seguidor apresentada em [28; 29]; abordagem utilizando estrutura virtual [30; 31]; e o método baseado em comportamento [32; 33].

A teoria dos grafos é uma ferramenta capaz de tratar adequadamente os problemas que envolvam comunicação de dados entre agentes do sistema. A abordagem via teoria dos grafos foi proposta por [34] para o controle cooperativo de múltiplos sistemas lineares. E, em virtude disto, diferentes leis de controle puderam ser projetadas com o auxílio da teoria dos grafos (vide [35] e [36]).

1.1.4 Controle $\mathcal{H}_\infty$ aplicado a robôs

Foram encontradas três importantes estratégias de controle de sistemas robóticos na literatura: na primeira, o modelo de um sistema robótico é considerado completamente conhecido, e este modelo é utilizado pelo controlador proposto em [37]; na segunda, os parâmetros do modelo são desconhecidos e são estimados baseados na propriedade de parametrização linear, resultados clássicos sobre controle adaptativo podem ser vistos em [37; 38]; e na terceira, o modelo é desconhecido e uma abordagem inteligente (baseada em redes neurais ou lógica fuzzy) é usada para estimar o modelo (veja, por exemplo, [39] e [40]). Se, em adição às incertezas paramétricas, distúrbios externos estão presentes, a dificuldade para controlar um sistema robótico aumenta. Uma abordagem interessante para resolver este problema de controle é baseado no controlador $\mathcal{H}_\infty$ não linear, que visa atenuar os efeitos de todos distúrbios no desempenho do sistema. O Controle $\mathcal{H}_\infty$ não linear consiste em garantir que o ganho $\mathcal{L}_2$ entre o distúrbio e a saída, seja limitado por um nível de atenuação $\gamma > 0$.

Controladores $\mathcal{H}_\infty$ baseados nas estratégias descritas anteriormente são propostos em [39; 40; 41; 42; 43; 44; 45; 46; 47], para robôs manipuladores. Em [41], uma solução explícita para

o problema de controle $\mathcal{H}_\infty$ não linear, na qual o modelo do manipulador é considerado completamente conhecido, é desenvolvida baseada na Teoria dos Jogos (TJ) (em [48] uma metodologia similar é usada para controlar um manipulador real). Controladores $\mathcal{H}_\infty$ não lineares para sistemas LPVs, que têm sido aplicados em robôs manipuladores, podem ser vistos em [42; 43; 46; 47].

Um algoritmo de controle adaptativo $\mathcal{H}_\infty$ não linear é proposto em [44], no qual um projeto de controle robusto de acompanhamento de trajetória considera que os parâmetros desconhecidos podem ser aprendidos por uma lei clássica de adaptação atualizável. Controles adaptativos $\mathcal{H}_\infty$ não lineares baseados em técnicas inteligentes podem ser vistos em [39; 40; 45].

Em [49] foi proposta uma combinação de um controlador baseado na cinemática e um controlador $\mathcal{H}_\infty$ robusto baseado na dinâmica para acompanhamento de trajetória. A solução proposta em [49] resulta em matrizes constantes do controlador, e não leva em consideração a natureza variante dos parâmetros do robô. Já em [50], equações dinâmicas de um robô móvel são descritas na forma quase-LPV. Um controlador baseado no modelo cinemático, proposto em [51], é utilizado para gerar as velocidades desejadas para as rodas e controladores robustos baseados no modelo dinâmico são projetados levando em consideração as variações paramétricas do robô.

Na literatura existe uma vasta gama de resultados envolvendo o controle cinemático para o problema de alternância de líder. A estratégia do controle robusto baseado na dinâmica de robôs sujeitos a alternância de líder e a estratégia do controle de formação tolerantes a falhas baseado no modelo do sistema sujeito a saltos Markovianos foram apresentados em [52; 53].

1.2 Objetivo

Esta pesquisa tem como finalidade um estudo sobre a coordenação de robôs móveis com desvio de obstáculos em ambientes externos, assim como, projetar controladores $\mathcal{H}_\infty$ não lineares para o problema de navegação com coordenação de robôs móveis. Isto será feito através de métodos de localização, desvio de obstáculos e controle para um conjunto de robôs móveis, considerando que qualquer um dos robôs possa assumir a liderança da formação e que robôs possam ser removidos durante a trajetória.

1.3 Disposição dos capítulos

No **Capítulo 2** é descrita a modelagem completa, cinemática e dinâmica, de um robô móvel de rodas deslizantes, sendo que, o controle cinemático utilizado para obter as velocidades angulares desejadas é calculado com base na postura atual e na trajetória de referência.

No **Capítulo 3** é descrito o processo de estimativa da posição do robô móvel usando a fusão de dados do *encoder* e do receptor GPS.

No **Capítulo 4** são apresentados os resultados obtidos através de simulação computacional para a localização do robô móvel.

No **Capítulo 5** são descritos os algoritmos utilizados para o desvio de obstáculos em ambientes desconhecidos.

No **Capítulo 6** são apresentados os resultados obtidos através de simulação computacional para o algoritmo de desvio de obstáculos.

No **Capítulo 7** é apresentado o desenvolvimento da teoria de controle de formação proposta para um conjunto de robôs móveis.

No **Capítulo 8** é apresentado o controlador Markoviano tolerante a falhas proposto para um conjunto de robôs móveis.

No **Capítulo 9** são apresentados os resultados obtidos através de simulação computacional para os controladores projetados.

No **Capítulo 10** são apresentadas as conclusões referentes aos resultados obtidos e as previsões de trabalhos futuros.

Capítulo 2

Modelagem de robô móvel de rodas deslizantes

Neste capítulo, são apresentadas a cinemática e o modelo dinâmico de um robô móvel de rodas deslizantes (RMRD). A rotação de um RMRD é realizada pelo acionamento diferencial dos pares de rodas em cada lado do robô, isto é, o par de rodas de um lado gira a uma velocidade maior que o par de rodas do outro lado. A modelagem do robô está baseada em [54; 55]. O modelo dinâmico é reescrito em espaço de estados na forma quase linear a parâmetros variantes (quase-LPV).

Figura 2.1: Geometria do RMRD.

2.1 Modelo cinemático

A geometria do robô móvel que estamos considerando neste trabalho é mostrada na Figura 2.1, sendo (X, Y) o sistema de coordenadas inercial; (X', Y') o sistema de coordenadas do corpo; CG o centro de gravidade do robô; (x, y) as coordenadas do robô no centro de gravidade no sistema de coordenadas inerciais; ψ o ângulo entre o eixo X' e o eixo X , CIR o centro instantâneo de rotação; (x'_{CIR}, y'_{CIR}) as coordenadas do centro instantâneo de rotação; a a distância entre o eixo de simetria da roda atuada frontal paralelo ao eixo Y' e o centro de gravidade; b a distância entre o eixo de simetria paralelo ao eixo Y' da roda atuada traseira e o centro de gravidade; c a distância entre o eixo de simetria paralelo ao eixo X' da roda atuada e o eixo X' e r o raio das rodas atuadas.

No sistema de coordenadas do corpo (X', Y') , a velocidade linear do robô móvel é dada por $\mathbf{v} = [v_{x'} \ v_{y'} \ 0]^T$, sendo $v_{x'}$ e $v_{y'}$ as componentes de $\mathbf{v}$ no eixo X' e Y' , e a velocidade angular por $\boldsymbol{\omega} = [0 \ 0 \ \omega]$, sendo $\omega = \dot{\psi}$. As velocidades lineares e angulares de cada roda atuada são dadas por v_i e ω_i , sendo $i = 1, 2, \dots, 4$ referente à i -ésima roda atuada.

As velocidades absolutas $\dot{x}$, $\dot{y}$ e $\dot{\psi}$ no sistema de coordenadas inercial são dadas por

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{x'} \\ v_{y'} \\ \omega \end{bmatrix} = R(\psi) \begin{bmatrix} v_{x'} \\ v_{y'} \\ \omega \end{bmatrix}. \quad (2.1)$$

Diferenciando a equação acima em relação ao tempo obtém-se

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\psi} \end{bmatrix} = R(\psi) \begin{bmatrix} \dot{v}_{x'} - \dot{\psi} v_{y'} \\ \dot{v}_{y'} + \dot{\psi} v_{x'} \\ \dot{\omega} \end{bmatrix} = R(\psi) \begin{bmatrix} a_{x'} \\ a_{y'} \\ \dot{\omega} \end{bmatrix}. \quad (2.2)$$

Da Figura 2.1, como mostrado em [55], obtém-se

$$v_L = v_{x'1} = v_{x'3},$$

$$v_R = v_{x'2} = v_{x'4},$$

$$v_F = v_{y'1} = v_{y'2},$$

$$v_B = v_{y'3} = v_{y'4},$$

$$\begin{bmatrix} v_L \\ v_R \\ v_F \\ v_B \end{bmatrix} = \begin{bmatrix} 1 & c \\ 1 & -c \\ 0 & -x'_{CIR} + a \\ 0 & -x'_{CIR} - b \end{bmatrix} \begin{bmatrix} v_{x'} \\ \omega \end{bmatrix}, \quad (2.3)$$

$$\boldsymbol{\omega}_w = \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} = \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} v_L \\ v_R \end{bmatrix}, \quad (2.4)$$

$$\boldsymbol{\eta} = \begin{bmatrix} v_{x'} \\ \omega \end{bmatrix} = r \begin{bmatrix} \frac{\omega_L + \omega_R}{2} \\ \frac{\omega_L - \omega_R}{2c} \end{bmatrix}, \quad (2.5)$$

sendo $v_F, v_B, v_L, v_R, \omega_L, \omega_R, \boldsymbol{\eta}$ as velocidades lineares das rodas frontais e traseiras do robô, as velocidades lineares das rodas do lado esquerdo e direito do robô, as velocidades angulares das rodas do lado esquerdo e direito do robô e a nova entrada de controle, respectivamente.

Escrevendo matricialmente (2.5) e isolando o vetor $\boldsymbol{\omega}_w$, tem-se

$$\begin{bmatrix} v_{x'} \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{2c} & -\frac{r}{2c} \end{bmatrix} \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix}, \quad (2.6)$$

$$\begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{c}{r} \\ \frac{1}{r} & -\frac{c}{r} \end{bmatrix} \begin{bmatrix} v_{x'} \\ \omega \end{bmatrix}. \quad (2.7)$$

Diferentemente da maioria dos robôs móveis, a velocidade lateral do RMRD é geralmente diferente de zero, isso vem da estrutura mecânica do RMRD que faz o deslizamento lateral necessário, caso o veículo mude de orientação. Quando a velocidade do robô é $v_{y'} = 0$, não há deslocamento lateral. A velocidade angular ω e a velocidade lateral $v_{y'}$ desaparecem quando o robô movimenta-se em linha reta, e o CIR vai para infinito ao longo do eixo Y' . Já em uma trajetória curvada o CIR desloca x'_{CIR} ao longo do eixo X' . Se x'_{CIR} ultrapassar a base das rodas do robô, o veículo desliza com perda de estabilidade de movimento. Assim, para completar o

modelo cinemático do RMRD, a restrição não-holonômica a seguir foi introduzida em [54]

$$v_{y'} = x'_{CIR} \dot{\psi}, \quad (2.8)$$

sendo $x'_{CIR} \in (-b, a)$. Por outro lado, de (2.1) obtém-se

$$v_{y'} = -\sin \psi \dot{x} + \cos \psi \dot{y}. \quad (2.9)$$

Substituindo a equação acima em (2.8) e colocando todos os termos da equação no lado esquerdo, obtém-se a seguinte restrição não-holonômica:

$$\begin{bmatrix} -\sin \psi & \cos \psi & -x'_{CIR} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = A(\mathbf{q}) \dot{\mathbf{q}} = 0, \quad (2.10)$$

sendo $\mathbf{q} = [x \ y \ \psi]^T$. As velocidades $\dot{\mathbf{q}}$ podem ser expressas como

$$\dot{\mathbf{q}} = S(\mathbf{q}) \boldsymbol{\eta}, \quad (2.11)$$

sendo $\boldsymbol{\eta} = [v_{x'} \ \omega]^T$ e $S(\mathbf{q})$ é uma matriz de posto completo, cujas colunas estão no espaço nulo de $A(\mathbf{q})$, ou seja,

$$S^T(\mathbf{q}) A^T(\mathbf{q}) = 0 \quad (2.12)$$

e

$$S(\mathbf{q}) = \begin{bmatrix} \cos \psi & -x'_{CIR} \sin \psi \\ \sin \psi & x'_{CIR} \cos \psi \\ 0 & 1 \end{bmatrix}. \quad (2.13)$$

É interessante observar que como a $\dim(\boldsymbol{\eta}) = 2 < \dim(\mathbf{q}) = 3$, a equação (2.11) descreve a cinemática de um robô subatuado. E ainda que este é um sistema não-holonômico devido à restrição (2.8).

A equação cinemática (2.11) pode ser escrita como

$$\dot{x} = v_{x'} \cos \psi - \omega x'_{CIR} \sin \psi, \quad (2.14)$$

$$\dot{y} = v_{x'} \sin \psi + \omega x'_{CIR} \cos \psi, \quad (2.15)$$

$$\dot{\psi} = \omega. \quad (2.16)$$

2.2 Modelo dinâmico

A Figura 2.2 mostra as forças de tração $F_{x'i}$ que estão sujeitas às forças resistivas longitudinais $R_{x'i}$. Para reduzir o deslize longitudinal, assume-se que a atuação das rodas é igual em cada lado, ou seja, $F_{x'1} = F_{x'3}$ e $F_{x'2} = F_{x'4}$. As forças laterais $R_{y'i}$ atuam sobre as rodas em consequência do deslizamento lateral. O momento resistivo M_r em volta do centro de massa se opõe ao momento M , geralmente induzido pelas forças $R_{y'i}$ e $R_{x'i}$.

Figura 2.2: Forças de tração e resistivas do robô móvel.

Para um veículo, como mostrado na Figura 2.2, de massa m e de inércia I no centro de massa, as equações de movimento no sistema de coordenadas do corpo são

$$m.a_{x'} = 2F_{x'1} + 2F_{x'2} - R_{x'}, \quad (2.17)$$

$$m.a_{y'} = -R_{y'},$$

$$I.\ddot{\psi} = 2c(F_{x'1} - F_{x'2}) - M_r.$$

Para representar as forças longitudinais $R_{x'}$, as forças laterais $R_{y'}$, e o momento resistivo M_r , deve-se considerar mg como a carga gravitacional, sendo m a massa do robô e g a aceleração da gravidade, que é dividida sobre as rodas do robô, além de introduzir o modelo Coulomb de atrito para o contato das rodas com a superfície. Desse modo, desde que o robô tenha uma linha mediana longitudinal de simetria, obtém-se

$$\begin{aligned} R_{z'1} = R_{z'2} &= \frac{b}{2(a+b)}mg, \\ R_{z'3} = R_{z'4} &= \frac{a}{2(a+b)}mg. \end{aligned} \quad (2.18)$$

Em velocidades baixas, a carga lateral transferida devido às forças centrífugas sobre trajetórias curvas pode ser desconsiderada. No caso de superfícies duras, pode-se assumir que o contato entre as rodas e o solo é retangular e que a carga vertical produz uma pressão uniforme distribuída. Nessas condições, $R_{x'i} = f_r R_{z'i} \text{sgn}(v_{x'i})$, sendo f_r o coeficiente resistivo de rolagem, e $\text{sgn}(\cdot)$ a função sinal, assumido independentemente da velocidade. Assim, a força resistiva total longitudinal é

$$R_{x'} = \sum_{i=1}^4 R_{x'i} = f_r \frac{mg}{2} (\text{sgn}(v_{x'1}) + \text{sgn}(v_{x'2})). \quad (2.19)$$

Introduzindo o coeficiente de atrito lateral μ , a força lateral atuando sobre cada roda será $R_{y'i} = \mu R_{z'i} \text{sgn}(v_{y'i})$. A força lateral total será

$$R_{y'} = \sum_{i=1}^4 R_{y'i} = \mu \frac{mg}{a+b} (b \text{sgn}(v_{y'1}) + a \text{sgn}(v_{y'3})), \quad (2.20)$$

enquanto o momento resistivo será

$$\begin{aligned} M_r &= a(R_{y'1} + R_{y'2}) - b(R_{y'3} + R_{y'4}) + c[(R_{x'1} + R_{x'3}) - (R_{x'2} + R_{x'4})] \\ &= \mu \frac{abmg}{a+b} (\text{sgn}(v_{y'1}) - \text{sgn}(v_{y'3})) + f_r cmg (\text{sgn}(v_{x'1}) - \text{sgn}(v_{x'2})). \end{aligned} \quad (2.21)$$

Escrevendo (2.17) de forma matricial, tem-se

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} a_{x'} \\ a_{y'} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 2F_{x'1} + 2F_{x'2} - R_{x'} \\ -R_{y'} \\ 2c(F_{x'1} - F_{x'2}) - M_r \end{bmatrix}. \quad (2.22)$$

Multiplicando a equação acima pela matriz de rotação $R(\psi)$, obtém-se a equação de movimento no sistema de coordenadas inerciais:

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} R(\psi) \begin{bmatrix} a_{x'} \\ a_{y'} \\ \ddot{\psi} \end{bmatrix} = R(\psi) \begin{bmatrix} 2F_{x'1} + 2F_{x'2} - R_{x'} \\ -R_{y'} \\ 2c(F_{x'1} - F_{x'2}) - M_r \end{bmatrix}. \quad (2.23)$$

Substituindo (2.2) na equação acima e fazendo as devidas manipulações algébricas, obtém-se

$$M\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) = E(\mathbf{q})\boldsymbol{\tau}, \quad (2.24)$$

sendo

$$M = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix}, C(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \cos \psi R_{x'} - \sin \psi R_{y'} \\ \sin \psi R_{x'} + \cos \psi R_{y'} \\ M_r \end{bmatrix}, E(\mathbf{q}) = \begin{bmatrix} \frac{\cos \psi}{r} & \frac{\cos \psi}{r} \\ \frac{\sin \psi}{r} & \frac{\sin \psi}{r} \\ \frac{c}{r} & -\frac{c}{r} \end{bmatrix},$$

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_L \\ \tau_R \end{bmatrix} = \begin{bmatrix} \tau_1 + \tau_3 \\ \tau_2 + \tau_4 \end{bmatrix} = \begin{bmatrix} 2\tau_1 \\ 2\tau_2 \end{bmatrix} = \begin{bmatrix} 2\tau_3 \\ 2\tau_4 \end{bmatrix}, \quad \tau_i = rF_{x'i}.$$

Incluindo a restrição não-holonômica no modelo dinâmico (2.24), como introduzido em [54], tem-se

$$M\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) = E(\mathbf{q})\boldsymbol{\tau} + A^T(\mathbf{q})\boldsymbol{\lambda}, \quad (2.25)$$

sendo $\boldsymbol{\lambda}$ o vetor de multiplicadores de Lagrange correspondente à equação (2.8).

2.3 Formulação do problema

O modelo dinâmico mostrado na Seção 2.2 é descrito em espaço de estado na forma quase linear a parâmetros variantes (quase-LPV), ou seja, os parâmetros das matrizes do sistema dependem dos estados. Diferenciando (2.11) com relação ao tempo, tem-se

$$\ddot{\mathbf{q}} = \dot{S}(\mathbf{q})\dot{\boldsymbol{\eta}} + S(\mathbf{q})\ddot{\boldsymbol{\eta}}. \quad (2.26)$$

Multiplicando (2.24) por $S^T(\mathbf{q})$ à esquerda

$$S^T(\mathbf{q})M(\mathbf{q})\ddot{\mathbf{q}} + S^T(\mathbf{q})C(\mathbf{q}, \dot{\mathbf{q}}) = S^T(\mathbf{q})E(\mathbf{q})\boldsymbol{\tau} + S^T(\mathbf{q})A^T(\mathbf{q})\boldsymbol{\lambda}. \quad (2.27)$$

Note que nesta passagem o termo $A^T\boldsymbol{\lambda}$ é eliminado, pois $S^T(\mathbf{q})A^T = 0$ (as colunas de $S(\mathbf{q})$ estão no espaço nulo de $A(\mathbf{q})$).

Substituindo (2.26) na equação acima, tem-se

$$\overline{M}\dot{\boldsymbol{\eta}} + \overline{N}\boldsymbol{\eta} + \overline{C} = \overline{E}\boldsymbol{\tau}, \quad (2.28)$$

sendo

$$\begin{aligned} \overline{M} &= S^T(\mathbf{q})M(\mathbf{q})S(\mathbf{q}) = \begin{bmatrix} m & 0 \\ 0 & mx_{CIR}'^2 + I \end{bmatrix}, \\ \overline{N} &= S^T(\mathbf{q})M(\mathbf{q})\dot{S}(\mathbf{q}) = mx_{CIR}' \begin{bmatrix} 0 & -\dot{\psi} \\ \dot{\psi} & \dot{x}_{CIR}' \end{bmatrix}, \\ \overline{C} &= S^T(\mathbf{q})C(\mathbf{q}) = \begin{bmatrix} R_{x'} \\ M_r + x_{CIR}'R_{y'} \end{bmatrix} \mathbf{e} \\ \overline{E} &= S^T(\mathbf{q})E(\mathbf{q}) = \frac{1}{r} \begin{bmatrix} 1 & 1 \\ c & -c \end{bmatrix}. \end{aligned} \quad (2.29)$$

Isolando $\dot{\boldsymbol{\eta}}$ e acrescentado um distúrbio de torque $\mathbf{w} = [w_L \ w_R]^T$ em (2.28), obtém-se

$$\dot{\boldsymbol{\eta}} = -\overline{M}^{-1}\overline{N}\boldsymbol{\eta} - \overline{M}^{-1}\overline{C} + \overline{M}^{-1}\overline{E}\boldsymbol{\tau} + \overline{M}^{-1}\overline{E}\mathbf{w}. \quad (2.30)$$

Somando e subtraindo $\dot{\boldsymbol{\eta}}^d$ e $\overline{M}^{-1}\overline{N}\boldsymbol{\eta}^d$ e definindo o estado como

$$\tilde{\mathbf{x}} = \begin{bmatrix} v_{x'} - v_{x'}^d \\ \omega - \omega^d \\ x' - x'^d \\ \psi - \psi^d \end{bmatrix}, \quad (2.31)$$

a representação quase-LPV, em espaço de estado, para o controle de acompanhamento de traje-

tória do RMRD é dada por

$$\dot{\tilde{\mathbf{x}}} = \begin{bmatrix} A(\mathbf{q}) & 0 \\ I & 0 \end{bmatrix} \tilde{\mathbf{x}} + \begin{bmatrix} I \\ 0 \end{bmatrix} \mathbf{u} + \begin{bmatrix} B \\ 0 \end{bmatrix} \mathbf{w}, \quad (2.32)$$

sendo

$$A(\mathbf{q}) = -\overline{M}^{-1}\overline{N} = \begin{bmatrix} 0 & mx'_{CIR}\dot{\psi} \\ \frac{-mx'_{CIR}\dot{\psi}}{I+mx'^2_{CIR}} & -\frac{mx'_{CIR}\dot{x}'_{CIR}}{I+mx'^2_{CIR}} \end{bmatrix}, \quad (2.33)$$

$$B = \overline{M}^{-1}\overline{E} = \begin{bmatrix} \frac{1}{rm} & \frac{1}{rm} \\ \frac{c}{r(I+mx'^2_{CIR})} & -\frac{c}{r(I+mx'^2_{CIR})} \end{bmatrix}, \quad (2.34)$$

$$\mathbf{u} = -\dot{\boldsymbol{\eta}}^d + A(\mathbf{q})\boldsymbol{\eta}^d - \overline{M}^{-1}\overline{C} + B\boldsymbol{\tau}, \quad (2.35)$$

ou

$$\boldsymbol{\tau} = B^{-1}(\mathbf{u} + \dot{\boldsymbol{\eta}}^d - A(\mathbf{q})\boldsymbol{\eta}^d + \overline{M}^{-1}\overline{C}). \quad (2.36)$$

2.4 Controlador baseado na cinemática

O controlador baseado na cinemática proposto em [51], gera as velocidades desejadas para o problema de acompanhamento de trajetória de referência do RMRD. Considere o erro $q_e = [x_e \ y_e \ \psi_e]^T$, entre a posição de referência, $P_r = [x_r, y_r, \psi_r]^T$, e a posição real do RMR, $P_0 = [x, y, \psi]^T$, dada por

$$\begin{aligned} x_e &= \cos\psi(x_r - x) + \sin\psi(y_r - y), \\ y_e &= -\sin\psi(x_r - x) + \cos\psi(y_r - y), \\ \psi_e &= \psi_r - \psi, \end{aligned} \quad (2.37)$$

sendo $[x_r, y_r]^T = q_r$ a trajetória de referência escolhida e $\psi_r = \tan^{-1}(\dot{y}_r/\dot{x}_r)$. As velocidades linear, v_x^d , e angular, ω^d , desejadas de um RMRD são dadas por

$$\begin{aligned} v_x^d &= v_r \cos \psi_e + k_x x_e, \\ \omega^d &= \omega_r + v_r(k_y y_e + k_\psi \sin \psi_e), \end{aligned} \tag{2.38}$$

sendo k_x, k_y, k_ψ constantes definidas pelo projetista,

$$v_r = \sqrt{(\dot{x}_r)^2 + (\dot{y}_r)^2} \quad \text{e} \quad \omega_r = \dot{\psi}_r. \tag{2.39}$$

O modelo cinemático e dinâmico do robô móvel de rodas deslizantes apresentados neste capítulo serão utilizados na implementação da simulação do sistema de localização, de desvio de obstáculos e de coordenação de robôs móveis. O controlador baseado na cinemática é utilizado para os sistemas de desvio de obstáculos e de coordenação de robôs móveis.

Parte I

Localização de robôs móveis em ambientes externos

Capítulo 3

Fusão de dados de *encoder* e GPS usando filtragem

Neste capítulo apresenta-se o método para estimativa da posição de um robô móvel (RM) em ambientes externos baseado na fusão de dados usando filtros robustos.

A localização de um robô móvel é essencial em aplicações tais como exploração e procura e regaste. Os principais métodos utilizados para a localização estão baseados em receptores GPS e odometria. Neste trabalho utiliza-se como odometria os dados dos *encoders* das rodas. Os dados do receptor GPS são utilizados para melhorar a estimativa da posição baseada na odometria através da fusão de dados.

Assume-se que o plano onde vai se deslocar o robô é horizontal em relação à superfície da Terra no ponto p_0^e , o qual será considerado a origem do sistema de coordenadas geodésicas (veja Figura 3.1).

A posição do robô móvel no plano tangente é dado por

$$p^t = [x \ y \ h]^T, \quad (3.1)$$

sendo (x, y) as coordenadas do centro de massa do robô móvel (veja Figura 2.1) e h a altitude do robô móvel.

O problema de estimativa da posição de navegação será considerado em duas dimensões. Assim, a posição horizontal no plano tangente é definida por $p^t = [x \ y]^T$ e a velocidade no plano tangente é definida por $v^t = [\dot{x} \ \dot{y}]^T$. A única variável de orientação do robô móvel, ψ , é o ângulo

Figura 3.1: Sistemas de coordenadas retangulares ECEF (*Earth Centered - Earth Fixed*) e sistema de coordenadas geodésicas (Plano Tangente).

de guinada (rotação em torno do eixo vertical).

As entradas cinemáticas são a variação da contagem de pulsos dos *encoders* Δe_L e Δe_R , colocados nos eixos esquerdo e direito, respectivamente, durante o intervalo de tempo ΔT . O vetor de estado é $X = [x \ y \ \psi \ r_L \ r_R]^T$, onde r_L e r_R são os raios das rodas esquerda e direita, respectivamente.

A abordagem apresentada pode estender-se a superfícies planas não horizontais e a superfícies não planas suaves com sensores adicionais para medir os ângulos de rolagem (inclinação para frente ou para trás) e arfagem (inclinação para esquerda ou direita).

3.1 *Dead reckoning*

Dead reckoning é um procedimento para determinação da posição atual de um veículo através da análise das informações sobre sua posição anterior, então sua velocidade e curso conhecido. Baseia-se na idéia de que se o curso é conhecido, então podemos determinar a posição do veículo neste curso através da observação da velocidade de seu deslocamento.

A mais simples implementação de *dead reckoning* é conhecida como odometria, onde a posição do veículo ao longo do caminho é derivada diretamente de algum tipo do odômetro embarcado.

Uma forma de odometria bastante comum envolve codificadores ópticos diretamente acoplados aos eixos das rodas dos veículos.

A expressão “*Dead reckoning*” deriva da expressão “*Deduced reckoning*” (contagem deduzida).

3.1.1 *Encoder*

Os *encoders* medem a posição ou a velocidade das rodas ou motores; o princípio básico de funcionamento é o seguinte: um feixe de luz direcionado a um fotodetector é periodicamente interrompido por um padrão opaco/transparente codificado em um disco rotativo acoplado ao eixo a ser medido. Existem dois tipos de *encoders*, incremental e absoluto. O *encoder* absoluto mede a posição angular e infere a velocidade e o *encoder* incremental mede a velocidade rotacional e pode inferir a posição relativa à última medida. Neste trabalho, são usados *encoders* incrementais nos eixos das rodas do robô. Na maioria deles a saída é um número inteiro proporcional à rotação angular do eixo.

As saídas dos *encoders* são discretas, enquanto que, as rotações angulares das rodas são contínuas. Nesta seção, a modelagem dos sinais dos *encoders* e_L e e_R será contínua, pois esta representação é útil para o desenvolvimento do modelo de erro do sistema. Os sinais contínuos e_L e e_R quantizados serão os sinais de saída dos *encoders* $\tilde{e}_L$ e $\tilde{e}_R$, veja Figura 3.2. Portanto, os sinais de saída dos *encoders* são sinais discretos e serão utilizados na implementação das equações de navegação.

Figura 3.2: Exemplo de quantização de um *encoder* com resolução $C = 16$ ciclos por revolução.

Seja C a resolução do *encoder*, medida em ciclos por revolução. As variáveis θ_L e θ_R denotam a rotação angular em radianos das rodas esquerda e direita, respectivamente. A relação entre a

rotação angular e o sinal contínuo do *encoder* é dada por

$$\begin{aligned} e_L &= \frac{C}{2\pi} \theta_L(t), \\ e_R &= \frac{C}{2\pi} \theta_R(t), \end{aligned} \quad (3.2)$$

onde os sinais e_L e e_R são valores reais. A saída dos *encoders* são quantizados a valores inteiros no intervalo $[0, C - 1]$

$$\begin{aligned} \tilde{e}_L &= h(e_L), \\ \tilde{e}_R &= h(e_R). \end{aligned} \quad (3.3)$$

Derivando a equação (3.2) e resolvendo para $\dot{\theta}_L$ e $\dot{\theta}_R$ obtêm-se

$$\begin{aligned} \dot{\theta}_L &= \frac{2\pi}{C} \dot{e}_L(t), \\ \dot{\theta}_R &= \frac{2\pi}{C} \dot{e}_R(t). \end{aligned} \quad (3.4)$$

Substituindo a Equação (3.4) em (2.6) obtêm-se

$$\begin{bmatrix} v_{x'} \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r_L}{2} & \frac{r_R}{2} \\ \frac{r_L}{2c} & -\frac{r_R}{2c} \end{bmatrix} \begin{bmatrix} \frac{2\pi}{C} \dot{e}_L(t) \\ \frac{2\pi}{C} \dot{e}_R(t) \end{bmatrix}, \quad (3.5)$$

sendo $v_{x'}$ e ω as velocidades linear e angular do robô, respectivamente. Substituindo a Equação (3.5) em (2.14), (2.15) e (2.16) obtêm-se

$$\begin{aligned} \dot{x} &= \frac{1}{2} \frac{2\pi}{C} (r_L \dot{e}_L + r_R \dot{e}_R) \cos \psi - \frac{1}{2c} \frac{2\pi}{C} (r_L \dot{e}_L - r_R \dot{e}_R) x'_{CIR} \sin \psi + w_x, \\ \dot{y} &= \frac{1}{2} \frac{2\pi}{C} (r_L \dot{e}_L + r_R \dot{e}_R) \sin \psi + \frac{1}{2c} \frac{2\pi}{C} (r_L \dot{e}_L - r_R \dot{e}_R) x'_{CIR} \cos \psi + w_y, \\ \dot{\psi} &= \frac{1}{2c} \frac{2\pi}{C} (r_L \dot{e}_L - r_R \dot{e}_R) + w_\psi, \end{aligned} \quad (3.6)$$

onde os sinais w_x , w_y e w_ψ representam erros devidos à resolução do *encoder* e/ou escorregamento. Os modelos destes erros serão considerados como processos aleatórios independentes, isto é, não correlacionados.

Os comprimentos dos raios das rodas r_L e r_R variam em relação a seus valores nominais. Essa variação depende do tipo da roda, além de outros fatores imprevisíveis, como por exemplo

desgaste das rodas. Assim, o modelo do raio é dado por:

$$\begin{aligned} r_L &= r_0 + \delta r_L, \\ r_R &= r_0 + \delta r_R, \end{aligned} \quad (3.7)$$

onde r_0 é o valor nominal do raio da roda. O erro do raio da roda pode ser modelado como um processo escalar de Gauss-Markov ([11]) na forma

$$\begin{aligned} \delta \dot{r}_L &= -\lambda_R \delta r_L + w_L, \\ \delta \dot{r}_R &= -\lambda_R \delta r_R + w_R, \end{aligned} \quad (3.8)$$

onde $\lambda_R > 0$, $E \langle \delta r_L(0) \rangle = E \langle \delta r_R(0) \rangle = 0$, e

$$var(\delta r_R(0)) = var(\delta r_L(0)) = \bar{P}_R > 0. \quad (3.9)$$

O tempo de correlação $\frac{1}{\lambda_R}$ é selecionado como sendo um valor alto, por exemplo, 15 horas, pois a variação do raio da roda no tempo é lenta. As variáveis w_L e w_R são consideradas ruídos Gaussianos brancos com variância de $\sigma_R^2 = 2\lambda_R \bar{P}_R$.

3.2 Modelo de erro do sistema

O vetor de estado do erro é $\delta X = [\delta x, \delta y, \delta \psi, \delta r_L, \delta r_R]^T$, onde cada termo do erro é definido como $\delta X = X - \hat{X}$, sendo $\hat{X} = [\hat{x} \ \hat{y} \ \hat{\psi} \ \hat{r}_L \ \hat{r}_R]^T$ o vetor de estado estimado.

Expandindo a Equação (3.6) em uma série de Taylor com $X = \hat{X} + \delta X$ obtém-se as equações de estado do erro

$$\begin{aligned} \delta \dot{x} = & \frac{1}{2} \frac{2\pi}{C} (\delta r_L \dot{e}_L + \delta r_R \dot{e}_R) \cos \hat{\psi} - \frac{1}{2c} \frac{2\pi}{C} (\delta r_L \dot{e}_L - \delta r_R \dot{e}_R) x'_{CIR} \sin \hat{\psi} - \\ & \frac{1}{2} \frac{2\pi}{C} \left((\hat{r}_L \dot{e}_L + \hat{r}_R \dot{e}_R) \sin(\hat{\psi}) + \frac{(\hat{r}_L \dot{e}_L - \hat{r}_R \dot{e}_R)}{c} x'_{CIR} \cos(\hat{\psi}) \right) \delta \psi + w_x, \end{aligned} \quad (3.10)$$

$$\begin{aligned} \delta \dot{y} = & \frac{1}{2} \frac{2\pi}{C} (\delta r_L \dot{e}_L + \delta r_R \dot{e}_R) \sin \hat{\psi} + \frac{1}{2c} \frac{2\pi}{C} (\delta r_L \dot{e}_L - \delta r_R \dot{e}_R) x'_{CIR} \cos \hat{\psi} + \\ & \frac{1}{2} \frac{2\pi}{C} \left((\hat{r}_L \dot{e}_L + \hat{r}_R \dot{e}_R) \cos(\hat{\psi}) - \frac{(\hat{r}_L \dot{e}_L - \hat{r}_R \dot{e}_R)}{c} x'_{CIR} \sin(\hat{\psi}) \right) \delta \psi + w_y, \end{aligned} \quad (3.11)$$

$$\delta \dot{\psi} = \frac{1}{2c} \frac{2\pi}{C} (\delta r_L \dot{e}_L - \delta r_R \dot{e}_R) + w_\psi. \quad (3.12)$$

Definindo as seguintes variáveis para diminuir a complexidade das equações anteriores e

utilizando a Equação (3.5) tem-se:

$$\begin{aligned}
\hat{v}_L &= \hat{r}_L \dot{\theta}_L = \hat{r}_L \frac{2\pi}{C} \dot{e}_L, \\
\hat{v}_R &= \hat{r}_R \dot{\theta}_R = \hat{r}_R \frac{2\pi}{C} \dot{e}_R, \\
\hat{v} &= \frac{(\hat{v}_L + \hat{v}_R)}{2}, \\
\hat{\omega} &= \frac{(\hat{v}_L - \hat{v}_R)}{2c}.
\end{aligned} \tag{3.13}$$

As equações de estado do erro podem ser escritas na forma de espaço de estado como

$$\delta \dot{X}(t) = F(t)\delta X + G(t)w, \tag{3.14}$$

sendo

$$F(t) = \begin{bmatrix} 0 & 0 & -\hat{v} \sin \hat{\psi} - \hat{\omega} x'_{CIR} \cos \hat{\psi} & \frac{\hat{v}_L}{2\hat{r}_L} (\cos \hat{\psi} - \frac{x'_{CIR}}{c} \sin \hat{\psi}) & \frac{\hat{v}_R}{2\hat{r}_R} (\cos \hat{\psi} + \frac{x'_{CIR}}{c} \sin \hat{\psi}) \\ 0 & 0 & \hat{v} \cos \hat{\psi} - \hat{\omega} x'_{CIR} \sin \hat{\psi} & \frac{\hat{v}_L}{2\hat{r}_L} (\sin \hat{\psi} + \frac{x'_{CIR}}{c} \cos \hat{\psi}) & \frac{\hat{v}_R}{2\hat{r}_R} (\sin \hat{\psi} - \frac{x'_{CIR}}{c} \cos \hat{\psi}) \\ 0 & 0 & 0 & \frac{1}{2c} \frac{\hat{v}_L}{\hat{r}_L} & -\frac{1}{2c} \frac{\hat{v}_R}{\hat{r}_R} \\ 0 & 0 & 0 & -\lambda_R & 0 \\ 0 & 0 & 0 & 0 & -\lambda_R \end{bmatrix} \tag{3.15}$$

e $G(t) = I_5$, sendo $w = [w_x, w_y, w_\psi, w_L, w_R]^T$.

Sejam σ_1^2 , σ_2^2 e σ_3^2 as variâncias do ruído w_x , w_y e w_ψ , respectivamente. Os termos de ruído, w_x , w_y e w_ψ , incluídos no modelo são devido a escorregamentos ou superfícies não planas. O escorregamento das rodas é uma função da velocidade ou aceleração das rodas. Similarmente, o efeito não planar da superfície na posição estimada está em função da velocidade. Portanto, quando os ruídos foram devido a escorregamentos, os valores de σ_1^2 , σ_2^2 e σ_3^2 são definidos como funções crescentes da velocidade v e/ou da aceleração.

3.3 Modelo do GPS

Nesta seção apresentam-se a predição e as equações residuais da medida do GPS que serão usadas na fusão de dados.

O GPS (*Global Positioning System*) é um conjunto de 27 satélites localizados em planos inclinados com relação ao equador, 24 em operação e 3 extras caso haja falha nos outros. Cada

satélite transmite continuamente dados que indicam sua localização e o tempo (relógios atômicos). Quando um receptor GPS lê a transmissão de dois ou mais satélites, a diferença do tempo (ns) entre a transmissão e a recepção do sinal informa ao receptor sua distância relativa a cada satélite. São necessários 3 satélites para identificar a posição, e um satélite adicional permite que haja correção do erro no tempo, que ocorre devido ao receptor não ter um relógio de precisão.

A posição estimada do robô móvel no sistema de coordenadas ECEF (*Earth Centered - Earth Fixed*) é dada por $\hat{p}^e$ e a posição estimada do i -ésimo satélite no sistema de coordenadas ECEF é dado por $\hat{p}^i$, estas posições são calculadas a partir dos dados obtidos do receptor GPS, veja Anexo A. A distância estimada calculada do robô móvel ao i -ésimo satélite é dada por

$$\hat{R}(\hat{p}^e, \hat{p}^i) = \|\hat{p}^e - \hat{p}^i\|_2, \quad (3.16)$$

sendo

$$\hat{p}^e = p_0^e + R_t^e \hat{p}^t, \quad (3.17)$$

sendo p_0^e a origem do sistema de coordenadas geodésicas, veja Figura 3.1, $\hat{p}^t = [\hat{x} \ \hat{y} \ \hat{h}_r]^T$ a posição estimada do robô móvel no plano tangente e R_t^e a matriz de rotação constante do sistema de coordenadas geodésicas (plano tangente) para o sistema de coordenadas ECEF, e é dada por

$$R_t^e = (R_e^t)^T = \begin{bmatrix} -\sin(\phi) \cos(\lambda) & -\sin(\phi) \sin(\lambda) & \cos(\phi) \\ -\sin(\lambda) & \cos(\lambda) & 0 \\ -\cos(\phi) \cos(\lambda) & -\cos(\phi) \sin(\lambda) & -\sin(\phi) \end{bmatrix}^T. \quad (3.18)$$

3.3.1 Pseudodistância do GPS

A pseudodistância do GPS é resultado do tempo transcorrido entre a transmissão de um satélite e o tempo que leva para a transmissão chegar ao receptor, e que inclui os erros dos relógios do satélite e do receptor, veja Figura 3.3.

A pseudodistância da posição p ao i -ésimo satélite é dada por:

$$\rho(p, p^i) = \|p - p^i\|_2 + c\Delta t_r, \quad (3.19)$$

sendo Δt_r o erro do relógio do receptor em relação à medida do tempo do GPS e c a velocidade da luz. Adicionalmente existem erros que são devidos à dispersão atmosférica, aos diversos caminhos do sinal e ao ruído aleatório no sinal. Em [11] encontra-se uma descrição detalhada do processo

Figura 3.3: Pseudodistância da posição p ao i -ésimo satélite.

de medida destes erros. A posição p é estimada com base em medidas da pseudodistância, cujo modelo para o serviço de posicionamento padrão (SPS), transmissão na frequência L1 ($f_1 = 1575,42$ MHz) em código C/A (*course acquisition*), é dado por

$$\begin{aligned}
 \tilde{\rho}_r^i &= \rho(p, p^i) + c\delta t^i + \frac{f_2}{f_1}I_r^i + T_r^i + M_\rho^i + \nu_\rho^i \\
 &= \rho(p, \hat{p}^i) + (\rho(p, p^i) - \rho(p, \hat{p}^i)) + c\delta t^i + \frac{f_2}{f_1}I_r^i + T_r^i + M_\rho^i + \nu_\rho^i \\
 &= \rho(p, \hat{p}^i) + E^i + c\delta t^i + \frac{f_2}{f_1}I_r^i + T_r^i + M_\rho^i + \nu_\rho^i \\
 &= R(p, \hat{p}^i) + c\Delta t_r + E^i + c\delta t^i + \frac{f_2}{f_1}I_r^i + T_r^i + M_\rho^i + \nu_\rho^i,
 \end{aligned} \tag{3.20}$$

sendo c a velocidade da luz, $f_2 = 1227,60$ MHz a frequência da portadora L2, E^i o erro devido ao cálculo da posição no processo de solução no i -ésimo satélite. Os subíndices r e ρ estão relacionados ao receptor. As componentes do erro na medida são definidas como:

- $\delta t^i \rightarrow$ erro residual do relógio do satélite,
- $I_r^i \rightarrow$ erro devido aos efeitos de dispersão da atmosfera,
- $T_r^i \rightarrow$ erro devido aos efeitos de não dispersão da atmosfera,
- $M_\rho^i \rightarrow$ erro devido aos diversos caminhos do sinal,
- $\nu_\rho^i \rightarrow$ ruído aleatório na medida da pseudodistância.

A combinação dos erros na medida da pseudodistância pode ser escrita ([11]) como

$$\chi^i = E^i + c\delta t^i + \frac{f_2}{f_1}I_r^i + T_r^i + M_\rho^i + \nu_\rho^i, \quad (3.21)$$

onde χ^i é chamado UERE (*User Equivalent Range Error*) ou URE (*User Range Error*). Assumindo que os erros dos diferentes satélites são independentes,

$$\text{var}(\chi) = \sigma^2 I, \quad (3.22)$$

onde o valor de σ para L1 C/A está entre os valores 8 e 11 metros, o valor de χ^i pode ser obtido, por exemplo, com a correção da medida. Se não existe informação disponível sobre os erros na medida da pseudodistância, então $\chi^i = 0$ e pode-se usar métodos, tais como o Estado Aumentado [56], para adaptar os erros correlacionados no tempo do GPS.

3.3.2 Modelo da diferença da medida

Nesta abordagem, a medida obtida de um satélite é subtraída das medidas de todos os outros satélites para eliminar o erro do relógio do receptor $c\Delta t_r$. Assumindo que m satélites estão disponíveis e as medidas obtidas do primeiro satélite são subtraídas de todos os outros satélites, a diferença das medidas da pseudodistância pode ser colocada como

$$\nabla \hat{\rho} = \begin{bmatrix} \left(\hat{R}(\hat{p}, \hat{p}^2) + \hat{\chi}^2 \right) - \left(\hat{R}(\hat{p}, \hat{p}^1) + \hat{\chi}^1 \right) \\ \left(\hat{R}(\hat{p}, \hat{p}^3) + \hat{\chi}^3 \right) - \left(\hat{R}(\hat{p}, \hat{p}^1) + \hat{\chi}^1 \right) \\ \vdots \\ \left(\hat{R}(\hat{p}, \hat{p}^m) + \hat{\chi}^m \right) - \left(\hat{R}(\hat{p}, \hat{p}^1) + \hat{\chi}^1 \right) \end{bmatrix}. \quad (3.23)$$

As medidas são assumidas como simultâneas. As medidas residuais para i -ésimo satélite são dadas por

$$\begin{aligned} \delta \rho^i(t) &= \tilde{\rho}^i - \hat{\rho}^i \\ &= (\tilde{\rho}^i - \tilde{\rho}^1) - (\hat{\rho}^i - \tilde{\rho}^1) \\ &= (\tilde{\rho}^i - \tilde{\rho}^1) - \nabla \hat{\rho}^i. \end{aligned} \quad (3.24)$$

Colocando a Equação (3.24) para $i = 2, \dots, m$ como

$$\delta\rho(t) = \begin{bmatrix} (\tilde{\rho}^2 - \tilde{\rho}^1) \\ (\tilde{\rho}^3 - \tilde{\rho}^1) \\ \vdots \\ (\tilde{\rho}^m - \tilde{\rho}^1) \end{bmatrix} - \nabla \hat{\rho}, \quad (3.25)$$

o modelo do erro para a medida residual é dado por

$$\delta\rho(t) = H\delta x + \boldsymbol{\eta}, \quad (3.26)$$

$$H = \begin{bmatrix} \bar{h}_g^2 \\ \bar{h}_g^3 \\ \vdots \\ \bar{h}_g^m \end{bmatrix} \quad \boldsymbol{\eta} = \begin{bmatrix} (\eta^2 - \eta^1) \\ (\eta^3 - \eta^1) \\ \vdots \\ (\eta^m - \eta^1) \end{bmatrix}$$

onde η^i contém o erro χ^i e $\bar{h}_g^i = h_g^i - h_g^1$. A matriz de medida para o i -ésimo satélite é descrito como

$$h_g^i = [\vec{h}_1^t, \vec{h}_2^t, 0, 0, 0], \quad (3.27)$$

sendo $\vec{h}^t = \vec{h}^e R_e^t$, os termos $\vec{h}_1^t$ e $\vec{h}_2^t$ são as duas primeiras componentes de $\vec{h}^t$, e $\vec{h}^e = \frac{(\hat{p}^e - \hat{p}^i)}{\|\hat{p}^e - \hat{p}^i\|_2}$.

A diferença das medidas incrementa a quantidade de ruído em virtude de cada medida, e causa um ruído adicional sobre a medida residual para os diversos satélites a serem correlacionados. Considera-se a variância do ruído de medida para a medida da pseudodistância residual da Equação (3.26) como

$$\text{var}(\eta^i - \eta^1) = E\langle (\eta^i - \eta^1)(\eta^i - \eta^1) \rangle = 2\sigma_p^2, \quad (3.28)$$

$$\text{var}((\eta^2 - \eta^1)(\eta^3 - \eta^1)) = E\langle (\eta^2 - \eta^1)(\eta^3 - \eta^1) \rangle = \sigma_p^2. \quad (3.29)$$

Portanto, quando usamos o modelo da diferença da medida, a matriz de covariância da medida é não diagonal. Assumindo que a covariância do ruído é a mesma para todos os satélites, a matriz

de covariância da medida é dada por

$$R = \begin{bmatrix} 2 & 1 & 1 & \dots & 1 \\ 1 & 2 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 2 \end{bmatrix} \sigma_\rho^2, \quad (3.30)$$

onde $R \in \mathbb{R}^{(m-1) \times (m-1)}$, sendo m o número de satélites.

3.4 Fusão de dados usando filtro de Kalman robusto

A fusão de dados é definida através do uso de técnicas que combinam dados de vários sensores e reúne essas informações a fim de obter estados do sistema, que serão mais precisos do que se fossem obtidos através de um único sensor. Filtragem é o método utilizado para combinar as medidas de vários sensores. O filtro mais utilizado para fusão de dados é o filtro de Kalman. Neste trabalho, utiliza-se filtro de Kalman estendido e filtro de Kalman robusto. Este filtro de Kalman robusto é utilizado para diminuir os erros da estimativa devido as incertezas do sistema.

A Figura 3.4 mostra o diagrama de blocos da estimação da posição utilizando a fusão de dados do *encoder* e do receptor GPS. A odometria do robô móvel é utilizada para a predição da próxima posição do RM (*Dead Reckoning*), e os dados do receptor GPS são utilizados para corrigir a predição para obter uma melhor estimativa da posição do robô móvel.

Figura 3.4: Esquema da fusão de dados.

Com as equações do sistema de navegação do robô dadas por (3.14), (3.24) e (3.26), nesta seção apresentam-se as equações e o procedimento para o uso do Filtro de Kalman estendido [57; 11] e do filtro de Kalman robusto *Bounded Data Uncertainties* (BDU) [58].

Considere o seguinte sistema não linear:

$$\begin{aligned}\dot{X} &= f(X, t) \\ Z &= h(X, t)\end{aligned}\tag{3.31}$$

Define-se $X^*(t)$ o ponto de operação que satisfaz

$$\begin{aligned}\dot{X}^*(t) &= f(X^*, t), \\ Z^*(t) &= h(X^*, t).\end{aligned}$$

Linearizando o sistema em torno a $X^*(t)$ obtém-se

$$\begin{aligned}\dot{X} &\approx \dot{X}^*(t) + \left. \frac{\partial f(X, t)}{\partial X} \right|_{X^*} \delta X + w(t) \rightarrow \delta \dot{X} = F(t) \delta X + w(t), \\ Z &\approx Z^*(t) + \left. \frac{\partial h(X, t)}{\partial X} \right|_{X^*} \delta X + \eta(t) \rightarrow \delta Z = H(t) \delta X + \eta(t),\end{aligned}\tag{3.32}$$

sendo $\delta X = X - X^*$ e $\delta Z = Z - Z^*$. Discretizando o sistema anterior (3.32), obtém-se o modelo do erro do sistema

$$x_{k+1} = F_k x_k + w_k,\tag{3.33}$$

$$z_k = H_k x_k + \eta_k,\tag{3.34}$$

sendo $x = \delta X$ e $z = \delta Z$. A Equação (3.33) representa o modelo do erro do sistema baseado no modelo cinemático do robô móvel (3.14), esta equação utiliza os dados do *encoder* para calcular a próxima posição (predição). A Equação (3.34) representa o modelo da diferença da medida do receptor GPS, dados dos n satélites disponíveis, para corrigir a posição calculada pela Equação (3.33). A matriz F_k e a matriz da covariância do ruído Q_k são calculadas como:

$$\begin{aligned}F_k &= e^{F(t_k) \Delta t_k}, \\ Q_k &\approx Q \Delta t_k + (F(t_k) Q + Q F(t_k)^T) \frac{\Delta t_k^2}{2} + 2F(t_k) Q F(t_k)^T \frac{\Delta t_k^3}{6}.\end{aligned}$$

sendo $\Delta t_k = t_{k+1} - t_k$. Introduz-se as seguintes variáveis para o estado predito e filtrado:

$$\begin{aligned} x_{k|k-1} &\equiv \text{Estado predito dado por } \{z_0, z_1, \dots, z_{k-1}\}, \\ x_{k|k} &\equiv \text{Estado filtrado dado por } \{z_0, z_1, \dots, z_{k-1}, z_k\}. \end{aligned}$$

3.4.1 Filtro de Kalman estendido

O filtro de Kalman estendido é uma modificação do algoritmo do filtro de Kalman para aplicação em sistemas não lineares, veja [57; 11].

A seguir apresenta-se o algoritmo recursivo:

Passo 0 (Condições iniciais): Faça

$$\begin{aligned} \hat{x}_{0|0} &= 0, \\ P_{0|0} &= \text{var}(X_0^*). \end{aligned} \tag{3.35}$$

Passo 1: Propagando no tempo

$$\hat{x}_{k+1|k} = F_k \hat{x}_{k|k}, \tag{3.36}$$

$$P_{k+1|k} = F_k P_{k|k} F_k^T + Q_k. \tag{3.37}$$

Passo 2: Calcular a matriz de ganho do filtro e atualização da medida, usando a Equação (3.24) para a medida residual

$$K_{k+1} = P_{k+1|k} H_{k+1}^T (H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1})^{-1}, \tag{3.38}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} (z_k - H_{k+1} \hat{x}_{k+1|k}), \tag{3.39}$$

$$P_{k+1|k+1} = (I - K_{k+1} H_{k+1}) P_{k+1|k}. \tag{3.40}$$

Passo 3: Atualização do estado estimado $\hat{X}_{k+1|k+1}$

$$\begin{aligned} \hat{X}_{k+1|k}^* &= \hat{X}_{k|k} + f(\hat{X}_{k|k}, t_k) \Delta t_k, \\ \hat{X}_{k+1|k+1} &= \hat{X}_{k+1|k}^* + \hat{x}_{k+1|k+1}. \end{aligned} \tag{3.41}$$

Portanto, a posição estimada do robô móvel obtém-se do estado estimado $\hat{X}_{k+1|k+1} = [\hat{x} \ \hat{y} \ \hat{\psi} \ \hat{r}_L \ \hat{r}_R]^T$, o qual foi calculado no algoritmo do filtro de Kalman estendido utilizando os

dados do *encoder* e do receptor GPS.

3.4.2 Filtro de Kalman robusto BDU

Este filtro considera as incertezas de modelagem, veja [58]. Essas incertezas estruturadas afetam as matrizes F_k e H_k . O sistema (3.33) pode ser reescrito na seguinte forma

$$\begin{aligned} x_{k+1} &= (F_k + \delta F_k)x_k + w_k, \\ z_k &= (H_k + \delta H_k)x_k + \eta_k. \end{aligned} \quad (3.42)$$

As perturbações são assumidas como sendo:

$$\delta F_k = M_{f,k} \Delta_k N_{f,k}; \quad \delta H_k = M_{h,k} \Delta_k N_{h,k}; \quad \|\Delta_k\| \leq 1,$$

e $M_{f,k}$, $M_{h,k}$, $N_{f,k}$, $N_{h,k}$ são matrizes definidas a priori pelo projetista, Δ_k é uma matriz limitada como norma menor que 1 e arbitrária.

A seguir apresenta-se o algoritmo recursivo proposto em [58]:

Passo 0 (Condições iniciais): Calcule-se o parâmetro preditor $\hat{\lambda}_{-1} = (1 + \alpha) \|M_{h,0}^T R_0^{-1} M_{h,0}\|$

$$\begin{aligned} \hat{R}_0 &= R_0 - \hat{\lambda}_{-1}^{-1} M_{h,0} M_{h,0}^T, \\ P_{0|0} &= \left(P_0^{-1} + H_0^T \hat{R}_0^{-1} H_0 + \hat{\lambda}_{-1} N_{h,0}^T N_{h,0} \right)^{-1}, \\ x_{0|0} &= 0, \end{aligned} \quad (3.43)$$

Passo 1: Calcule-se o parâmetro $\hat{\lambda}_k$ como

$$\hat{\lambda}_k := (1 + \alpha) \|diag\{M_{f,k}^T Q_k^{-1} M_{f,k}, M_{h,k+1}^T R_{k+1}^{-1} M_{h,k+1}\}\|, \quad (3.44)$$

sendo $\alpha > 0$.

Passo 2: Substitua os parâmetros Q_k , R_{k+1} , $P_{k|k}$ pelos parâmetros corrigidos

$$\begin{aligned} \hat{Q}_k &= Q_k - \hat{\lambda}_k^{-1} M_{f,k} M_{f,k}^T, \\ \hat{R}_{k+1} &= R_{k+1} - \hat{\lambda}_k^{-1} M_{h,k+1} M_{h,k+1}^T, \\ \hat{P}_{k|k} &= \left(P_{k|k}^{-1} - \hat{\lambda}_k N_{f,k}^T N_{f,k} \right)^{-1}. \end{aligned} \quad (3.45)$$

Passo 3: Atualize $\{P_{k|k}, \hat{x}_{k|k}\}$ com $\{P_{k+1|k+1}, \hat{x}_{k+1|k+1}\}$ da seguinte forma

$$\begin{aligned} P_{k+1|k+1} &= \left(\left(\hat{Q}_k + F_k \hat{P}_{k|k} F_k^T \right)^{-1} + H_{k+1}^T \hat{R}_{k+1}^{-1} H_{k+1} + \hat{\lambda}_k N_{h,k}^T N_{h,k} \right)^{-1}, \\ \hat{x}_{k+1|k+1} &= P_{k+1|k+1} \left(\left(\hat{Q}_k + F_k \hat{P}_{k|k} F_k^T \right)^{-1} F_k (I - \hat{\lambda}_k \hat{P}_{k|k} N_{f,k}^T N_{f,k}) \right) \hat{x}_{k|k} \\ &\quad + P_{k+1|k+1} H_{k+1}^T \hat{R}_{k+1}^{-1} z_{k+1}. \end{aligned} \quad (3.46)$$

Passo 4: Atualização do estado estimado $\hat{X}_{k+1|k+1}$

$$\begin{aligned} \hat{X}_{k+1|k}^* &= \hat{X}_{k|k} + f(\hat{X}_{k|k}, t_k) \Delta t_k, \\ \hat{X}_{k+1|k+1} &= \hat{X}_{k+1|k}^* + \hat{x}_{k+1|k+1}. \end{aligned} \quad (3.47)$$

Portanto, a posição estimada do robô móvel obtém-se do estado estimado $\hat{X}_{k+1|k+1} = [\hat{x} \ \hat{y} \ \hat{\psi} \ \hat{r}_L \ \hat{r}_R]^T$, o qual foi calculado no algoritmo do filtro de Kalman robusto BDU utilizando os dados do *encoder* e do receptor GPS.

Neste capítulo foram apresentadas as equações em espaço de estado do sistema utilizando os dados do receptor GPS e dos *encoders*. Também, foram apresentados métodos para a fusão de dados, para obter uma melhor estimativa da posição do robô móvel.

Capítulo 4

Resultados

Neste capítulo, são apresentados os resultados obtidos via simulação e implementação da fusão de dados descrita no Capítulo 3, considerando o robô móvel descrito no Capítulo 2. Os parâmetros nominais do robô móvel *Pioneer 3-AT* obtidos do *datasheet* do motor do robô e da dissertação de mestrado [59], são mostrados na Tabela 4.1.

Tabela 4.1: Parâmetros do RMRD *Pioneer 3-AT*.

	Parâmetro	Valor	Unidade
Motor	k_T	0,0230	Nm/A
	k_E	0,0230	V s/rad
	n	38,3	
	R	0,71	Ω
Robô	I	0,413	Kgm ²
	m	40	Kg
	f_r	0,043	
	μ	0,506	
	x_{CIR}	0,008	m
	a	0,138	m
	b	0,122	m
	c	0,1975	m
	r	0,22	m
	λ_R	1/36000	Hz
	$\bar{P}_R$	10^{-4}	m^2

Figura 4.1: Robô móvel de rodas deslizantes, *Pioneer 3AT*.

4.1 Simulação computacional da fusão de dados do *encoder* e do receptor GPS

Esta simulação foi realizada com o objetivo de comparar os filtros usados na fusão de dados do *encoder* e do receptor GPS, apresentados na seção anterior.

Considera-se que o ambiente externo é um plano horizontal e livre de obstáculos, sendo a origem do plano o estacionamento do Departamento de Elétrica da ESSC-USP. As medidas do GPS são obtidas a cada 1 segundo, ($T = 1\text{ s}$) e os dados do GPS são calculados usando a equação (3.20). Os valores da velocidade linear $\hat{v}$ e da velocidade angular $\hat{\omega}$ são calculados a partir dos dados obtidos das saídas do *encoders* $\tilde{e}_L$ e $\tilde{e}_R$ do RMRD *Pioneer 3-AT* (equação (3.13)).

Os valores da latitude e longitude da origem de coordenadas geodésicas são $\phi = -22.005687^\circ$ e $\lambda = -47.898087^\circ$, respectivamente. Conhecendo estes valores e substituindo na Equação (A.1), encontra-se a posição ECEF $p_0^e = [3969365.292713 \quad -4392688.767848 \quad -2376679.922817]^T$. Nessa simulação foram utilizados $m = 4$ satélites, com as seguintes posições ECEF e UREs:

$$\begin{aligned} \hat{p}^1 &= [-5446876.120880 \quad 15818876.879613 \quad 20627070.888486]^T m; & \hat{\chi}^1 &= -3.321394 m; \\ \hat{p}^2 &= [-17721027.431238 \quad 1550389.006265 \quad 19724609.990632]^T m; & \hat{\chi}^2 &= 3.917814 m; \\ \hat{p}^3 &= [-7517837.450445 \quad -16122054.439745 \quad 19724609.990632]^T m; & \hat{\chi}^3 &= 3.780262 m; \\ \hat{p}^4 &= [15988394.618051 \quad -7798061.094074 \quad 19724609.990632]^T m; & \hat{\chi}^4 &= -0.122195 m. \end{aligned}$$

A matriz de covariância do ruído na medida η é dada por $R = (0, 3)^2 I_3$, a matriz inicial de covariância do estado do erro δx é dada por

$$P = \text{diag}([1 \quad 1 \quad 0, 1^2 \quad 0, 1\bar{P}_R \quad 0, 1\bar{P}_R]),$$

e a matriz de covariância do ruído no estado w é dada por

$$Q(t) = \text{diag} \left(\left[(0,001v(t))^2, (0,001v(t))^2, (0,001v(t))^2, 2\bar{P}_R\lambda_R, 2\bar{P}_R\lambda_R \right] \right),$$

onde valores de $\bar{P}_R$ (equação (3.8)) e λ_R (equação (3.9)), estão definidos na Tabela 4.1.

A localização do RMRD usando-se apenas os dados do GPS é apresentada na Figura 4.2, e foi mostrada em figura separada pois tem um deslocamento da origem da trajetória devido aos erros do receptor GPS. A localização do RMRD com fusão de dados é apresentada na Figura 4.3(a) e o ângulo de orientação do robô na Figura 4.3(b). Na Figura 4.4 apresenta-se o erro da estimativa da posição do robô móvel.

Figura 4.2: Posição do robô móvel usando GPS.

Figura 4.3: Posição real e estimada usando fusão de dados do GPS e do *encoder*.

Observa-se pelos resultados gráficos que a estimativa para a trajetória real do robô móvel é

melhor obtida através da fusão de dados do *encoder* e do receptor GPS, do que utilizando-se apenas dados do receptor GPS ou só odometria. A Figura 4.4 mostra que os erros de estimação da posição através de fusão de dados resultam em valores menores que 10 cm. Portanto, a fusão de dados do *encoder* e do receptor GPS melhora a estimação da posição do veículo.

Figura 4.4: Erro da estimativa da posição usando fusão de dados do GPS e do *encoder*.

Figura 4.5: Variância dos filtros usados para a fusão de dados do GPS e do *encoder*.

A comparação entre o filtros EKF e o filtro robusto BDU utilizados para a fusão de dados foi realizada calculando-se a variância do erro de estimativa (veja Figura 4.5 em dB) utilizando a seguinte aproximação

$$\varepsilon \|x_i - \hat{x}_i\| \approx \frac{1}{T} \sum_{j=1}^T \|x_i^{(j)} - \hat{x}_i^{(j)}\| \quad (4.1)$$

onde $\|\cdot\|$ é a norma euclidiana. Cada ponto no instante i em cada curva da variância é uma

média calculada sobre $T = 50$ experimentos.

Observa-se que o filtro EKF apresenta uma variância maior do erro de estimativa se comparado com o filtro robusto BDU considerado neste trabalho.

4.2 Resultados experimentais da fusão de dados do *encoder* e do receptor GPS

Nesta seção, apresentam-se os resultados experimentais utilizando a fusão de dados da odometria do RMRD *Pioneer 3-AT* e do receptor GPS de precisão de $3,3\text{ m}$ (especificações na Figura A.2). Os dados da odometria e do receptor GPS foram salvos para posteriormente serem processados. Utilizou-se como referência os dados fornecidos pelo *Google Earth*® do estacionamento do Departamento de Elétrica da ESSC-USP para a localização do RMRD, veja Figura 4.6.

A trajetória real percorrida pelo robô móvel (linha branca) é mostrada na Figura 4.6, e foi realizada utilizando as formas do terreno de fácil identificação no *Google Earth*®. O robô móvel *Pioneer 3-AT* possui um algoritmo interno para a estimação da posição usando o *encoder* e o giroscópio do robô, os dados obtidos da posição (linha amarela) e a trajetória estimada usando o modelo cinemático e os dados do *encoder* (linha azul clara) são mostrados nesta figura. Os filtros utilizados para fusão de dados do receptor GPS e do *encoder*, EKF (linha vermelha) e robusto BDU (linha verde), tem resultados similares, portanto na figura as duas trajetórias aparecem sobrepostas, e a linha roxa é a trajetória estimada utilizando só os dados do receptor GPS.

Como era de se esperar pelos resultados obtidos na simulação da seção anterior, a estimação da trajetória real do robô móvel obtida através da fusão de dados do *encoder* e do receptor GPS é mais exata do que utilizando-se apenas dados do receptor GPS ou só da odometria. Portanto, a fusão de dados do *encoder* e do receptor GPS melhora a estimação da posição do veículo, neste caso do robô móvel.

Nesta seção não foi possível mostrar o estudo comparativo da variância, visto que, não foi possível obter a trajetória real do robô.

(a) Posição do robô móvel.

(b) Ângulo de orientação do robô móvel.

Figura 4.6: Posição real e estimada usando fusão de dados do GPS e do *encoder*.

Parte II

Algoritmo para desvio de obstáculos

Capítulo 5

Algoritmo para desvio de obstáculos

Neste capítulo são apresentados dois algoritmos para desvio de obstáculos em ambientes desconhecidos, *Tangent Bug* e E^* . Estes algoritmos foram selecionados por apresentarem vantagens distintas para a navegação em ambientes desconhecidos que serão apresentadas neste capítulo. O algoritmo *Tangent Bug* foi escolhido por explorar os dados de distância obtidos pelo sensor laser e por ser de fácil implementação, apresentando uma convergência satisfatória ao alvo que em muitos casos resulta próximo ao caminho ótimo. O algoritmo E^* também utiliza dados de distância obtidos pelo sensor laser para encontrar o caminho ótimo ao alvo e permite replanejar a trajetória na presença de obstáculos móveis. Dependendo das informações do ambiente um deles será utilizado, no Capítulo 6 apresentam-se dois ambientes para utilizar estes algoritmos.

5.1 Algoritmo *Tangent Bug*

O algoritmo *Tangent Bug* orienta a navegação de um robô móvel em um ambiente desconhecido plano, o qual contém obstáculos estacionários. Foi apresentado em [26] como uma otimização do algoritmo VisBug [25] e assim como os outros algoritmos *Bug* [60], estão baseados em dois comportamentos básicos, “movimento rumo ao alvo” e “acompanhamento da borda do obstáculo”, os quais dependem da distância do robô ao alvo $d(p, T)$, veja Figura 5.1. Em [61], mais detalhes de sistemas baseados em comportamento são apresentados. No comportamento “movimento ao alvo”, a $d(p, T)$ sempre diminui e no comportamento “acompanhamento da borda do obstáculo” tenta-se escapar de um mínimo de $d(p, T)$. Em cada passo o robô constrói um *Local Tangent Graph* (LTG) baseado em medidas de distância, obtidas por sensores tais como, laser, sonares, etc., para decidir a próxima ação. O LTG é um *Tangent Graph* (TG) que inclui

somente as partes dos obstáculos que estão dentro do máximo alcance de medição dos sensores. O TG em ambientes poligonais foi apresentado em [25], sendo definido por uma porção do gráfico de visibilidade cujos vértices são os vértices convexos dos obstáculos (O_i) incluindo a posição do robô e a posição (p) do alvo (T), e as bordas do TG são as linhas que conectam os vértices. Um exemplo de LTG para um sensor laser de varredura de 180° é mostrado na Figura 5.1. Em [62], demonstra-se que o algoritmo *Tangent Bug* possui uma convergência global ao alvo.

Figura 5.1: O LTG usando um sensor laser de varredura de 180° .

A seguir, apresenta-se o algoritmo *Tangent Bug*:

Passo 0 (Condições Iniciais): Posição inicial do robô p_0 e posição do alvo (T)

$$p_0 = [x_0, y_0, \psi_0],$$

$$T = [x_T, y_T, \psi_T].$$

Passo 1: Movimentar em direção ao ponto $n \in \{T, O_i\}$, sendo $\{T, O_i\}$ os vértices do atual LTG, tal que n minimiza $d = d(p, n) + d(n, T)$ (direção ótima local), sendo $d(p, n) > d(n, T)$. Fazer até que um dos seguintes eventos aconteça:

- O alvo é alcançado. Então, parar o robô.
- O robô detecta um mínimo local, isto é, a direção que minimiza $d = d(p, n) + d(n, T)$ começa a incrementar $d(p, T)$. Então, ir para o passo 2.

Passo 2: Escolher a direção de acompanhamento da borda do obstáculo, esquerda ou direita, e movimentar sobre a borda, atualizando continuamente o LTG, a $d_{percorrida}(T)$ que é a mínima distância entre a borda que tem sido detectada e o alvo, e a $d_{alcance}(T)$ que é a mínima distância entre o alvo e o ponto qualquer $n \in \{T, O_i\}$ dentro do ambiente visível. Fazer até que um dos seguintes eventos aconteça:

- O alvo é alcançado. Então, parar o robô.
- Cumpra a condição para trocar de comportamento que é $d_{alcance}(T) < d_{percorrida}(T)$. Então, ir para o passo 1.
- O robô completa uma volta ao redor do obstáculo, nesse caso o alvo não pode ser alcançado. Então, parar o robô.

5.1.1 Considerações na implementação do algoritmo *Tangent Bug*

Na implementação do algoritmo *Tangent Bug* os seguintes aspectos devem ser considerados:

1. No comportamento “movimento ao alvo”, a trajetória planejada ao ponto $n \in \{T, O_i\}$ será realizada utilizando o método de interpolação *Cubic Spline* [63; 64], para gerar trajetórias suaves. As vantagens deste método são que os caminhos possuem curvaturas mínimas e são simples de gerar em tempo real [62].
2. No comportamento “acompanhamento da borda do obstáculo”, temos a dificuldade de que as bordas dos obstáculos não são conhecidas. Quando o robô detecta um obstáculo, a superfície normal à borda do obstáculo é a direção do ponto de contato ao centro do robô, veja Figura 5.2. Utilizando esta informação, o robô pode determinar a construção de uma série de tangentes para realizar o acompanhamento da borda do obstáculo. Seja $d(p, x)$ o conjunto de todas as medidas obtidas pelo sensor laser de varredura de 180° e d_{Lim} a distância limite que o robô pode se aproximar do obstáculo, a borda mais próxima do robô é dada por

$$D(x) = \min_{x \in \partial LTG} d(p, x).$$

Figura 5.2: Método para o acompanhamento da borda de um obstáculo.

Tomando o gradiente de $D(x)$, $\nabla D(x)$ obtém-se a direção relacionada ao menor valor de $d(p, x)$. Portanto, $\pm(\nabla D(x))^\perp$ é a tangente à borda do obstáculo, veja Figura 5.2, sendo que o sinal $\pm$ indica se o acompanhamento será pela direita ou pela esquerda.

5.2 Algoritmo E *

O algoritmo E* é um método de planejamento de trajetória com capacidade de replanejamento dinâmico e com interpolação configurável do custo da trajetória, isso resulta em caminhos mais apropriados durante a descida do gradiente. O algoritmo E* foi apresentado em [65] como uma solução do problema da trajetória não ótima gerada pelos algoritmos NF1, A* ou D*, veja [21]. Estes algoritmos calculam uma função de navegação restrita a escolhas de movimento ao longo das bordas dos grafos ou transições discretas entre as células do mapa, que, assim, produzem caminhos que não são ótimos para a execução.

A formulação do algoritmo E* está baseada no domínio contínuo, independente de qualquer representação específica do espaço de configuração $\mathcal{C}$ (em [21] mais detalhes de espaço de configurações). Considere uma frente de onda que se propaga desde o alvo por todo o ambiente, veja Figura 5.3. Ao avaliar a evolução da frente de onda, obtém-se o mapa de cruzamento-tempo que é zero no alvo e monotonicamente crescente. Isso o torna adequado para o uso do método do gradiente para encontrar o alvo desde qualquer ponto no ambiente movendo-se ao longo da linha na direção do menor gradiente. Pela configuração da velocidade de propagação em função das características do ambiente, ou seja, o esforço de percorrer certas regiões, podemos criar um mapa de cruzamento-tempo cujo gradiente é mais acentuado quando está perto de obstáculos.

Figura 5.3: Domínio contínuo da frente de onda.

A monotonicidade do mapa cruzamento-tempo resulta da propriedade de *upwind*: a frente de onda pode ser interpretada como o envelope de todas as localizações possíveis que o robô pode alcançar se ele começa no alvo e sempre se move com a velocidade máxima permitida. Esse envelope está sempre se expandindo, o que nos permite determinar a região de influência de cada localização, traçando a direção de propagação. Se uma localização é posteriormente modificada, isto é, ocupada por um novo obstáculo, é possível determinar quais partes de um mapa cruzamento-tempo existentes podem ser mantidas intactas, e que porções precisam ser

recalculadas. Esta é a chave para o replanejamento eficiente, reduzindo o esforço de incorporar novas informações ambientais. O replanejamento significa que qualquer mudança no modelo do ambiente pode ser reparado para evitar o gasto do planejamento completo. Isso ajuda a compensar o maior esforço computacional requerido para a interpolação.

Os elementos do algoritmo E^* são apresentados a seguir:

- Alguns modelos do ambiente baseado em grafos.
- Uma fila ordenada para garantir uma propagação apropriada.
- Identificar as informações do *upwind* para inicializar corretamente as mudanças no custo do caminho, nas condições para os *kernels* de interpolação.
- Um mecanismo para determinar o conjunto de nós que podem influenciar a interpolação.

5.2.1 Representação do ambiente

O ambiente no qual o robô vai se movimentar é representado como um grafo não direcionado G incorporado no espaço de configurações $\mathcal{C}$. Muitas propriedades são ligadas aos nós $c \in G$, duas das quais são relevantes:

- O valor $v(c) \geq 0$ representa a amostra do mapa cruzamento-tempo no nó c , ou peso da função de navegação. Isto é similar ao custo ótimo do caminho do algoritmo A^* ou etiquetas das células do algoritmo NF1.
- A dificuldade ou custo de percorrer uma determinada configuração é codificado em esforço ou risco $r(c) \in [0, 1]$. Um valor menor de r implica uma maior velocidade. O esforço é convertido em meta-informação $m(c)$ para os diferentes *kernels* de interpolação, como será explicado nas próximas subseções.

Muitos planejadores baseados em grafos vinculam o esforço (custo do caminho) com os valores das bordas dos grafos, contrário ao método apresentado aqui. A escolha foi baseada no raciocínio de que os nós representam regiões do ambiente, e as bordas codificam informação topológica vinculando as regiões que são adjacentes umas as outras.

5.2.2 Propagação da frente de onda

Como no algoritmo A^* , onde é chamado “lista aberta”, a frente de onda é uma fila de nós que esperam ser expandidos, que é o passo elementar de propagação de um dado nó para seus vizinhos. O planejamento prossegue até que a frente de onda esteja vazia ou o nó que contém o robô tenha sido expandido. A frente de onda é ordenada ascendentemente pela chave, que é projetada para resultar em uma estrita ordem de propagação do *upwind*. À semelhança do D^* *lite* [21], um passo que processa o tempo de cruzamento chamado valor *rhs* é usado em conjunto com o valor estimado de $v(c)$ para calcular a chave da fila como $\min(v(c), rhs(c))$. Quando o valor do *rhs* e o valor de um nó são iguais, o nó é chamado (localmente) consistente. A propagação da frente de onda leva o algoritmo para um estado no qual todos os nós são localmente consistentes.

5.2.3 Grafo *upwind*

O algoritmo A^* utiliza uma árvore de expansão para traçar o caminho desde a posição do robô ao alvo. O algoritmo E^* estende isto a um grafo de *upwind* direcionado U com bordas únicas $(c_1, c_2) \in U \Rightarrow (c_2, c_1) \notin U$. Isto permite resgatar o conjunto de nós do *upwind* $U(c)$ [66] que estiveram envolvidos na computação de $v(c)$ assim como o conjunto de nós do *downwind* $D(c)$ [66] que foram influenciados por $v(c)$ durante a interpolação deste valor. Assim, todos os descendentes de um nó podem ser determinados se as informações do ambiente mudam, semelhante aos *backpointers* no algoritmo D^* .

5.2.4 Kernel de interpolação

Um *kernel* de interpolação k é uma função que estima o valor de cruzamento-tempo de um nó, baseado no risco ou esforço de atravessá-lo, em conjunto com os valores de seus vizinhos. O *kernel* de interpolação k é definido na equação (5.1)

$$(u, B) = k(c, Q) \quad (5.1)$$

onde u é o novo valor para o nó c , $B \subseteq Q(c)$ é o conjunto de vizinhos usados na computação de u , e $Q(c)$ é o propagador de c no tempo de expansão (5.2), o qual garante que só vizinhos candidatos válidos são proporcionados ao *kernel*

$$Q(c) = \{n \in N(c) | v(n) < \infty\} \quad (5.2)$$

onde $N(c)$ é o conjunto de vizinhos (nó adjacentes) de c . Os nós com valor infinito são obstáculos ou não foram expandidos, como depois da inicialização ou de um incremento em $r(c)$. Q é (parcialmente) ordenado pelo valor crescente v dos nós que pertence a Q : $v(Q_i) \leq v(Q_j), \forall i < j$, onde Q_i denota o i -ésimo nó que pertence a Q . Existe a possibilidade de Q estar vazia ($Q = \{\}$) ou conter um só nó ($Q = Q_1$). Para estes casos apresentam-se as seguintes propriedades do *kernel*:

$$\begin{aligned} (a) \quad & u(c, Q) > v(Q_1), \\ (b) \quad & u(c, Q) \leq u(c, Q_1), \\ (c) \quad & u(c, Q = \{\}) = \infty, \end{aligned} \tag{5.3}$$

onde a propriedade (a) é a condição do *upwind* que garante a propagação monótona, (b) significa que usando a interpolação leva-se ao menor valor em comparação à escolha de um movimento discreto, e (c) formaliza que não é válida a propagação salvo que um dos vizinhos seja um nó sem obstáculo, o qual conhece-se que é atingível.

5.2.5 O Algoritmo

O pseudo-código do algoritmo E^* é mostrado no pseudo-código 1. A variável g é o alvo, W denota a fila da frente de onda, e $chave(c)$ é a chave com a qual c tem sido inserida no W . O procedimento $ComputePropagator(c)$ utiliza a Equação 5.2, $Pop(W)$ tira e retorna o mais alto nó de W , e $TopChave(W)$ retorna a chave do mais alto nó ou ∞ .

Pseudo-código: 1 Pseudo-código do algoritmo E^*

```

1: Procedimento Principal(c)
2: Inicializar  $rhs(c) = v(c) = \infty, \forall c \in G$ 
3: Inicializar alvo  $rhs(g) = \text{distância real}$ 
4: Inicializar  $W$  com  $g$ 
5: enquanto ( $rhs(c_{robot}) \neq v(c_{robot})$ ) ou  $TopChave(W) < v(c_{robot})$  fazer
6:   se  $W = \{\}$  então
7:     Alvo  $g$  não pode ser alcançado
8:   fim se
9:   Propagar()
10: fim enquanto

```

Pseudo-código: 1 Pseudo-código do algoritmo E^*

```

11: Procedimento Reordenamento( $c$ )
12:   se  $v(c) = rhs(c)$  então
13:     se  $c \in W$  então
14:       remover  $c$  desde  $W$ 
15:     fim se
16:   senão
17:     se  $c \notin W$  então
18:       inserir  $c$  com  $chave = \min(v(c), rhs(c))$  em  $W$ 
19:     senão se  $chave(c) \neq chave = \min(v(c), rhs(c))$  então
20:       remover  $c$  desde  $W$ 
21:       inserir  $c$  com  $chave = \min(v(c), rhs(c))$  em  $W$ 
22:     fim se
23:   fim se
24: Procedimento AtualizarVertice( $c$ )
25: se  $c \notin g$  então
26:    $Q \leftarrow ComputarPropagador(c)$ 
27:    $(rhs(c), B) \leftarrow k(c, Q)$ 
28:   para cada  $u \in U(c)$  fazer
29:     remover  $(u, c)$  de  $U$ 
30:   fim para cada
31:   para cada  $b \in B$  fazer
32:     se  $(c, b)$  então
33:       remover  $(c, b)$  de  $U$ 
34:     fim se
35:     agregar( $b, c$ ) a  $U$ 
36:   fim para cada
37:   Reordenamento( $c$ )
38: fim se
39: Procedimento Propagar( $c$ )
40:  $c \leftarrow Pop(W)$ 
41: se  $v(c) > rhs(c)$  então
42:    $v(c) \leftarrow rhs(c)$ 
43:   para cada  $n \in N(c)$  fazer
44:     AtualizarVertice( $n$ )
45:   fim para cada
46: senão
47:    $v(c) \leftarrow rhs(c)$ 
48:   para cada  $d \in D(c)$  fazer
49:     AtualizarVertice( $d$ )
50:   fim para cada
51:   AtualizarVertice( $c$ )
52: fim se

```

Capítulo 6

Resultados

Neste capítulo, são apresentados os resultados obtidos via simulação das implementações dos algoritmos de desvio de obstáculos descritos no Capítulo 5. Estes algoritmos de desvio de obstáculos foram aplicados ao modelo do robô móvel descrito no Capítulo 2, utilizando os parâmetros nominais do robô móvel *Pioneer 3-AT*, os quais foram mostrados na Tabela 4.1.

O algoritmo de desvio de obstáculos gera uma trajetória de referência, posição e velocidade, as quais o robô tem que acompanhar para conseguir atingir o alvo. A Figura 6.1 mostra o sistema de acompanhamento da trajetória de referência.

Figura 6.1: Esquema do sistema de acompanhamento de trajetória.

As posições e velocidades de referências formam o vetor de entrada para o controlador baseado na cinemática descrito na seção 2.4, com os seguintes parâmetros $k_x = 0,5$, $k_y = 3$, $k_\psi = 0,5$.

A saída do controlador baseado na cinemática fornece as velocidades desejadas das rodas. Estes valores de velocidade compõem o vetor entrada para a atualização do modelo cinemático

do RMRD.

6.1 Simulação computacional do algoritmo *Tangent Bug*

Com o objetivo de testar o algoritmo *Tangent Bug*, uma simulação computacional foi realizada considerando um RMRD dotado de um sensor laser de varredura 180° com alcance máximo $R_{max} = 3,0 m$. O ambiente utilizado é uma área definida pelo projetista. O robô móvel deve atingir o alvo a partir de qualquer posição no ambiente, desviando dos obstáculos do ambiente. Definiu-se a velocidade do robô móvel como $v_r = 0,1 m/s$ e a distância mínima do robô ao obstáculo $d_{min} = 0,13 m$.

As Figuras 6.2 e 6.3 apresentam os resultados das simulações realizadas para diferentes posições iniciais (círculo vermelho) e alvos (x roxo), sendo a linha vermelha a trajetória de referência gerada pelo algoritmo *Tangent Bug* e a linha preta a trajetória realizada pelo RMRD. As indicações *C1* e *C2* apontam o início de cada comportamento, sendo *C1* o comportamento “movimento ao alvo” e *C2* o comportamento “acompanhamento da borda”.

Figura 6.2: Exemplo 1 da aplicação do algoritmo *Tangent Bug* para desvio de obstáculos.

Observa-se pelos resultados que o robô móvel alcançou o alvo efetivamente. O robô móvel realizou trajetórias suaves pelo ambiente desconhecido tanto para cada um dos comportamentos descritos anteriormente, quanto para a mudança entre esses comportamentos durante o movimento. Esta trajetória suave é gerada pela função polinomial *Cubic spline* [63], a qual gera a função polinomial cúbica. Os comportamentos *C1* e *C2* são evidenciados ao longo da traje-

tória de referência. Além disso, nota-se que os controladores aplicados ajudam a realizar um acompanhamento da trajetória de referência de maneira satisfatória.

Figura 6.3: Exemplo 2 da aplicação do algoritmo *Tangent Bug* para desvio de obstáculos.

6.2 Simulação computacional do algoritmo E*

Com o objetivo de testar o algoritmo E*, uma simulação computacional foi realizada considerando um mapa gerado por um sensor laser de varredura 180° com alcance máximo $R_{max} = 8,0 m$. O ambiente utilizado é área de trabalho do Laboratório de Sistemas Inteligentes (LASI), Departamento de Engenharia Elétrica, EESC, USP. O robô móvel deve atingir o alvo a partir de qualquer posição do ambiente, desviando dos obstáculos estáticos ou dinâmicos do ambiente, e a trajetória percorrida possui a menor distância possível entre a posição do robô e o alvo. Definiu-se a velocidade do robô móvel como $v_r = 0,1 m/s$.

O algoritmo foi implementado em *Visual C++*[©], e se utilizou a biblioteca ARIA e o software *MobileSim*[©] fornecidos por *MobileRobots*[©] para simular o robô móvel Pioneer 3-AT.

A primeira simulação foi realizada no ambiente descrito anteriormente, utilizando-se uma grade quadrada $h = 0,5 m$ e com obstáculos estáticos. Neste ambiente o RMRD deve atingir o alvo percorrendo a trajetória com menor distância.

A Figura 6.4(a) mostra a trajetória livre de obstáculos gerada pelo algoritmo E* no mapa em grade, isto é, o caminho cinza é a trajetória discretizada em grades do caminho gerada pelo algoritmo E*. A Figura 6.4(b) mostra o percurso da trajetória de referência gerada pelo algoritmo

E* feito pela simulação do RMRD, no simulador *MobileSim*[©]. A Figura 6.5 mostra a trajetória gerada pelo algoritmo E*, trajetória real do RMRD e a trajetória estimada usando a fusão de dados da odometria do RMRD e do receptor GPS. Os dados do receptor GPS foram simulados utilizando o método apresentado no Capítulo 4.

Figura 6.4: Trajetória gerada pelo algoritmo E* e percorrida pelo RMRD.

Figura 6.5: Trajetória de referência, real e estimada.

A segunda simulação foi realizada no mesmo ambiente com obstáculos estáticos e com a mesma grade da primeira simulação. O objetivo desta simulação é testar o replanejamento da trajetória feito pelo algoritmo E* quando o RMRD detecta um novo obstáculo estático ou dinâmico. Para isso, foi simulada a detecção do novo obstáculo para o tempo $t = 8$ s. Neste ambiente o RMRD deve atingir o alvo percorrendo a trajetória replanejada com a menor distância.

A Figura 6.6(a) mostra a trajetória replanejada livre de obstáculos gerada pelo algoritmo E* no mapa em grade, isto é, o caminho cinza é a trajetória discretizada em grades do caminho

gerada pelo algoritmo E^* . A Figura 6.6(b) mostra o percurso da trajetória de referência gerada pelo algoritmo E^* pela simulação do RMRD, no simulador *MobileSim*®. A Figura 6.7 mostra a trajetória inicial gerada pelo algoritmo E^* , a trajetória replanejada gerada pelo algoritmo E^* , a trajetória real do RMRD e a trajetória estimada usando a fusão de dados da odometria do RMRD e do receptor GPS. Os dados do receptor GPS foram simulados utilizando o método apresentado no capítulo 4.

Figura 6.6: Trajetória replanejada gerada pelo algoritmo E^* e percorrida pelo RMRD.

Figura 6.7: Trajetória de referência, real e estimada.

Observa-se pelos resultados que o robô móvel alcançou o alvo com sucesso, realizando uma trajetória suave pelo ambiente desconhecido. O algoritmo E^* permite realizar o replanejamento da trajetória efetivamente na presença de novos obstáculos ou obstáculos móveis. Além disso, nota-se que os controladores aplicados ajudam a realizar um acompanhamento da trajetória de referência de maneira satisfatória.

Parte III

Coordenação de robôs móveis

Capítulo 7

Controle de formação

Um dos principais objetivos do controle de formação, baseado apenas em posição e velocidade, que será apresentado neste capítulo, é fazer com que os robôs alcancem e mantenham posições e orientações pré-especificadas com relação a cada robô móvel da formação. Este tipo de controle, pode ser visto em [67].

7.1 Modelo do sistema

Assume-se que as posições e velocidades para cada um dos N robôs móveis no plano são descritas pela seguinte equação de estado:

$$\dot{x}_i = A_{veh}x_i + B_{veh}u_i \quad i = 1, 2, \dots, N \quad x_i \in \mathbb{R}^{2n}, \quad (7.1)$$

sendo que x_i representa as posições e as respectivas velocidades no plano x - y do robô móvel i , e é dado por:

$$x_i = [x_{r_i} \quad \dot{x}_{r_i} \quad y_{r_i} \quad \dot{y}_{r_i}]^T,$$

e, u_i representa as entradas de controle. As matrizes A_{veh} e B_{veh} tem a seguinte forma:

$$\mathbf{A}_{veh} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & a_{22} & 0 & a_{24} \\ 0 & 0 & 0 & 1 \\ 0 & a_{42} & 0 & a_{44} \end{pmatrix} \quad \mathbf{B}_{veh} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}. \quad (7.2)$$

Os zeros nas colunas um e três de A_{veh} são necessários para que a convergência dos robôs móveis para formação seja garantida (veja [68] Proposição 3.1 e [69] Proposição 4.2).

O vetor $x = (x_1, x_2, \dots, x_N)^T$ descreve a combinação de todos os estados dos N robôs móveis. Utiliza-se a notação

$$\begin{aligned} x_p &= ((x_p)_1, \dots, (x_p)_N)^T \\ x_v &= ((x_v)_1, \dots, (x_v)_N)^T \end{aligned}$$

para denotar os vetores de posição e velocidade respectivamente, para os N robôs móveis, tal que

$$x = x_p \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} + x_v \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

sendo $\otimes$ a representação do produto de Kronecker.

Definição 7.1.1. *Uma formação é definida pelo vetor*

$$h = h_p \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \in \mathbb{R}^{2nN}.$$

Os N robôs móveis estão em formação h no tempo t se existirem vetores $q, w \in \mathbb{R}^n$, tais que:

$$(x_p)_i(t) - (h_p)_i = q \quad e \quad (x_v)_i(t) = w,$$

para $i = 1, \dots, N$. Os robôs móveis convergem para formação h se existirem funções $q(\cdot), w(\cdot) \in \mathbb{R}^n$, tais que:

$$(x_p)_i(t) - (h_p)_i - q(t) \rightarrow 0 \quad e \quad (x_v)_i(t) - w(t) \rightarrow 0,$$

quando $t \rightarrow \infty$, para $i = 1, \dots, N$.

A Figura 7.1 ilustra este conceito de formação, sendo representada apenas a análise com relação à coordenada x do sistema de coordenadas inercial (X,Y).

Grafo de Comunicação: A topologia de comunicação entre os robôs móveis é representada por um dígrafo Γ (veja o Capítulo B) que captura a conectividade entre os robôs móveis. Cada vértice representa um robô e há uma borda direcionada de um vértice ao outro se há comunicação entre os robôs móveis. O robô móvel que recebe a informação é considerado vizinho do robô que está enviando a informação. Para cada robô i , J_i denota o conjunto de seus vizinhos.

Figura 7.1: Representação de uma formação com três RMs com alternância de líder.

Controle u : A natureza descentralizada do sistema de controle que está sendo considerado nesta pesquisa é devido ao fato de que cada controle u_i está em função de $x_j - x_i$ e $h_j - h_i$ para cada $j \in J_i$, ou seja, o sistema de controle consiste de componentes u_i que dependem somente das informações de posicionamento relativas ao i -ésimo robô e do conjunto de seus vizinhos.

Pode-se combinar as informações relativas a cada robô móvel definindo funções de saída y_i calculadas pela média dos deslocamentos relativos (e velocidades relativas) da vizinhança dos respectivos robôs móveis como:

$$y_i = (x_i - h_i) - \frac{1}{|J_i|} \sum_{j \in J_i} (x_j - h_j) \quad i = 1, \dots, N, \quad (7.3)$$

sendo que $|J_i|$ denota o número de vizinhos do robô i . Pode-se fazer uma pequena modificação na Equação (7.3) para permitir de que um robô não receba nenhuma informação e assim, o vértice correspondente no gráfico não tenha nenhum vizinho. Esta situação ocorre naturalmente quando um dos robôs é designado para ser o líder, em torno do qual os outros robôs móveis tem que ajustar seus movimentos (veja [35]). Neste caso, define-se a função de saída z_i como:

$$z_i = \begin{cases} \frac{1}{|J_i|} \sum_{j \in J_i} ((x_i - h_i) - (x_j - h_j)), & \text{se } (|J_i| \neq 0) \\ 0 & \text{caso contrário.} \end{cases} \quad i = 1, \dots, N. \quad (7.4)$$

O vetor de saída z pode ser escrito como $z = L(x - h)$ sendo $L = L_\Gamma \otimes I_{2n}$ e L_Γ a matriz (direcionada) Laplaciana do grafo de comunicação Γ (veja Capítulo B). Agrupando as equações

para todos os RMs em um único sistema no espaço de estado tem-se:

$$\dot{x} = Ax + Bu, \quad (7.5)$$

$$z = L(x - h), \quad (7.6)$$

sendo $A = I_N \otimes A_{veh}$ e $B = I_N \otimes B_{veh}$. A existência de uma matriz de realimentação F tal que a solução de

$$\dot{x} = Ax + BFL(x - h), \quad (7.7)$$

convirja para formação h pode ser vista em [67]. Este é um problema de estabilização envolvendo realimentação da saída. Considerando as estruturas de A , B e L tem-se F da forma $F = I_N \otimes F_{veh}$ (que é uma lei de controle descentralizada aplicada a todos os robôs móveis), neste caso pode-se reescrever a Equação (7.7) como:

$$\dot{x} = I_N \otimes A_{veh}x + L_\Gamma \otimes B_{veh}F_{veh}(x - h). \quad (7.8)$$

O Teorema apresentado a seguir será utilizado para a análise dos resultados do controle da formação sujeito à alternância de líderes.

Teorema 7.1.1. ([35]): *Considere o sistema de controle dado em (7.8). Existe uma matriz F_{veh} de tal modo que, para toda formação h a solução de (7.8) converge para a formação se e somente se o autovalor zero do Laplaciano do grafo direcionado L_Γ tem multiplicidade um.*

Utilizaremos a matriz de realimentação F_{veh} tendo a seguinte forma:

$$\mathbf{F}_{veh} = \begin{bmatrix} f_1 & f_2 & 0 & 0 \\ 0 & 0 & f_1 & f_2 \end{bmatrix}.$$

Em [67] mostra-se que as condições necessárias e suficientes para o sistema dado em (7.8) convergir para a formação são obtidas com $f_1 < 0$ e $f_2 < 0$. Porém, na simulação do programa para alternância de líderes verificou-se que esta condição não foi suficiente para que os robôs convergissem para formação.

Pode-se modificar a Equação (7.7) para que o robô líder acompanhe uma trajetória de refe-

rência,

$$\dot{x} = Ax + BFL(x - h) + K_{track}e_{track}, \quad (7.9)$$

$$e_{track} = r_{ref} - x, \quad (7.10)$$

onde e_{track} é o erro de acompanhamento, r_{ref} é a trajetória de referência, e K_{track} é o vetor de ganho do erro de acompanhamento.

7.2 Estratégia de alternância de líder

Um líder da formação é caracterizado pelo fato de não receber nenhuma informação de outro robô móvel. Isto significa que os outros robôs são forçados a arranjar suas posições em resposta ao movimento desse robô. O movimento de toda a formação é ditado pelo líder (veja [35]). Para que haja este líder o dígrafo de comunicação tem que possuir uma *árvore enraizada direcionada* (Lema B.1.1), ou a linha da matriz de adjacência Q (Equação (B.1)) do vértice que representa o líder tem que ser zero e a matriz D (Equação (B.2)) não invertível.

A estratégia de alternância de líder está baseada no controle reativo e é usada para o desvio de obstáculos, veja [6]. Considera-se que os RMs têm sensores de distância que permitem conhecer a distância a um obstáculo. A liderança da formação é assumida pelo RM que está mais próximo do que a distância limite a um obstáculo. Portanto, o novo líder muda sua posição e velocidade para acompanhar a trajetória planejada. Como resultado do controle de formação, o grupo todo de RMs muda suas posições e velocidades. Assim, a liderança alterna reativamente, o que possibilita ao grupo de RMs a capacidade de evitar colisões com os obstáculos.

Capítulo 8

Modelo Markoviano dos robôs móveis

Neste capítulo será apresentada a formulação do controle de formação tolerante a falhas baseada nas metodologias de projeto para sistemas lineares sujeitos a saltos Markovianos (SLSM). A falha considerada aqui corresponde à perda de um dos robôs da formação, em especial, pode-se considerar a perda do líder. Após a ocorrência da falha, a dinâmica do robô perdido não é considerada no projeto do controlador baseado em SLSM. Além disso, considera-se que o robô perdido não se tornará, após a falha, um obstáculo para os demais robôs da formação.

Os modelos dinâmicos dos robôs móveis em formação são linearizados em torno de pontos definidos na faixa de operação dos robôs móveis, caracterizada a partir dos limites máximos e mínimos das variáveis de posição e velocidade. O modelo Markoviano para este sistema considera os pontos de operação utilizados e a configuração de formação (se há ou não perda de robôs). O estado Markoviano atual é definido pelo ponto de operação atual, pelo número de robôs na formação e pela informação de qual robô é o líder.

Cabe salientar que esta abordagem de controle, tolerante a falhas, permite considerar outros aspectos que podem ser úteis para determinadas aplicações tais como alternância de líder, falhas de comunicação e obstáculos estáticos e dinâmicos.

Para obter o sistema linear referente aos pontos de operação, o modelo dinâmico dos RMRDs (2.28) com o acréscimo da perturbação no torque, é representado por:

$$\hat{M}\ddot{q}_{\theta_i} + \hat{N}\dot{q}_{\theta_i} + \hat{C} = \tau_i + \delta_i, \quad (8.1)$$

sendo

$$\begin{aligned}\hat{M} &= T_{wh2v}^T \bar{M} T_{wh2v}, & \hat{N} &= T_{wh2v}^T \bar{N} T_{wh2v}, \\ \hat{C} &= T_{wh2v}^T \bar{C}, & q_{\theta_i} &= \begin{bmatrix} \theta_{i_L} \\ \theta_{i_R} \end{bmatrix},\end{aligned}$$

onde θ_{i_L} e θ_{i_R} são os deslocamentos angulares das rodas esquerda e direita, respectivamente, T_{wh2v} é a matriz de transformação das velocidades das rodas ($\dot{q}_{\theta_i}$) a velocidade do RMRD ($\eta = [v_{x'}, \omega]^T$), veja Equação (2.6).

A linearização de (8.1) em torno de um ponto de operação ψ com posição $q_{\theta_i}^\psi$ e velocidade $\dot{q}_{\theta_i}^\psi$, para cada RM em formação, é dada por:

$$\dot{\bar{x}}_i = \bar{A}_i^\psi \bar{x}_i + \bar{B}_i^\psi \bar{\tau}_i + \bar{E}_i^\psi \delta_i, \quad (8.2)$$

sendo

$$\begin{aligned}\bar{A}_i^\psi &= \begin{bmatrix} 0 & I \\ -\hat{M}^{-1} \left(\frac{\partial(\hat{N}\dot{q}_{\theta_i})}{\partial q_{\theta_i}} \right) & -\hat{M}^{-1} \left(\frac{\partial(\hat{N}\dot{q}_{\theta_i})}{\partial \dot{q}_{\theta_i}} \right) \end{bmatrix} \bigg|_{(q_{\theta_i}^\psi, \dot{q}_{\theta_i}^\psi)}, \\ \bar{B}_i^\psi &= \bar{E}_i^\psi = \begin{bmatrix} 0 \\ \hat{M}^{-1} \end{bmatrix} \bigg|_{(q_{\theta_i}^\psi, \dot{q}_{\theta_i}^\psi)}, \quad \bar{x}_i = \begin{bmatrix} q_{\theta_i}^\psi - q_{\theta_i} \\ \dot{q}_{\theta_i}^\psi - \dot{q}_{\theta_i} \end{bmatrix}.\end{aligned}$$

O torque $\bar{\tau}_i$ é dado por

$$\bar{\tau}_i = \tau_i - \tau_0, \quad (8.3)$$

onde $\tau_0 = \hat{N}\dot{q}_{\theta_i} + \hat{C}$ é o torque para as condições iniciais.

Um controlador proporcional derivativo (PD) preliminar que pode ser introduzido no torque aplicado no i -ésimo robô é definido por

$$\bar{\tau}_i = [k_p \quad k_d] \bar{x}_i + \bar{u}_i, \quad (8.4)$$

sendo $\bar{u}_i$ a entrada de controle, para compensar incertezas no modelo. As variáveis de saída controladas do sistema, $\bar{z}_i$, são definidas como ponderações dos erros de posição, $\alpha_i (q_{\theta_i}^\psi - q_{\theta_i})$, e da entrada de controle, $\beta_i \bar{u}_i$. As variáveis de saída medidas, $\bar{y}_i$, são os erros de posição,

$(q_{\theta_i}^\psi - q_{\theta_i})$. Portanto, o sistema é representado como

$$\dot{\bar{x}}_i = A_i^\psi \bar{x}_i + B_i^\psi \bar{u}_i + E_i^\psi \bar{w}_i,$$

$$\bar{z}_i = C_{1_i}^\psi \bar{x}_i + D_{1_i}^\psi \bar{u}_i,$$

$$\bar{y}_i = C_{2_i}^\psi \bar{x}_i + D_{2_i}^\psi \bar{w}_i,$$

sendo

$$A_i^\psi = \begin{bmatrix} 0 & I \\ -\hat{M}^{-1} \left(\frac{\partial(\hat{N}\dot{q}_{\theta_i})}{\partial q_{\theta_i}} \right) + k_p \hat{M}^{-1} & -\hat{M}^{-1} \left(\frac{\partial(\hat{N}\dot{q}_{\theta_i})}{\partial \dot{q}_{\theta_i}} \right) + k_d \hat{M}^{-1} \end{bmatrix} \bigg|_{(q_{\theta_i}^\psi, \dot{q}_{\theta_i}^\psi)},$$

$$B_i^\psi = E_i^\psi = \bar{B}_i^\psi, \quad C_{1_i}^\psi = \begin{bmatrix} \alpha_i I & 0 \\ 0 & 0 \end{bmatrix}, \quad D_{1_i}^\psi = \begin{bmatrix} 0 \\ \beta_i I \end{bmatrix},$$

$$C_{2_i}^\psi = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, \quad D_{2_i}^\psi = 0,$$

$\bar{u}_i$ a entrada de controle calculada de acordo com o controlador Markoviano apresentado a seguir e $\bar{w}_i = \delta_i$. α_i e β_i são ponderações definidas pelo projetista para os erros de acompanhamento de trajetória e para a entrada de controle, respectivamente.

Figura 8.1: Modelo Markoviano para $n = 6$ robôs móveis.

Considera-se neste trabalho que o número de pontos de operação, Ψ , é o mesmo para cada uma das formações. Considera-se também que as formações resultantes de falha possuem apenas um robô perdido. Ou seja, para uma formação inicial com n robôs, teremos $n + 1$ possibilidades de formação. Uma formação com todos os robôs e n formações com um robô perdido, portanto o número de estados do sistema $N = 2$, veja Figura 8.1. As variáveis $P_0, P_f, P_{6f0}, P_{5fi}$ são as matrizes de probabilidades que indicam, respectivamente, a probabilidade de um robô móvel voltar a funcionar corretamente (neste modelo $P_0 = 0$), a probabilidade de uma ocorrência de falha, a probabilidade do sistema de $n = 6$ robôs móveis permanecer nessa configuração, a probabilidade do sistema de $n - 1$ robôs móveis com o i -ésimo robô permanecer nessa configuração. Formações com falhas que envolvam mais que um robô ao mesmo tempo serão consideradas em trabalhos futuros.

A matriz dinâmica de uma formação inicial com n robôs, para um determinado ponto de operação, é dada por:

$${}_0A^\psi = \begin{bmatrix} A_1^\psi & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \cdots & \vdots \\ 0 & \cdots & A_i^\psi & \cdots & 0 \\ \vdots & \cdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & A_n^\psi \end{bmatrix}.$$

A matriz dinâmica de uma formação com perda do i -ésimo robô é dada por:

$${}_iA^\psi = \begin{bmatrix} A_1^\psi & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \cdots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \cdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & A_n^\psi \end{bmatrix}.$$

As demais matrizes do sistema, ${}_iE^\psi, {}_iB^\psi, {}_iC_1^\psi, {}_iC_2^\psi, {}_iD_1^\psi$ e ${}_iD_2^\psi$, podem ser construídas de forma similar às matrizes ${}_iA^\psi$, $\psi = 1, \dots, \Psi$. Considere as coleções de matrizes reais dadas por:

$$\begin{aligned} A_\Theta &= ({}_0A^1, \dots, {}_0A^\Psi, {}_1A^1, \dots, {}_1A^\Psi, \dots, {}_nA^1, \dots, {}_nA^\Psi), \\ B_\Theta &= ({}_0B^1, \dots, {}_0B^\Psi, {}_1B^1, \dots, {}_1B^\Psi, \dots, {}_nB^1, \dots, {}_nB^\Psi), \\ E_\Theta &= ({}_0E^1, \dots, {}_0E^\Psi, {}_1E^1, \dots, {}_1E^\Psi, \dots, {}_nE^1, \dots, {}_nE^\Psi), \\ C_{1\Theta} &= ({}_0C_1^1, \dots, {}_0C_1^\Psi, {}_1C_1^1, \dots, {}_1C_1^\Psi, \dots, {}_nC_1^1, \dots, {}_nC_1^\Psi), \end{aligned}$$

$$\begin{aligned}
C_{2\Theta} &= ({}_0C_2^1, \dots, {}_0C_2^\Psi, {}_1C_2^1, \dots, {}_1C_2^\Psi, \dots, {}_nC_2^1, \dots, {}_nC_2^\Psi), \\
D_{1\Theta} &= ({}_0D_1^1, \dots, {}_0D_1^\Psi, {}_1D_1^1, \dots, {}_1D_1^\Psi, \dots, {}_nD_1^1, \dots, {}_nD_1^\Psi), \\
D_{2\Theta} &= ({}_0D_2^1, \dots, {}_0D_2^\Psi, {}_1D_2^1, \dots, {}_1D_2^\Psi, \dots, {}_nD_2^1, \dots, {}_nD_2^\Psi),
\end{aligned}$$

sendo o conjunto Θ definido como $\Theta = \{1, \dots, \Psi(n+1)\}$. O número de elementos do conjunto Θ define o número de estados Markovianos do sistema.

8.1 Controle $\mathcal{H}_\infty$ por realimentação de saída para sistemas lineares sujeitos a saltos Markovianos

O Controle $\mathcal{H}_\infty$ com realimentação da saída para SLSM apresentado neste trabalho pode ser visto em [70] e em [71]. Considere uma cadeia de Markov homogênea de tempo contínuo, $\theta(t) = \{\theta \in \Theta : t > 0\}$, com probabilidade de transição P_r definida como:

$$P_r(\theta(t + \Delta t) = j | \theta(t) = i) = \begin{cases} \lambda_{ij}\Delta + o(\delta), & \text{se } i \neq j \\ 1 + \lambda_{ii}\Delta + o(\delta), & \text{se } i = j, \end{cases}$$

com $i, j \in \Theta$, $\Delta > 0$, e $\lambda_{ij} \geq 0$ a taxa de transição do estado Markoviano i para o estado j ($i \neq j$), sendo

$$\lambda_{ii} = - \sum_{j=1, j \neq i}^{\Psi(N)} \lambda_{ij}.$$

A distribuição de probabilidade da cadeia de Markov no tempo inicial é dada por $\mu = (\mu_1, \dots, \mu_{\Psi(N)})$, sendo $P_r(\theta(0) = i) = \mu_i$. O sistema linear sujeito ao salto Markoviano é dado por:

$$\begin{aligned}
\dot{\bar{x}} &= A_{\theta(t)}\bar{x} + E_{\theta(t)}\bar{w} + B_{\theta(t)}\bar{u}, \\
\bar{z} &= C_{1\theta(t)}\bar{x} + D_{1\theta(t)}\bar{u}, \\
\bar{y} &= C_{2\theta(t)}\bar{x} + D_{2\theta(t)}\bar{w},
\end{aligned}$$

sendo que $\bar{w}$ possui norma limitada e $E(|\bar{x}(0)|^2) < \infty$. O controlador dinâmico é dado por:

$$\begin{aligned}
\dot{\bar{x}}_c &= A_{c\theta(t)}\bar{x}_c + B_{c\theta(t)}\bar{y}, \\
\bar{u} &= C_{c\theta(t)}\bar{x}_c,
\end{aligned} \tag{8.5}$$

sendo $A_{c\theta(t)} \in A_{c\Theta}$, $B_{c\theta(t)} \in B_{c\Theta}$ e $C_{c\theta(t)} \in C_{c\Theta}$. O problema de Controle $\mathcal{H}_\infty$ via realimentação de saída para SLSM consiste em encontrar controladores $(A_{c\theta}, B_{c\theta}, C_{c\theta})$, com $\theta \in \Theta$, tais que, a norma $\mathcal{H}_\infty$ do sistema em malha fechada seja menor que γ .

Para encontrar estes controladores, o conjunto de DMLs descrito em (8.6-8.8) deve ser resolvido:

$$\begin{bmatrix} A_\theta^T X_\theta + X_\theta A_\theta + L_\theta C_{2\theta} + C_{2\theta}^T L_\theta^T + C_{1\theta}^T C_{1\theta} + \sum_{j=1}^{\Psi(N+1)} \lambda_{\theta j} X_j & X_\theta E_\theta + L_\theta D_{2\theta} \\ E_\theta^T X_\theta + D_{2\theta}^T L_\theta^T & -\gamma^2 I \end{bmatrix} < 0, \quad (8.6)$$

$$\begin{bmatrix} A_\theta Y_\theta + Y_\theta A_\theta^T + B_\theta F_\theta + F_\theta^T B_\theta^T + \lambda_{\theta\theta} Y_\theta + \gamma^{-2} E_\theta E_\theta^T & Y_\theta C_{1\theta}^T + F_\theta^T D_{1\theta}^T & R_\theta Y \\ C_{1\theta} Y_\theta + D_{1\theta} F_\theta & -I & 0 \\ R_\theta^T(Y) & 0 & S_\theta(Y) \end{bmatrix} < 0, \quad (8.7)$$

$$\begin{bmatrix} Y_\theta & I \\ I & X_\theta \end{bmatrix} > 0. \quad (8.8)$$

sendo

$$R_\theta(Y) = \left[\sqrt{\lambda_{1\theta}} Y_\theta, \dots, \sqrt{\lambda_{(\theta-1)\theta}} Y_\theta, \sqrt{\lambda_{(\theta+1)\theta}} Y_\theta, \dots, \sqrt{\lambda_{(\Psi(N+1))\theta}} Y_\theta \right],$$

$$S_\theta(Y) = -\text{diag}(Y_1, \dots, Y_{\theta-1}, \dots, Y_{\theta+1}, \dots, Y_{\Psi(N)}).$$

Uma vez obtida a solução para o conjunto de DMLs, os controladores são projetados por:

$$C_{c\theta} = F_\theta Y_\theta^{-1}$$

$$B_{c\theta} = (Y_\theta^{-1} - X_\theta)^{-1} L_\theta$$

$$A_{c\theta} = (Y_\theta^{-1} - X_\theta)^{-1} M_\theta Y_\theta^{-1}$$

sendo,

$$M_\theta = -A_\theta^T - X_\theta A_\theta Y_\theta - X_\theta B_\theta F_\theta - L_\theta C_{2\theta} Y_\theta - C_{1\theta}^T \theta (C_{1\theta} Y_\theta + D_{1\theta} F_\theta)$$

$$- \gamma^{-2} (X_\theta E_\theta + L_\theta D_{2\theta}) E_\theta^T - \sum_{j=1}^{\Psi(N+1)} \lambda_{\theta j} Y_j^{-1} Y_\theta.$$

Capítulo 9

Resultados

Neste capítulo são apresentados os resultados de simulação e implementação para o controle de robôs em formação com alternância de líder e desvio de obstáculos, baseado no controle de formação tolerante a falhas. Este controle tolerante a falhas está baseado no controle Markoviano descrito no Capítulo 8 e aplicado a um grupo de robôs móveis descrito no Capítulo 2. Utilizam-se os parâmetros nominais do robô móvel *Pioneer 3-AT*, os quais foram descritos na Tabela 4.1.

A Figura 9.1 mostra o diagrama de blocos das estratégias de controle propostas para a coordenação de robôs móveis, e dos sistemas de localização e desvio de obstáculos desenvolvidos. O sistema de localização é o responsável por estimar a posição do robô móvel. Nesta simulação, este sistema utiliza só os dados do *encoder* para fazer a estimativa da posição. O sistema de desvio de obstáculos utiliza um dos dois algoritmos apresentados no Capítulo 5 para gerar a trajetória de referência livre de obstáculos para o líder da formação. A estratégia de alternância de líder seleciona qual dos robôs assume a liderança baseado na distância de robô ao obstáculo. O controle de formação gera as trajetórias e velocidades necessárias para que os robôs da formação converjam à formação desejada. O controlador Markoviano gera os torques que devem ser aplicados nas rodas para o acompanhamento da trajetória de referência gerada pelo sistema de desvio de obstáculos. Este controlador também considera os distúrbios nas rodas e a possível falha em um dos robôs da formação.

Figura 9.1: Sistema de controle de acompanhamento de trajetória e sistema de controle tolerante a falhas para os RMs.

9.1 Simulação computacional do controlador Markoviano

A simulação computacional foi realizado no software *Matlab*®. A trajetória de referência (x_r, y_r) para o robô líder é uma trajetória em linha reta paralela ao eixo x . Os $n = 6$ robôs considerados neste exemplo possuem as seguintes condições iniciais:

$$\begin{aligned}(x_{c_1}(0), y_{c_1}(0), \psi_{c_1}(0)) &= (0, 2p, 0), \\(x_{c_2}(0), y_{c_2}(0), \psi_{c_2}(0)) &= (0, 3p, 0), \\(x_{c_3}(0), y_{c_3}(0), \psi_{c_3}(0)) &= (0, 1p, 0), \\(x_{c_4}(0), y_{c_4}(0), \psi_{c_4}(0)) &= (0, 7p, 0), \\(x_{c_5}(0), y_{c_5}(0), \psi_{c_5}(0)) &= (0, 6p, 0), \\(x_{c_6}(0), y_{c_6}(0), \psi_{c_6}(0)) &= (0, 5p, 0).\end{aligned}$$

sendo $p = 1 \text{ m}$. Neste trabalho os RMs foram simulados considerando-se que estavam na seguinte formação (Figura 9.2):

$$\begin{aligned}h_1 &= [3, 5 \text{ , } 0]p; \quad h_2 = [2 \text{ , } 0, 5]p; \quad h_3 = [2 \text{ , } -0, 5]p; \\h_4 &= [0 \text{ , } 1, 5]p; \quad h_5 = [0 \text{ , } 0]p; \quad h_6 = [0 \text{ , } -1, 5]p.\end{aligned}$$

A mesma matriz de realimentação foi aplicada para todos os robôs, dada por:

$$\mathbf{F}_{\text{veh}} = \begin{bmatrix} -0,02555 & -0,26 & 0 & 0 \\ 0 & 0 & -0,02555 & -0,26 \end{bmatrix}.$$

O tempo de amostragem do sistema é $t_s = 0,20 \text{ s}$ e o tempo de amostragem de cada robô é $t_{RM} = 0,040 \text{ s}$.

Para realizar a simulação considerando a remoção de um dos robôs móveis, 4 dígrafos foram projetados, cada dígrafo possui duas possibilidades de falha nos robôs da formação de modo que os outros robôs não percam a comunicação. Se a falha acontecer no robô líder, a liderança muda para outro robô. A Figura 9.2 mostra a forma desejada da formação, e também os 4 dígrafos de comunicação projetados: 3 dígrafos quando o líder é o robô 1 e 1 dígrafo quando acontece a falha no robô 1. Os círculos negros correspondem ao robô líder da formação, e os círculos tracejados são os robôs com probabilidade de falha. Assim, para cada falha em um robô tem-se um L_Γ .

Figura 9.2: Dígrafos para possibilidade de falha nos robôs.

Foi considerado o robô 1 como o líder da formação durante toda a simulação. A simulação da falha foi obtida removendo o robô da formação em um tempo pré-determinado, de maneira que mesmo com a perda do líder os robôs não perderiam a comunicação, veja Figura 9.2.

Da Figura 9.1, o controlador de formação fornece as posições e as velocidades de referência de cada robô móvel para convergir à formação desejada. Estas posições e velocidades de referência de cada robô formam as entradas do controlador baseado na cinemática descrito na Seção 2.4, com os seguintes parâmetros: $k_x = 0,15$; $k_y = 0,4$; $k_\psi = 0,2$. Nesta simulação, a saída do controlador baseado na dinâmica fornece os torques a serem aplicados nas rodas do robô móvel (modelo dinâmico) para conseguir atingir as velocidades desejadas geradas pelo controlador cinemático.

Assim, o sistema inicia o movimento coordenado com um conjunto de seis robôs e durante o movimento o robô 5 será removido. Considera-se também que um ponto de linearização em 1 rad/s caracteriza ambos os cenários pré e pós falha para calcular as respectivas matrizes dinâmicas do sistema linear, ou seja, $\Psi = 1$. Portanto, temos 2 estados Markovianos, um para cada formação. Seguindo a notação apresentada no Capítulo 8, o conjunto de matrizes dinâmicas do modelo Markoviano é dado por $A_\Theta = ({}_0A^1, {}_5A^1)$, sendo que 0 à esquerda de A significa que o movimento coordenado evolui inicialmente com todos os robôs e 5 significa que houve uma falha e o robô 5 foi excluído da formação. Para esse modelo Markoviano $\Theta = \{1, 2\}$.

As taxas de transição entre os estados Markovianos são agrupadas em uma matriz de taxa de transição Λ de dimensão 2×2 . A matriz Λ é particionada nas seguintes submatrizes:

$$\Lambda = \begin{bmatrix} \Lambda_{6f0} & \Lambda_f \\ \Lambda_0 & \Lambda_{5f5} \end{bmatrix}.$$

A submatriz Λ_{6f0} mostra as relações entre os pontos de operação, e a submatriz diagonal Λ_f determina a probabilidade de ocorrência de falha. Depois da falha ter ocorrido, o sistema muda para a segunda linha de Λ , e Λ_0 mostra que a formação inicial não pode ser recuperada. Para o

problema em questão, os termos da matriz de taxa de transição Λ foram selecionados como:

$$\Lambda_{6f0} = -0.1, \quad \Lambda_{5f5} = 0, \quad \Lambda_f = 0.1, \quad \Lambda_0 = 0.$$

A Figura 9.3 mostra os resultados experimentais da trajetória efetuada pela formação utilizando o controlador Markoviano via realimentação de saída. Durante toda a simulação, o robô 1 é líder da formação, representado pelo círculo em negro, e, em $t_d = 9.36$ s o robô 5 foi removido e a falha detectada. Note que as linhas tracejadas na Figura 9.3 denotam as trajetórias desejadas dos RMRDs e a linha contínua a trajetória real. A Figura 9.4 mostra que o distúrbio externo foi aplicado em dois instantes nas rodas dos robôs móveis, um quando a formação estava completa e outro quando a formação tinha perdido o robô 5.

Figura 9.3: Trajetória da formação com falha no robô 5.

Figura 9.4: Distúrbios aplicados nas rodas esquerda e direita dos robôs.

Das simulações obtidas, observa-se que após a ocorrência da falha o sistema mantém a estabilidade com os robôs em movimento coordenado durante a reconfiguração do controle.

As Figuras 9.5(a) e 9.5(b) mostram as velocidades das rodas esquerda e direita de cada robô da formação, respectivamente. A parte superior destas figuras mostra as velocidades dos três primeiros robôs (1, 2, 3), e a parte inferior, dos outros robôs (4, 5, 6). Nestas figuras, observa-se que a velocidade das rodas do robô 5 é zero após a ocorrência da falha. Observa-se o efeito do distúrbio sobre a velocidade das rodas esquerda e direita.

Figura 9.5: Velocidades das rodas esquerdas e direitas dos robôs da formação.

As Figuras 9.6(a) e 9.6(b) mostram os torques aplicados nas rodas esquerda e direita de cada robô da formação, respectivamente. A parte superior destas figuras mostra os torques dos três primeiros robôs (1, 2, 3), e parte inferior, dos outros robôs (4, 5, 6). Nestas figuras, observa-se que os torques das rodas do robô 5 é zero após a ocorrência da falha. Também, observa-se o efeito do distúrbio sobre os torques das rodas.

Figura 9.6: Torques das rodas esquerdas e direitas dos robôs da formação.

9.1.1 Estudo comparativo

O controlador Markoviano foi comparado com um controlador *PD* utilizando os seguintes índices de desempenho.

- Norma $\mathcal{L}_2$ dos erros de posição e velocidade

$$\mathcal{L}_2[e] = \left(\frac{1}{(t_r - t_0)} \int_{t_0}^{t_r} \|e\|_2^2 dt \right)^{0.5}$$

sendo $e = [q_e \dot{q}_e]$,

- Energia total

$$E[\tau] = \sum_{i=1}^2 \left(\int_{t_0}^{t_r} \|\tau_i\| dt \right)$$

sendo $t_0 = 0$ até $t_r = 28$ s o tempo de experimento e $\tau_i(t)$ o torque da roda i (direita ou esquerda).

Os resultados estão apresentados na Tabela 9.1

Tabela 9.1: Índices de desempenho.

Controlador	Sem distúrbios		Com distúrbios	
	$\mathcal{L}_2[e]$	$E[\tau]$	$\mathcal{L}_2[e]$	$E[\tau]$
Markoviano	0,4842	63,1759	0,5489	78,6642
PD	0,4355	51,4160	0,6368	79,2231

Observa-se na Tabela 9.1 que sem a presença de distúrbios, o controlador PD apresenta melhores resultados, no índice do erro de acompanhamento de trajetória, que o controlador Markoviano. Também, vale ressaltar que o controlador *PD* apresentou melhor economia de energia. Com a presença de distúrbios, verifica-se que o controlador Markoviano apresenta melhores resultados, no índice do erro de acompanhamento de trajetória, que o controlador PD. Também, os dois controladores apresentam resultados equivalentes no índice de energia total.

9.1.2 Implementação do controlador Markoviano

O controlador Markoviano, simulado em *Matlab*®, foi implementado em *C++*, utilizando o software *Player/Stage*® para simular os robôs móveis Pioneer 3-AT. Foram considerados os mesmos parâmetros utilizados na simulação em *Matlab*®.

Os valores dos parâmetros do controlador de formação e do controlador baseado na cinemática foram projetados para não superar os limites máximos das velocidades linear e angular dos robôs móveis. Para os *Pioneer-3AT* a velocidade linear máxima é $1,5 \text{ m/s}$ e velocidade angular máxima é $2,0 \text{ rad/s}$.

As Figuras 9.7 e 9.8 mostram os resultados experimentais da trajetória efetuada pela formação utilizando o controlador Markoviano no simulador *Player/Stage*®. Durante toda a simulação o líder da formação é o robô 1 representado pela cor vermelha, e, em $t_d = 9.36 \text{ s}$ o robô 5 foi removido e a falha detectada.

Figura 9.7: Grupo de 6 *Pioneer 3-AT* com falha no robô 5.

Figura 9.8: Trajetórias do grupo de 6 *Pioneer 3-AT*.

Da simulação realizada, observa-se que o controlador Markoviano foi implementado com sucesso e que, após a ocorrência da falha, o sistema mantém a estabilidade com os robôs em movimento coordenado durante a reconfiguração do controle.

9.2 Simulação computacional do controlador Markoviano com alternância de líder

Para realizar a simulação considerando a estratégia de alternância de líder e a remoção de um dos robôs móveis, seis dígrafos foram projetados, um para cada robô assumir a liderança da formação. Cada dígrafo possui duas possibilidades de falha nos robôs da formação, de modo que os outros robôs não percam a comunicação. Se a falha acontecer no robô líder, a liderança passa para outro robô.

Figura 9.9: Dígrafos para cada líder da formação.

A Figura 9.9 mostra a forma desejada da formação e também os 4 dígrafos utilizados pela alternância de líder, quando o líder é o robô que está mais próximo aos obstáculos, nesta simulação são os robôs 3, 6, 1 e 4. Os círculos negros correspondem ao robô líder da formação, e os círculos tracejados são os robôs com probabilidade de falha. Assim, para cada robô líder tem-se um L_Γ .

A trajetória de referência livre de obstáculos a ser percorrida pelo robô líder foi gerada utilizando os algoritmos apresentados no Capítulo 5. A Figura 9.10 mostra a trajetória gerada pelo algoritmo *Tangent Bug*. Inicialmente o líder da formação é o robô 3 representado pelo círculo em negro, e, em $t_d = 44,32$ s o robô 5 foi removido e a falha detectada, nesse instante de tempo o robô 1 era o líder. As figuras 9.11(a) e 9.11(b) mostram o movimento do grupo de robôs com a estratégia de alternância de líder e as respectivas trajetórias deles.

Figura 9.10: Trajetória de referência gerada pelo algoritmo *Tangent Bug*.

Figura 9.11: Trajetória da formação com alternância de líder e falha no robô 5.

A Figura 9.12 mostra a trajetória gerada pelo algoritmo E^* . Inicialmente o líder da formação é o robô representado pelo círculo em negro, e, em $t_d = 44,32$ s o robô 5 foi removido e a falha detectada. As figuras 9.13(a) e 9.13(b) mostram o movimento do grupo de robôs com a estratégia de alternância de líder e as trajetórias destes.

Figura 9.12: Trajetória de referência gerada pelo algoritmo E^* .

Figura 9.13: Trajetória da formação com alternância de líder e falha no robô 5.

Das simulações obtidas, observa-se que o grupo de robôs móveis converge para a formação desejada, e toda a formação realiza o desvio de obstáculos utilizando a estratégia de alternância de líder. Também, observa-se que após a ocorrência da falha o sistema mantém a estabilidade com os robôs em movimento coordenado durante a reconfiguração do controle.

Capítulo 10

Conclusões

Neste trabalho, foi implementada a coordenação de robôs móveis baseada no controle de formação considerando a probabilidade de falha em um dos robôs da formação. Os robôs móveis mantêm uma formação desejada enquanto percorrem uma trajetória livre de obstáculos gerada pelo sistema de desvio de obstáculos. A estratégia de alternância de líder é utilizada pelo robôs para desviar dos obstáculos que estão mais próximo a estes.

Inicialmente apresentou-se o sistema de localização baseado na fusão de dados do *encoder* e do receptor GPS, o qual oferece uma estimativa satisfatória da posição do robô móvel, sendo esta melhor do que a estimativa da posição usando apenas dados da odometria ou do receptor GPS.

Os algoritmos do sistema de desvio de obstáculos em ambientes desconhecidos apresentaram-se como métodos efetivos, de simples implementação e com capacidade de replanejamento da trajetória em ambientes com novos obstáculos.

O movimento coordenado foi realizado com as seguintes abordagens: controle de formação, controle cinemático e controle dinâmico. O controle de formação foi implementado para um conjunto de RMRs de modo que os robôs se alternassem na liderança da formação para desviar dos obstáculos. Os resultados obtidos foram satisfatórios, porém, este tipo de controle não considera as restrições dinâmicas dos robôs. Assim, foi projetado um controlador robusto tolerante a falhas que considera o modelo dinâmico dos robôs. Sendo que para este controlador utilizou-se como trajetória de referência as trajetórias desejadas geradas pelo controle de formação.

A estratégia de controle tolerante a falhas garante a estabilidade da formação quando há perda de um dos robôs, mesmo que este robô seja o líder. O controle tolerante a falhas é baseado

em controle $\mathcal{H}_\infty$ por realimentação da saída de sistemas lineares sujeitos a saltos Markovianos.

Foram mostrados exemplos de simulação nos quais a alternância de líder foi utilizado para realizar o desvio de obstáculos e com a ocorrência de uma falha.

Verificou-se que a comunicação é assegurada na estratégia tolerante a falhas. Sendo a troca de informações entre os robôs móveis, realizada de maneira sistematizada de acordo com um grafo de comunicação direcionado, definido a-priori.

Verificou-se a efetividade dos três sistemas desenvolvidos, sistema de localização, desvio de obstáculos e coordenação de robôs móveis, para a navegação em ambientes externos e desviando de obstáculos, considerando que estes ambientes possuem o espaço para a navegação em formação dos seis robôs e também a ocorrência de falha em um dos robôs.

Em trabalhos futuros pretende-se:

- Estudar outros métodos para a localização de um conjunto de robôs móveis, tal como sistema de visão artificial.
- Estudar estratégias de mudança de formação, assim como, a estabilidade destas estratégias.
- Estudar controladores tolerante a falhas considerando a perda de mais de um robô.
- Implementar as estratégias e os controladores apresentados nesse trabalho em robôs móveis *Pioneer-3AT*.

Referências Bibliográficas

- [1] M. J. Mataric, “Behavior-based control: Examples from navigation, learning and group behavior,” *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 9, pp. 323 – 336, 1997.
- [2] H. Kruppa, D. Fox, W. Burgard, and S. Thrun, “A probabilistic approach to collaborative multirobot localization,” *Autonomous Robots*, pp. 325 – 344, 2000.
- [3] G. Whelan, J. Jennings, and W. Evans, “Cooperative search and rescue with a team of mobile robots,” *IEEE International Conference Advanced Robotics*, pp. 193 – 2000, 1997.
- [4] A. Howard, “Multi-robot simultaneous localization and mapping using particle filters,” in *In Robotics: Science and Systems*, 2005, pp. 201 – 208.
- [5] L. E. Parker, Y. Guo, and D. Jung, “Cooperative robot teams applied to the site preparation task,” in *In Proceedings About the Authors Parker & Zhang Collective Robotic Site Preparation 19 of the 10th International Conference on Advanced Robotics*, 2001, pp. 71 – 77.
- [6] L. Chaimowicz, R. V. Kumar, and M. F. M. Campos, “A mechanism for dynamic coordination of multiple robots,” *Autonomous Robots*, vol. 17, pp. 7 – 21, 2004.
- [7] L. E. Parker, “Toward the automated synthesis of cooperative mobile robot teams,” in *In Proceedings of SPIE Mobile Robots XIII*, 1998, pp. 82 – 93.
- [8] A. J. Healey, “Application of formation control for multi-vehicle robotic minesweeping,” in *Proceedings of the 40th IEEE Conference on Decision and Control*, vol. 2. IEEE, 2001, pp. 1497–1502.

- [9] R. Sattigeri and A. J. Calise, "An adaptive vision-based approach to decentralized formation control," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, Providence, Rhode Island, August 2004.
- [10] W. B. Dunbar and R. M. Murray, "Receding horizon control of multi-vehicle formations: A distributed implementation," in *43rd Conference on Decision and Control*, December 2004.
- [11] J. A. Farrel, *Aided Navigation*. McGraw-Hill, 2008.
- [12] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. Massachusetts Institute of Technology, 2004.
- [13] A. Boukerche, H. Oliveira, E. Nakamura, and A. Loureiro, "Vehicular ad hoc networks: A new challenge for localization-based systems," *Computer Communications*, vol. 31, no. 12, pp. 2838 – 2849, 2008.
- [14] J. R. Schoenberg, M. Campbell, and I. Miller, "Localization with multi-modal vision measurements in limited gps environments using gaussian sum filters," in *Proceedings of IEEE international conference on Robotics and Automation*, May 2009, pp. 1423 – 1428.
- [15] R. Mottaghi, M. Kaess, A. Ranganathan, R. Roberts, and F. Dellaert, "Place recognition-based fixed-lag smoothing for environments with unreliable gps," in *IEEE International Conference on Robotics and Automation*, May 2008.
- [16] Y. Morales, E. Takeuchi, and T. Tsubouchi, "Vehicle localization in outdoor woodland environments with sensor fault detection," in *IEEE International Conference on Robotics and Automation*, Pasadena, CA, June 2008, pp. 449 – 454.
- [17] C. Fouque, P. Bonnifait, and D. Betaille, "Enhancement of global vehicle localization using navigable road maps and dead-reckoning," in *IEEE PLANS08 Position, Location and Navigation Symposium*, Monterrey, California, United States, 2008, pp. 1286 – 1291.
- [18] R. Stancic and S. Graovac, "The integration of strap-down ins and gps based on adaptive error damping," *Robotics and Autonomous Systems*, vol. 58, no. 10, pp. 1117 – 1129, 2010.
- [19] M. Joerger and B. Pervan, "Autonomous ground vehicle navigation using integrated gps and laser-scanner measurements," in *IEEEION Position Location And Navigation Symposium*, 2008, pp. 988 – 997.

-
- [20] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2010, pp. 4372 – 4378.
- [21] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementation*. MIT Press, June 2005.
- [22] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.
- [23] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Springer, 2008.
- [24] H. Tang, A. Song, and X. Zhang, "Hybrid behavior coordination mechanism for navigation of reconnaissance robot," in *International Conference on Intelligent Robots and Systems*, Beijing - China, October 2006.
- [25] V. Lumelsky and T. Skewis, "Incorporating range sensing in the robot navigation function," *IEEE Transaction on Systems, Man and Cybernetics*, vol. 20, pp. 1058 – 1069, 1990.
- [26] I. Kamon, E. Rivlin, and E. Rimon, "A new range-sensor based globally convergent navigation algorithm for mobile robots," in *International Conference on Robotics and Automation*. Minneapolis, Minnesota: IEEE, April 1996.
- [27] I. Kamon, E. Rimon, and E. Rivlin, "Range sensor-based navigation in three-dimensional polyhedral environments," *International Journal of Robotics Research*, vol. 20, no. 1, pp. 6–25, January 2001.
- [28] J. Desai, J. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Transaction on Robotics and Automation*, vol. 17, no. 6, pp. 905 – 908, 2001.
- [29] B. S. Smith, J. Wang, and M. B. Egerstedt, "Persistent formation control of multi-robot networks," in *Proceedings of the IEEE Conference on Decision and Control*, . 476, Ed., Cancun, Mexico, December 2008.
- [30] W. Ren and R. Beard, "A decentralized scheme for spacecraft formation flying via the virtual structure approach," in *Proceeding of American Control Conference*, 2003, pp. 1746 – 1751.
- [31] M. Lewis and K. Tan, "High precision formation control of mobile robots using virtual structures," *Autonomous Robots*, vol. 4, no. 6, pp. 387 – 403, 1997.

- [32] D. P. Scharf, F. Y. Hadaegh, and S. R. Ploen., “A survey of space formation flying guidance and control (part 2),” *Proceedings of American Control Conference*, June 2004.
- [33] V. Ngo and Q. P. Ha, “Integration of planning and control in robotic formations,” in *Proceedings of the 2005 Australasian Conference on Robotics and Automation*, December 2005, pp. 1 – 8.
- [34] J. Fax and R. Murray, “Information flow and cooperative control of vehicle formations,” *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465 – 1476, 2003.
- [35] G. Laferriere, A. Williams, J. Caughman, and J. J. P. Veerman, “Decentralized control of vehicle formations,” *Systems and Control Letters*, 2005.
- [36] R. Olfati-Saber and R. M. Murray, “Distributed structural stabilization and tracking for formations of dynamic multiagents,” *Proceedings of IEEE Conference on Decision and Control*, pp. 209 – 215, 2002.
- [37] F. L. Lewis, C. T. Abdallah, and D. M. Dawson, *Control of robot manipulators*. New York: Macmillan., 1993.
- [38] J. Craig, *Adaptive Control of Mechanical Manipulators*. Addison-Wesley, 1985.
- [39] Y. C. Chang, “Neural network-based $\mathcal{H}_\infty$ tracking control for robotic systems,” *IEEE Proceedings of Control Theory Applications*, vol. 147, no. 3, pp. 303–311, 2000.
- [40] C. Y. C., “Intelligent robust control for uncertain nonlinear time-varying systems and its application to robotic systems,” *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, vol. 35, no. 6, pp. 1108–1119, 2005.
- [41] B. S. Chen, T. S. Lee, and J. H. Feng, “A nonlinear $\mathcal{H}_\infty$ control design in robotic systems under parameter perturbation and external disturbance,” *International Journal Control*, vol. 59, no. 2, pp. 439–461, 1994.
- [42] F. Wu, *Control of linear parameter-varying systems*. 150p. PhD Thesis - Department of Mechanical Engineering, University of California, Berkeley. 1995, 1995.
- [43] F. Wu, X. H. Yang, A. Packard, and G. Becker, “Induced $\mathcal{L}_2$ -norm control for lpv systems with bounded parameter variation rates,” *International Journal of Robust and Nonlinear Control*, vol. 6, no. 9-10, pp. 983–998, 1996.

-
- [44] B. S. Chen, Y. C. Chang, and T. C. Lee, "Adaptive control in robotic systems with $\mathcal{H}_\infty$ tracking performance," *Automatica*, vol. 33, no. 2, pp. 227–234, 1997.
 - [45] Y. C. Chang and B. S. Chen, "A nonlinear adaptive $\mathcal{H}_\infty$ tracking control design in robotic systems via neural networks," *IEEE Transaction on Control Systems Technology*, vol. 5, no. 1, pp. 13–29, 1997.
 - [46] Y. Huang and A. Jadbabaie, "Nonlinear $\mathcal{H}_\infty$ control: An enhanced quasi-lpv approach," in *IEEE International Conference on Decision and Control*, 37, Tampa, Florida, USA. Workshop in $\mathcal{H}_\infty$ nonlinear control by J. C. Doyle, Caltech, 1998.
 - [47] A. A. G. Siqueira and M. H. Terra, "Nonlinear and markovian $\mathcal{H}_\infty$ controls of underactuated manipulators," *IEEE Transactions on Control System Tecnology*, vol. 12, no. 6, pp. 811 – 826, November 2004.
 - [48] I. Postlethwaite and A. Bartoszewicz, "Application of non-linear $\mathcal{H}_\infty$ control to the tetrabot robot manipulator," *Proceedings of the Institution of Mechanical Engineers - Part I - Journal of Systems and Control Engineering*, vol. 212, no. 16, pp. 459–465, 1998.
 - [49] C. K. Hwang, B. S. Chen, and Y. T. Chang, "Combination of kinematical and robust dynamical controllers for mobile robotics tracking control: (i) optimal $\mathcal{H}_\infty$ control," in *Proceedings IEEE International Conference on Control Applications*, 2004, pp. 1205–1210.
 - [50] G. A. dos Reis, "Controle $\mathcal{H}_\infty$ não linear de robôs móveis com rodas," Master's thesis, Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos. 2005, 2005.
 - [51] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proceedings IEEE International Conference on Robotics and Automation*, 1990, pp. 384 – 389.
 - [52] T. B. R. Francisco, "Controle de movimentos coordenados de robôs móveis quando os robôs assumem a liderança de maneira aleatória," Master's thesis, Universidade de São Paulo, 2009.
 - [53] M. H. Terra, A. A. G. Siqueira, and T. B. Francisco, "Controle robusto de robôs móveis em formação sujeitos a falhas," *Controle e Automação*, vol. 21, no. 1, pp. 29 – 42, Fevereiro 2010.

- [54] L. Caracciolo, A. D. Luca, and S. Iannitti, "Trajectory tracking control of a four-wheel differentially driven mobile robot," in *IEEE International Conference on Robotics & Automation*, Detroit, Michigan, 1999.
- [55] K. Kozłowski and D. Pazderski, "Modeling and control of a 4-wheel skid-steering mobile robot," *International Journal of Applied Mathematics and Computer Science*, vol. 14, no. 4, pp. 477–496, 2004.
- [56] D. P. Bertsekas, *Dynamic Programming and Optimal Control vol. I*. Athena Scientific, 2000.
- [57] S. S. Ge and F. L. Lewis, *Autonomous Mobile Robots Sensing, Control, Decision Making and Applications*. Taylor & Francis Group, 2006.
- [58] A. H. Sayed, "A framework for state-space estimation with uncertain models," *IEEE Transaction on Automatic Control*, vol. 46, pp. 998 – 1013, July 2001.
- [59] K. N. Krishnanand and D. Ghose, "Formations of minimalist mobile robots using local-templates and spatially distributed interactions," *Robotic and Autonomous Systems*, vol. 53, pp. 194 – 213, October 2005.
- [60] V. J. Lumelsky and A. A. Stepanov, "Path-planning strategies for a point mobile automaton moving amidst obstacle of arbitrary shape," *Algoritmica*, vol. 2, pp. 403–430, 1987.
- [61] R. C. Arkin, *Behavior-Based Robotics*, M. Press, Ed. Cambridge, 1998.
- [62] S. Wei and M. Zefran, "Smooth path planning and control for mobile robots," in *IEEE Int. Conf. On Networking, Sensing and Control*, Tucson, Arizona, 2005.
- [63] C. de Boor, *A Practical Guide to Splines*. Springer - Verlag, 1978.
- [64] J. J. Craig, *Introduction To Robotics: Mechanics And Control*, 3rd ed. Person Prentice Hall, 2005.
- [65] R. Philippsen, "A light formulation of the e* interpolated path replanner," Autonomous Systems Lab, Ecole Polytechnique Federale de Lausanne, Tech. Rep., 2006.
- [66] R. Philippsen, "Motion planning and obstacle avoidance for mobile robots in highly cluttered dynamic environments," Ph.D. dissertation, École Polytechnique Fédéral de Lausanne, 2004.

-
- [67] A. Willians, G. Lafferriere, and J. Veerman, "Stable motions of vehicles formations," in *Proceedings of the 44th. IEEE Conference on Decision and Control*. Seville, Spain, 2005, pp. 72 – 77.
 - [68] G. Laferriere, J. Caughman, and A. Williams, "Graph theoretic methods in the stability of vehicle formations," *Proceedings of American Control Conference*, pp. 3724 – 3729, 2004.
 - [69] J. Veerman, G. Lafferriere, J. S. Caughman, and A. Williams, "Flocks and formations," *Journal of Statistical Physics*, vol. 121, no. 5-6, pp. 901 – 936, 2005.
 - [70] A. A. G. Siqueira, M. H. Terra, and C. Buosi, "Fault-tolerant robot manipulators based on output-feedback $\mathcal{H}_\infty$ controllers," *Robotic and Autonomous Systems*, April 2007.
 - [71] D. P. de Farias, J. C. Geromel, J. B. R. do Val, and O. L. V. Costa, "Output feedback control of Markov jump linear systems in continuous-time," *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 944 – 949, 2000.
 - [72] SANAV, *Model: FV-M8 GPS Receiver - USER'S GUIDE*, 05 2007.
 - [73] N. M. E. Associaton, *NMEA 0183 - Standard For Interfacing Marine Electronic Devices, Version 4.00*, 01 2002.
 - [74] R. Brualdi and H. Ryser, *Combinatorial Matrix Theory*. USA: Cambridge University Press, 1991.
 - [75] F. R. K. Chung, "Spectral graph theory," *CBMS Regional Conference Series in Mathematics*. Providence, RI: American Mathematical Society, 1997.
 - [76] J. A. Fax, "Optimal and cooperative control of vehicle formations," Ph.D. dissertation, California Institute of Thechnology, 2001.

Apêndice A

Cálculo da posição ECEF utilizando GPS

A.1 Módulo receptor GPS

O módulo receptor GPS que será utilizado é da série MTK-3301, modelo FV-M8, fabricado por *San Jose Technology*, veja Figura A.1.

Figura A.1: Módulo receptor GPS série MTK-3301.

As especificações mostradas em [72] são apresentadas na Figura A.2

A interface de comunicação serial é UART usando o protocolo NMEA (*National Marine Electronic Association*) [72]. O protocolo NMEA é um conjunto de especificações de dados e elétricas para comunicação de dispositivos eletrônicos de navegação, tal como, o receptor GPS. O módulo pode enviar 7 mensagens no padrão NMEA,

- GGA (*Global Positioning System Fix Data*).

- GSA (*GNSS DOP and Active Satellites*).
- GSV (*GNSS Satellites in View*).
- RMC (*Recommended Minimum Navigation Information*).
- VTG (*Course Over Ground and Ground Speed*).
- GLL (*Geographic Position - Latitude/ Longitude*).
- ZDA (*Time and Date*).

PHYSICAL CONSTRUCTION		
Dimension	L30mm*W30mm*H8.6mm	
Weight	10 grams	
Antenna	Active antenna on board	
Receiving frequency	1575.42MHZ; C/A code	
Connector	8pin connector with 1.0mm pitch	
Mounting	Soldering	
Construction	Full EMI Shielding	
ENVIRONMENTAL CONDITIONS		
Temperature	Operating: -30 ~ +80 ℃	
	Storage: -40 ~ +85 ℃	
COMMUNICATION		
Protocol	NMEA V3.01	
Signal level	UART @ 2.8V * 2	
INTERFACE CAPABILITY		
Standard Output Sentences	Default	RMC, GGA, GSV*5, VTG, GSA*5
	Optional	GLL, ZDA

PERFORMANCE		
Built-in Antenna	Highly-reliable ceramic patch	
Sensitivity	GPS	-158dbm
	Antenna	21db
SBAS	1 channel (Support WAAS, EGNOS, MSAS)	
DGPS	RTCM Protocol	
Receiver architecture	32 parallel channels	
Start-up time	Hot start	1 sec. typical
	Warm start	35 sec. typical
	Cold start	41sec. typical
Position accuracy	Without aid	3.3 m CEP
	DGPS (RTCM)	2.6 m
Velocity accuracy	0.1 Knot RMS steady state	
Update Rate	1 ~ 5Hz	
Power Supply	3.3~5V +- 5%	
Power Consumption	Acquisition	58mA
	Tracking	50mA (first 5 minutes)
		33mA (after 5 minutes)
		32mA (after 20 minutes)
Baud Rate	4800 bps (default) & 4800/9600/38400/57600/115200 bps are adjustable	

Figura A.2: Especificações do módulo receptor GPS.

Para este trabalho, usaremos 2 mensagens do padrão NMEA, GGA para obter a posição do robô e GSV para obter as posições dos satélites. Na Figura A.3(a) e A.3(b) mostra-se um resumo destas mensagens. Maiores informações sobre as mensagens do padrão NMEA encontra-se em [73].

A.2 Transformação para o sistema de coordenadas ECEF

Os dados obtidos pelo módulo receptor GPS estão em coordenadas Geodésicas (ϕ, λ, h) , onde ϕ é latitude, λ é longitude e h é altura em relação à superfície da terra. A transformação de dados em coordenadas Geodésicas à coordenadas ECEF $p = (x, y, z)$, é dado por

$$x = (R_N + h) \cos(\phi) \cos(\lambda) \quad (\text{A.1})$$

$$y = (R_N + h) \cos(\phi) \sin(\lambda) \quad (\text{A.2})$$

$$z = (R_N(1 - e^2) + h) \sin(\phi). \quad (\text{A.3})$$

sendo e a excentricidade da terra com valor de $e = 0,08181919$. O termo R_N é o raio de curvatura da elipse ao ponto p , e depende de a que é o raio equatorial da terra com valor de $a = 6\,378,137\,m$

$$R_N(\phi) = \frac{a}{(1 - e^2 \sin^2(\phi))^{1/2}} \quad (\text{A.4})$$

\$GPGGA,gga1,gga2,gga3,gga4,gga5,gga6,gga7,gga8,gga9,gga10,gga11,gga12,gga13,gga14*hh<CR><LF>

Parameters	Descriptions	Notes
gga1	UTC time as position is fixed	hhmmss.sss: hh – hour; mm – minute; ss.sss – second
gga2	Latitude	ddmm.mmmmmm: dd – degree; mm.mmmmmm – minute (0° ~ 90°)
gga3	Latitude sector	N – North; S – South
gga4	Longitude	dddmm.mmmmmm: dd – degree; mm.mmmmmm – minute (0° ~ 180°)
gga5	Longitude sector	E – East; W – West
gga6	GPS quality indicator	0 – No fixed or invalid position 1 – SPS Position available 2 – Differential GPS (SPS)
gga7	Number of SVs used in position estimation	xx: 00 ~ 12
gga8	HDOP	xx.xx: 00.00 ~ 99.99
gga9	Altitude above mean sea level (geoid)	xx.xxx: 00.000 ~ 99.999
gga10	Unit for Altitude	M: meter
gga11	Geoidal separation	
gga12	Unit for geoidal separation	M: meter
gga13	Age of differential corrections	unit : second; null when DGPS is not used
gga14	Reference station ID (DGPS)	xxxx: 0000 ~ 1023
hh	Checksum	hex number (2 – character)
<CR><LF>	End of message	

(a) Mensagem GGA.

\$GPGSV,gsv1,gsv2,gsv3,((gsv4,gsv5,gsv6,gsv7)*n)*hh<CR><LF>

Parameters	Descriptions	Notes
gsv1	Total number of messages	1 ~ 9
gsv2	Message number	1 ~ 9
gsv3	Total number of satellites in view	xx
gsv4	PRN number	xx
gsv5	Elevation (degrees)	90° maximum
gsv6	Azimuth (degrees)	0° ~ 360°
gsv7	SNR (C/N ₀)	0 ~ 99 dB-Hz, null when not tracking
hh	Checksum	hex number (2 – character)
<CR><LF>	End of message	

(b) Mensagem GSV.

Figura A.3: Mensagem GGA e GSV do padrão NMEA.

Apêndice B

Teoria algébrica dos grafos

Neste apêndice serão apresentados alguns conceitos básicos sobre a teoria dos grafos, que serão utilizados na planejamento da trajetória e na comunicação entre os robôs móveis.

B.1 Conceitos básicos

Grafos: Estruturas utilizadas para modelar problemas complexos, do ponto de vista computacional, nos quais existe bastante conhecimento matemático sobre suas propriedades e comportamento. Consiste de um conjunto finito de vértices e arestas (ou bordas). Graficamente, aparece representado por uma figura com nós ou vértices, significando os objetos, unidos por um traço, denominado aresta, configurando uma relação desejada.

Nesta pesquisa utilizaremos o grafo direcionado, cuja definição e notação utilizada é dada a seguir.

Grafo direcionado: Quando as arestas são pares ordenados de vértices, tem-se um grafo orientado (ou dígrafo), neste caso denomina-se de arco a aresta direcionada. Isto é, um dígrafo consiste de um conjunto finito de vértices $\mathcal{V}$ e um conjunto $\mathcal{E} \subseteq V \times V$ (bordas direcionadas). Assume-se que um dígrafo não tem voltas, isto significa que para $(x, y) \in \mathcal{E} \Rightarrow x \neq y$.

Se o dígrafo tem a seguinte propriedade: $(x, y) \in \mathcal{E} \Rightarrow (y, x) \in \mathcal{E}$ (trata-se de um grafo não direcionado) isto corresponde a uma comunicação bidirecional entre os robôs móveis. Um caminho (direcionado) em um dígrafo é uma sequência finita de bordas (a_k, b_k) $k = 1, \dots, r$, tal que, $b_k = a_{k+1}$ para $k = 1, \dots, r - 1$. Um caminho com vértices distintos é chamado trajeto (direcionado).

Os vértices de um dígrafo possuem:

Grau de entrada: número de arcos que chegam no vértice (*in-degree*).

Grau de saída: número de arcos que partem do vértice (*out-degree*).

Definição B.1.1. *Seja Γ um dígrafo com conjunto de vértices $\mathcal{V}$ e conjunto de bordas $\mathcal{E}$. Seja $Mat_{\mathcal{N}}$ o conjunto de todas matrizes quadradas $N \times N$ com entradas reais. A matriz de adjacência de Γ é a matriz $Q \in Mat_{\mathcal{N}}$ com entradas*

$$q_{ij} = \begin{cases} 1, & \text{se } (j, i) \in \mathcal{E}, \\ 0 & \text{caso contrário.} \end{cases} \quad (i, j \in \mathcal{V}). \quad (\text{B.1})$$

Quando Γ é não direcionado, a matriz Q é simétrica.

A matriz de grau interno de Γ é a matriz diagonal $D \in Mat_{\mathcal{V}}(\mathbb{R})$ com entradas diagonais

$$d_{ii} = |\{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}|, \quad (i \in \mathcal{V}). \quad (\text{B.2})$$

sendo, d_{ii} o grau de entrada do vértice i e $|\{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}|$ corresponde ao número total de j .

Definição B.1.2. *Dado um dígrafo Γ , a matriz Laplaciana associada à ele é dada por ([74]):*

$$L_{\Gamma} = D - Q. \quad (\text{B.3})$$

sendo D uma matriz não invertível.

Se D é invertível, então, $L_{\Gamma} = I_N - D^{-1}Q$ ([75] e [76]).

A seguir será apresentado o conceito de árvore enraizada direcionada, que será utilizado para a caracterização do líder da formação.

Lema B.1.1. : *Uma árvore enraizada direcionada é um dígrafo T com as seguintes propriedades:*

(I) T não tem ciclos;

(II) Existe um vértice v (a raiz) tal que há um trajeto direcionado de v a todos os outros vértices em T .