

INSTITUTO MILITAR DE ENGENHARIA

1º TEN FERNANDO APOLINÁRIO PEREIRA

**MÚLTIPLOS ROBÔS MÓVEIS AUTÔNOMOS EM UMA ESTRUTURA
COOPERATIVA**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Engenharia de Sistemas do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Ciências em Engenharia de Sistemas.

Orientador: Paulo Fernando Ferreira Rosa, PhD

Rio de Janeiro
2005

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

c2005

INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80 – Praia Vermelha
Rio de Janeiro – RJ CEP 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita à referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

P436 Pereira, Fernando Apolinário
Múltiplos robôs móveis autônomos em uma estrutura
cooperativa / Fernando Apolinário Pereira – Rio de Janeiro:
Instituto Militar de Engenharia, 2005.
108p.: il., graf., tab.

Dissertação (mestrado) – Instituto Militar de Engenharia – Rio de
Janeiro, 2005.

1. Robótica. 2. Computação Evolucionária 3. Planejamento de Trajetória 4. Inteligência Artificial. I. Título. II. Instituto Militar de Engenharia.

CDD 629.892

INSTITUTO MILITAR DE ENGENHARIA

FERNANDO APOLINÁRIO PEREIRA

**MÚLTIPLOS ROBÔS MÓVEIS AUTÔNOMOS EM UMA ESTRUTURA
COOPERATIVA**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Engenharia de Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Ciências em Engenharia de Sistemas e Computação.

Orientador: Paulo Fernando Ferreira Rosa, Ph. D.

Aprovada em 25 de novembro de 2005 pela seguinte Banca Examinadora:



Rio de Janeiro
2005

AGRADECIMENTOS

A todos que incentivaram, cooperaram e enriqueceram com sugestões e execuções na realização desta tese.

SUMÁRIO

LISTA DE ILUSTRAÇÕES	7	
LISTA DE TABELAS	10	
LISTA DE SÍMBOLOS E ABREVIATURAS.....	11	
1	INTRODUÇÃO	14
1.1.	Contexto e motivação	15
1.2.	Revisão de literatura.....	16
1.3.	Roteiro da dissertação	18
2	ROBÔS MÓVEIS: UMA BREVE ABORDAGEM	19
2.1.	Definições de robôs móveis	19
2.2.	Inteligência Robótica.....	21
3	TÓPICOS DE CONTROLE E PLANEJAMENTO DE TRAJETÓRIA ...	24
3.1.	Sistemas de controle não lineares.....	24
3.1.1.	Controlabilidade	24
3.1.2.	Teorema de Frobenius	25
3.1.3.	Condição de controlabilidade.....	26
3.1.4.	Lie Algebra.....	27
3.2.	Holonomicidade	28
3.2.1.	Confinamentos não holonômicos	29
3.2.2.	Classificação de sistemas não holonômicos.....	31
3.2.3.	Representação espaço de estados	34
3.2.4.	Linearização da realimentação entrada-saída.....	35
3.2.5.	Confinamentos cinemáticos e sistemas de controle	38
3.2.6.	Não-holonomicidade e controlabilidade	39
3.3.	Métodos de navegação para robôs.....	41

3.3.1.	Planejamento de trajetória – heurísticas relativas ao problema em questão	43
3.3.1.1.	Grafo de visibilidade	44
3.3.1.2.	Campo potencial artificial	45
3.3.1.3.	Métodos de Roadmap	49
3.3.1.3.1.	Diagrama de Voronoi.....	49
3.3.1.3.1.1.	Definições e propriedades básicas.....	51
3.3.1.3.2.	Árvores aleatórias de exploração rápida	54
3.3.1.3.3.	Árvores de navegação por saltos	55
3.3.1.3.4.	Decomposição em células	56
3.3.1.3.4.1.	Métodos exatos.....	56
3.3.1.3.4.2.	Métodos aproximados	57
3.3.2.	Tópicos em inteligência artificial	58
3.3.2.1.	Lógica Fuzzy	59
3.3.2.1.1.	Aspectos Qualitativos da Lógica Fuzzy	59
3.3.2.1.2.	Conjuntos Fuzzy.....	60
3.3.2.1.3.	Operações com conjuntos.....	60
3.3.2.1.4.	Funções de Pertinência.....	61
3.3.2.1.5.	Sistemas de Controle Fuzzy e neuro-fuzzy	64
3.3.2.1.6.	Aplicações da Lógica Fuzzy.....	66
3.4.	Erros de odometria sistemáticos e não sistemáticos.....	67
4	DESENVOLVIMENTO DO SISTEMA PROPOSTO	70
4.1.	Descrição do problema proposto.....	71
4.2.	Modelo cinemático e equações de confinamento.....	74
4.3.	Modelo dinâmico dos robôs móveis propostos	77
4.4.	Controlador fuzzy.....	84
4.5.	Campo potencial artificial	88
4.6.	Eletrônica embarcada	89
5	RESULTADOS.....	92
6	CONCLUSÃO	97

6.1.	Trabalhos futuros.....	97
6.2.	Comentários finais.....	98
7	REFERÊNCIAS BIBLIOGRÁFICAS	99
8	ANEXOS	106
9	ANEXO A – DESCRIÇÃO DOS CIRCUITOS, ESTRUTURA MECÂNICA DOS ROBÔS E MICROCONTROLADOR PIC16F877.....	107
10	ANEXO B – BREVE ESTUDO DE REDES NEURAI.....	113
11	ANEXO C – DEMONSTRAÇÃO DAS EQUAÇÕES 20 A 22 (LIE ALGEBRA).....	133
12	ANEXO D – DEMONSTRAÇÃO DO CÁLCULO DE S_3.....	134
13	ANEXO E – DEFINIÇÕES DE ESTABILIDADE	136
14	ANEXO F – TEOREMAS RELACIONADOS A DIAGRAMA DE VORONOI	138
15	ANEXO G – ANÁLISE DOS ERROS DE ODOMETRIA.....	142

LISTA DE ILUSTRAÇÕES

FIG. 2.1	Estruturas utilizadas em pesquisas.....	20
FIG. 3.1	Um veículo não holonômico.....	29
FIG. 3.2	Exemplo canônico de não holonomicidade (disco rolante).....	29
FIG. 3.3	Espaço de configuração do disco rolante.....	30
FIG. 3.4	Triangulação entre balizas e robô	42
FIG. 3.5	Representação do campo potencial artificial utilizado	47
FIG. 3.6	Triangulação Delaunay e Diagrama de Voronoi	51
FIG. 3.7	Diagrama de Voronoi dos pontos próximos.....	53
FIG. 3.8	Exemplo de utilização de árvores aleatórias de busca.....	55
FIG. 3.9	Funções de pertinência fuzzy.....	61
FIG. 3.10	Fuzzyficação de valores numéricos	62
FIG. 3.11	Representação do controlador fuzzy.....	66
FIG. 3.12	Crescimento das “elipses de erro”	69
FIG. 4.1	Modelo em 3 dimensões dos robôs utilizados na simulação	71
FIG. 4.2	Protótipo	72
FIG. 4.3	Suporte para o transporte da barra	72
FIG. 4.4	Módulo de possibilidade de colisão.....	73
FIG. 4.5	Controlador Fuzzy	73
FIG. 4.6	Fluxograma do algoritmo de controle.....	73
FIG. 4.7	Condição para mudança de liderança	74
FIG. 4.8	Representação do robô móvel.....	75
FIG. 4.9	Análise cinemática.....	79
FIG. 4.10	Estrutura líder-seguidor	81
FIG. 4.11	Variáveis de entrada e saída do controlador fuzzy.....	85
FIG. 4.12	Conjuntos fuzzy (a) Sensores, (b) Velocidades	85
FIG. 4.13	Conjunto fuzzy de possibilidade de colisão.....	87
FIG. 4.14	Representações em 2 D e 3 D do campo potencial artificial calculado.....	88
FIG. 4.15	Esquemático do circuito utilizado no líder	90
FIG. 4.16	Fluxograma do sonar	91
FIG. 5.1	Representação dos lóbulos do sonar e os pontos de contato	92
FIG. 5.2	Simulação com controlador fuzzy	93
FIG. 5.3	Sequência de movimentos do robô líder em um ambiente desestruturado.....	94

FIG. 5.4	Esquemático do circuito de controle do robô líder	96
FIG. 1	Diagrama dos pinos do microprocessador PIC da Microchip	107
FIG. 2	Diagrama esquemático da placa de circuito do kit de desenvolvimento McLab2	108
FIG. 3	Diagrama esquemático da Placa de circuito do kit McLab2 da Mosaico	108
FIG. 4	Diagrama do sensor de proximidade	109
FIG. 5	Lóbulo central dos sensores	110
FIG. 6	Esquemático da placa do sonar SRF08	110
FIG. 7	Estrutura dos robôs autônomos. a. Modelo virtual; b. Modelos reais	111
FIG. 8	Medidas do protótipo	111
FIG. 9	Desenho das peças que compõe o robô	112
FIG. 10	Camadas escondidas ou internas das redes neurais	114
FIG. 11	Rede de camada simples e alimentação direta	116
FIG. 12	Função de ativação sigmóide	121
FIG. 13	Função de ativação bipolar	122
FIG. 14	Rede neural em camadas com duas camadas contínuas de perceptron	123
FIG. 15	Rede neural de duas camadas com um nó de saída	124
FIG. 16	Rede de aprendizagem o vencedor leva tudo	128
FIG. 17	Atualização dos pesos do neurônio vencedor	130
FIG. 18	Vetores de peso final aponta para o centro de gravidade das classes	131
FIG. 19	Gráfico de estabilidade	137
FIG. 20	Representação do Diagrama de Voronoi	138
FIG. 21	Construção do Diagrama de Voronoi através dos bissetores	139
FIG. 22	Grafo planar conectado com vértice no infinito	140
FIG. 23	Construção dos vértices e bordas do Diagrama de Voronoi	141
FIG. 24	A configuração do triciclo	142
FIG. 25	Teste do trajeto quadrado	145
FIG. 26	Erros de odometria	146
FIG. 27	O efeito de dois erros sistemáticos dominantes E_b e E_d	148
FIG. 28	O efeito de dois erros odométricos sistemáticos E_b e E_d – direção oposta	149
FIG. 29	Resultados típicos da execução do experimento UMBmark	150
FIG. 30	Suscetibilidade aos erros não sistemáticos	153

LISTA DE TABELAS

TAB. 3.1 Métodos de implicações fuzzy.....	63
TAB. 4.1 Regras do controlador fuzzy.....	86
TAB. 1 Funções de ativação.....	121

LISTA DE SÍMBOLOS E ABREVIATURAS

\mathbb{R}	Conjunto dos números reais
\mathbb{R}^n	Conjunto dos vetores reais de dimensão n
$\mathbb{R}^{m \times n}$	Conjunto das matrizes reais $m \times n$

SIGLAS

CPA	Campo Potencial Artificial
CF	Controlador Fuzzy

NOTAÇÕES PARA SISTEMAS

$x(t)$	Sinal contínuo $x \in \mathbb{R}^n$
$u(t)$	Estado $u \in \mathbb{R}^m$ no instante t

RESUMO

A utilização de sistemas robóticos na realização de tarefas repetitivas já está bastante explorada na indústria. Porém, em ambientes desestruturados ou semi-estruturados, dos quais temos informações limitadas quanto à disposição estática ou dinâmica dos objetos, a utilização de novas abordagens, modelos e sistemas de controle faz-se necessária. Neste trabalho é proposto um sistema composto por dois robôs móveis autônomos em uma tarefa cooperativa, utilizando-se uma arquitetura líder-seguidor. Um dos robôs (líder) utiliza uma estratégia global (campo potencial artificial) e local (lógica fuzzy) para o seu planejamento de trajetória, do qual apenas os pontos de partida e de chegada são conhecidos. O robô seguidor acompanha fidedignamente a movimentação do líder e auxilia na solução de mínimos locais. A tarefa consiste no transporte livre de obstáculos de uma barra pelo conjunto líder-seguidor. Os protótipos experimentais dos robôs e simulações foram implementadas, a fim de validar o sistema proposto.

ABSTRACT

The use of robotic systems in the accomplishment of repetitive tasks are already very well explored in industry. However, in unstructured or semi-structured environments, about which we have limited information on the static or dynamic disposal of objects, the use of new approaches, models and control systems becomes necessary. This research proposes a cooperating system composed of two autonomous mobile robots in a cooperative task, using a leader-follower architecture. One of the robots (leader) uses a global (artificial potential field) and local (fuzzy logic) planning for its path planning, of which only the arrival and starting points are known. The follower robot faithfully follows the movement of leader and assists in the local minima solution. The task consists on the transport of a pipe by the robots, whilst they find their way out in real-time. The experimental archetypes of the robots and simulation had been implemented to validate of the proposed system.

1 INTRODUÇÃO

Este trabalho aborda o problema de coordenação de múltiplos robôs móveis autônomos na tarefa cooperativa de transporte de objetos em uma trajetória livre de obstáculos, em um ambiente semi-estruturado. Em Robótica, a cooperação entre robôs móveis tornou-se objeto de investigação em diversas situações, desde o transporte de peças em linhas de produção industrial, tarefas de montagem, de inspeção e vigilância, dentre outras. Um dos desafios é o de criar sistemas autônomos móveis capazes de realizar tais tarefas e outras similares com processamento em tempo real e, com um nível de inteligência suficiente para a resolução de problemas de planejamento de trajetória, aprendizagem e tomadas de decisão, com o mínimo custo computacional possível. Nos primórdios dessa área de pesquisa, o foco estava concentrado em ambientes estáticos estruturados (nos quais há o conhecimento prévio da localização dos obstáculos e de todos os parâmetros gerais que caracterizam este ambiente). Com o aumento da capacidade de processamento, os sistemas robóticos móveis passaram a dispor de um conjunto de sensores: ultrasônicos, óticos, de infravermelho e laser, dentre outros, para a realimentação do seu sistema de controle, auxiliando assim, o planejamento de trajetória e a execução adequada de tarefas. Os algoritmos baseiam-se em heurísticas e/ou meta-heurísticas, a fim de simplificar a modelagem do ambiente de trabalho.

O planejamento de trajetória é um dos mais importantes problemas de controle de robôs móveis, visto que atualmente uma vasta gama dos problemas da Robótica situam-se em espaços de trabalho semi-estruturados ou desestruturados. Por ambientes semi-estruturados, entende-se ambientes cujos parâmetros são apenas parcialmente conhecidos. Nesta situação, a trajetória dos robôs precisa ser formulada em tempo real, evitando-se obstáculos fixos e móveis, prevendo-se colisões e perseguindo um caminho otimizado. Na resolução deste problema é proposta a utilização de recursos da geometria computacional e da inteligência computacional.

O método do campo potencial artificial (CPA) foi primeiramente aplicado à navegação de robôs móveis por Oussama Khatib [Khatib, 1986]. O método estabelece um campo potencial virtual, imitando o movimento dos íons de um elemento químico, dentro de um campo eletromagnético. O planejamento de trajetória é implementado de acordo com a força do campo virtual. O método é eficiente em ambientes simples e pouco acidentados. No entanto, em ambientes complexos, com múltiplos obstáculos, áreas de mínimo local sempre aparecem na função potencial, as quais causam travamento ou vibração dos robôs. Por outro lado, o

método tradicional de campo potencial artificial somente busca áreas livres de obstáculos. Melhorias do método CPA foram propostas por Sato [Takuse, 1974], o qual eliminou as áreas de mínimo local através de uma pela definição mais elaborada do campo potencial, do que a anteriormente definida. Entretanto, o desempenho da trajetória não foi investigado.

A inteligência computacional, i.e., a área da computação que trata dos algoritmos adaptativos, que aprendem as mudanças do ambiente, fornecem um ferramental de grande valia em sistemas móveis, visto que podem ser implementados ou combinados de forma isolada com as heurísticas tradicionais. Dentre estes recursos, neste trabalho é realizado um controle híbrido utilizando-se CPA e lógica fuzzy para o planejamento de trajetória e realização do transporte de uma barra entre dois pontos. Os resultados obtidos foram satisfatórios.

simulador é o estudo da viabilidade de projetos de robôs através da simulação dos mesmos antes da sua construção, uma vez que a construção de robôs pode ser dispendiosa. Assim o uso do simulador permite desenvolver e testar técnicas de controle de maneira econômica, além de reduzir os custos do projeto físico dos robôs.

1.2. REVISÃO DE LITERATURA

A coordenação de múltiplos robôs móveis autônomos em tarefas de transporte foi implementada de maneira distribuída. Rus et al. [Rus, 1995], por exemplo, usou um grupo de robôs móveis para empurrar e colocar peças de mobília. Outro exemplo é fornecido por Chaimowicz et al. [Chaimowicz, 2001], que apresentou uma arquitetura hierárquica para coordenar manipuladores móveis na tarefa de carregar uma caixa. Novamente Chaimowicz et al. [Pereira, 2002] em outro trabalho propõe a utilização de uma arquitetura líder-seguidor com comunicação implícita entre os robôs a fim de eliminar a alta dependência da comunicação explícita na troca de mensagens de comando e dados sensoriais local. A comunicação explícita é definida como um ato específico projetado somente para enviar informações a outros robôs da equipe [Parker, 2000]. Por outro lado, a comunicação implícita ocorre devido ao efeito das ações dos robôs através da maneira como eles mudam o ambiente. A comunicação implícita oferece algumas vantagens imediatas sobre a forma explícita, dentre elas a robustez em ambientes com falhas de comunicação e menor consumo de energia. Balch e Arkin [Balch, 1994] mostraram que embora a comunicação melhore significativamente o desempenho de equipes de robôs, a comunicação explícita não é essencial quando a forma implícita está disponível. Khatib et al [Khatib, 1995] mostra uma tarefa de manipulação na qual a comunicação inter-robô é alcançada através da interação de forças sentidas por cada manipulador. Um outro exemplo de comunicação implícita é fornecido por Vaughan et al. [Vaughan, 2000], onde um grupo de robôs móveis desempenha uma tarefa em ambiente externo comunicando-se através de marcos deixados no ambiente. Stiweil e Bishop [Stiweil, 2001] apresentaram uma abordagem teórica mostrando que a quantidade de comunicação explícita pode ser reduzida pelo uso da forma implícita.

Este artigo é direcionado ao problema de coordenação de múltiplos robôs móveis autônomos na tarefa de transporte livre de obstáculos de uma barra em um ambiente semi-estruturado, dependendo dos confinamentos cinemáticos, da configuração observada entre

eles e os obstáculos no ambiente, tais como o problema do estacionamento de um caminhão-trailer [Altafini, 2001]. Em relação ao artigo de Chaimowicz et al. [Pereira, 2002] para a solução deste problema, a arquitetura líder-seguidor adotada nesta dissertação reduz a complexidade de hardware no robô seguidor para somente monitorar a posição e orientação da barra a fim de acompanhar o líder e realizar a troca de liderança. Adotaram-se dois tipos de sensores, em detrimento ao sensor resistivo utilizado, um ótico, para a movimentação transversal da barra, e um resistivo, para o monitoramento da rotação da barra. Além da mudança dos tipos de sensores nos robôs e da redução de processamento no robô seguidor, o algoritmo de controle embarcado no robô líder utiliza uma estratégia global (campo potencial artificial) e uma estratégia local (controlador fuzzy) para um melhor planejamento de trajetória, algo não abordado no trabalho de Chaimowicz et al. A não holonomicidade impõe restrições cinemáticas adicionais para o grupo quando obstáculos previamente desconhecidos (comuns a ambiente desestruturados e dinâmicos) precisam ser evitados.

Especificamente, uma situação mais simples onde dois robôs devem carregar uma barra cilíndrica de um ponto a outro em um ambiente desconhecido é considerado. Neste trabalho, um monitoramento da translação e rotação é realizado ao invés de sensores de força como implementado em [Sugar, 1998]. Os robôs podem localizar-se no ambiente através de sensores ultrasônicos e pela odometria incorporada ao líder. Talvez a principal dificuldade para controlar os robôs móveis em um ambiente desconhecido é a detecção e desvio dos obstáculos. Dependendo da configuração, o grupo pode não ser capaz de evitar completamente os obstáculos sem derrubar a barra. Em tais situações, os robôs devem recuar a fim de seguir uma nova trajetória devida, principalmente, à cinemática do conjunto. Portanto, a arquitetura dinâmica líder-seguidor incorporada ao controlador local fuzzy evita este problema. Embora os trabalhos de Kosuge e Oosumi [Kosuge, 1996] e Sugar e Kumar [Sugar, 1998] tenham utilizado para um problema de transporte similar um controle distribuído, adotamos para o nosso trabalho um controle centralizado no robô líder e o monitoramento da posição da barra concentra-se no robô seguidor. Além disso, apenas uma modalidade de sensor é utilizada para a detecção de obstáculos e um algoritmo híbrido com campo potencial e lógica fuzzy é usado para o planejamento de trajetória, com o intuito de, também, tornar o problema mais fácil de ser resolvido.

A principal contribuição desta dissertação é o desenvolvimento de uma metodologia hierárquica, onde a troca de liderança ocorre através da comunicação implícita entre os robôs, além do uso de um controle híbrido, unindo as vantagens do algoritmo do campo potencial artificial e da lógica fuzzy na realização do planejamento de trajetória.

1.3. ROTEIRO DA DISSERTAÇÃO

O objetivo principal deste projeto consiste no desenvolvimento de um algoritmo de controle baseado no CPA e lógica fuzzy implantados em um sistema cooperativo de transportar uma barra, satisfatoriamente, pelo ambiente sem derrubá-la, no menor tempo possível. O CPA é o algoritmo de planejamento de trajetória global iniciador do movimento do sistema robótico, ao passo que o controlador fuzzy tem a dupla função de monitorar a possibilidade de colisão e desvio de obstáculos não previstos pelo planejador global do robô móvel líder, sendo este acompanhado pelo robô móvel seguidor através de um ambiente semi-estruturado. A comunicação entre os robôs móveis é implícita, através do monitoramento da barra, as quais carregam. O controlador baseia-se na lógica fuzzy para fornecer comandos de direção ao robô, guiando-o de uma posição original a uma posição objetivo, evitando possíveis obstáculos, cujas informações advêm de três sonares localizados na região frontal da estrutura robótica e, defasados de 45 graus, entre si. A barra tem dois sensores, um monitorando o deslocamento transversal e outro o deslocamento angular de seu suporte.

Além deste objetivo principal, podemos considerar como objetivo secundário a construção, no espaço de trabalho, de dois protótipos a fim de validar as simulações realizadas na representação do espaço real. As informações obtidas do modelo real (sensores ultrasônicos, por exemplo), então, serão utilizadas pelo controlador para produzir valores adequados de velocidade e, assim comprovar as simulações. Assim, em vez de usarmos como entradas as leituras do modelo dos sonares do simulador, as leituras com as imprecisões inerentes a este tipo de sensor obtidas construirão o modelo real do ambiente. Nos capítulos seguintes, abordar-se-á a evolução da Robótica relacionada a sistemas autônomos móveis, um breve estudo de sistemas de controle não lineares, conceituação de holonomicidade, os métodos de navegação utilizados na área, os erros de odometria oriundos de características mecânicas e, finalmente, a implementação proposta com os detalhes do simulador e protótipos.

2 ROBÔS MÓVEIS: UMA BREVE ABORDAGEM

Neste capítulo, procura-se fornecer uma visão da Robótica, a conceituação, topologias e trabalhos relacionados a esta tese.

2.1. DEFINIÇÃO DE ROBÔS MÓVEIS

Segundo McComb, existem duas vertentes na definição de robôs móveis: uma que diz ser o robô completo, auto-contido, autônomo, que necessita de instruções de seu mestre apenas ocasionalmente. Já a segunda é aquela que define robô como sendo qualquer equipamento com capacidade de mover-se com seus próprios meios, com o objetivo de executar tarefas próximas às humanas [McComb, 1987]. As duas definições de McComb levam a aceitar que um robô móvel é capaz de manobrar livremente em seu ambiente, alcançando metas enquanto desvia de obstáculos [Spence, 1995].

Uma vez que necessitam ser livres, ou seja, sem conexões e fios, a aplicação de robôs móveis sofre limitações, basicamente à armazenagem e geração de energia para o próprio robô [McKerrow, 1991].

Um dos primeiros robôs móveis construídos e reconhecidos na bibliografia é o Shakey, desenvolvido pelo Stanford Research Institute, em 1968. Ele tinha uma variedade enorme de sensores, incluindo uma câmara de vídeo e sensores de toque, e navegava entre as salas de um laboratório, enviando sinais de rádio ao seu cérebro, um computador DEC PDP-10, que permitia efetuar algumas tarefas como empurrar caixas e evitar obstáculos. Deve-se atentar para o fato de que a unidade de processamento embarcada no robô apenas coletava os sinais sensoriais e os enviava para um computador remoto que executava o processamento, transmitindo ao robô o comando que geraria a ação desejada. Outra experiência relatada data de 1977, com o veículo StanfordCart do Stanford Artificial Intelligence Laboratory. O StanfordCart – trabalhando em um local plano com obstáculos espaçadamente colocados - utilizava um sistema de navegação baseado no “parar e seguir”, parando e fazendo leitura de seus sensores a cada metro percorrido, realizando o planejamento da rota a seguir.

Os robôs descritos acima, chamados robôs móveis autônomos, são capazes de encontrar solução para os seus próprios problemas em um ambiente sem supervisão externa. Entre os vários trabalhos nesta emergente área do conhecimento, impulsionada em grande parte pelo aumento da capacidade de manipulação de dados dos computadores, vale ressaltar o Hilare, um robô multisensorial desenvolvido no LAAS – Laboratoire d’Analyse et d’Architecture des Systèmes (unidade de pesquisa do Centro Nacional de Pesquisa Científica em Toulouse) em 1983.

Durante os anos 80, vários trabalhos foram desenvolvidos em todo o mundo como o NAVLAB do Carnegie-Mellon e o PSEIKI do Robot Vision Laboratory da Purdue University, baseado em reconhecimento de imagens através de câmeras. Outro estudo sobre robôs móveis foi o desenvolvimento do robô Khepera, do K-Team, no Microcomputing Laboratory do Swiss Federal Institute of Technology, com o apoio de outras entidades de pesquisa da Europa, como University of Zürich, University of Karlsruhe, University of Sussex, Laboratoire d’Etudes et Recherche en Informatique e Ecole Nationale Supérieure (Paris). Este robô, com apenas 55 mm de diâmetro por 30 mm de altura tem a capacidade de desviar-se de obstáculos e seguir ou evitar fontes luminosas, de forma similar ao descrito na obra de Braitenberg (Valentino Braitenberg, professor do Max Planck Institute for Biological Cybernetics – Tübingen – Alemanha), além de permitir a utilização de algumas extensões, como sistema de visão e um pequeno manipulador. Seu tamanho reduzido e seu custo relativamente baixo, comparado a outros robôs no mercado, têm feito o Khepera um sucesso em universidades, sendo objeto de pesquisa inclusive no Brasil. Outro robô comercial mais robusto para pesquisa desenvolvido pela empresa norte-americana Nomadic Technologies Inc é o Nomad, com 62 cm de diâmetro, 85 cm de altura e 60 kg podendo carregar até 100 kg, o qual é a base da estrutura do protótipo construído mostrado na figura 2.1. No próximo tópico serão descritas as plataformas utilizadas para robôs móveis e suas características.

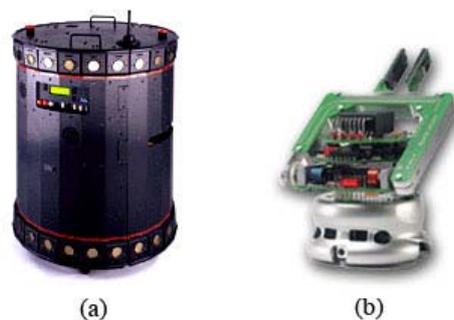


FIG. 2.1. Estruturas utilizadas em pesquisas: (a) Nomad modelo XR4000; (b) Khepera modelo Khepera II.

2.2. INTELIGÊNCIA ROBÓTICA

Como foi relatado anteriormente, para alguns autores os robôs sempre serão apenas conexões inteligentes entre percepção e ação [Jones, 1993; Rich, 1994] e, para que estas conexões possam ser definidas e tenham um comportamento realmente inteligente, técnicas de inteligência artificial podem ser utilizadas.

Ainda que o estudo semântico do que seja inteligência artificial não seja estudo deste trabalho, adotamos uma conceituação como balizamento.

A inteligência artificial, surgida no Dartmouth College, no verão de 1956 [Gevarter, 1984], pode ser definida, por sua vez, como um campo de estudo que busca explicar e emular comportamentos inteligentes em termos de processos computacionais [Schalkoff, 1990]. Se restringirmos o foco da discussão, encontraremos também conceitos mais simplistas, onde a inteligência artificial é uma tecnologia nova que tem sido utilizada de tal forma que os programas computacionais tornem-se otimizados [Gevarter, já citado anteriormente]. Fazendo um paralelo, finalmente encontra-se a interpretação, do ponto de vista da engenharia, dado por Shalkoff. Para ele a inteligência artificial é “... *gerar representações e procedimentos que automaticamente (autonomamente) resolvam problemas até agora resolvidos por humanos*” [Schalkoff, já citado anteriormente].

Algumas aplicações da inteligência artificial no campo da robótica estão relacionadas ao planejamento de trajetória e movimento (navegação), visão computacional, controle, sistemas de diagnóstico e planejamento da manufatura.

Alguns autores classificam robôs inteligentes como aqueles que realizam decisões em tempo real, sustentadas por algoritmos de inteligência artificial, baseados em informação de sensores [Koren, 1985], o que lhes possibilita trabalhar em ambientes completamente imprevisíveis.

Comparando-se as técnicas de controle tradicionais com aquelas provenientes da aplicação da inteligência artificial, podem-se classificar estas últimas como hierarquicamente superiores as primeiras [Gevarter, já citado anteriormente]. No caso de aplicação em máquinas autônomas, especificamente no caso ora estudado - ou seja, no campo da robótica - o emprego da inteligência nos robôs não os tornará apenas mais flexíveis nas linhas de montagens, mas também os tornarão cada vez mais aptos a executar atividades fora do ambiente industrial mais próximas das humanas, no conceito de Shalkoff.

No final da década de 1980, as pesquisas tornaram-se muito ativas em robótica cooperativa, iniciado com projetos tais como o CEBOT [Fukuda, 1988], SWARM [Beni, 1991], ACTRESS [Ishida, 1994] e GOFER [Le pape, 1990]. Os projetos anteriores eram realizados, primariamente, em simulação, ao passo que os trabalhos citados anteriormente foram implementados fisicamente (com no máximo 3 robôs) com o intuito de fornecer comprovação dos resultados simulados. A partir da década de 1990, estes sistemas robóticos cooperativos implementados fisicamente passaram a ter sua arquitetura de controle baseada em comportamento. Destes trabalhos, o planejamento de trajetória levava em consideração tanto os outros robôs quanto o ambiente global, era um problema intrinsecamente geométrico, quanto à cooperação, os trabalhos estavam direcionados ao problema de empurrar uma caixa apresentando algum aprendizado [Newman, 2003]. Com isso, surgiram alguns trabalhos, os quais utilizaram a comunicação implícita entre robôs com a função de empurrar duas caixas [Dam, 1996]. Outras abordagens para múltiplos robôs móveis eram a montagem de objetos [Dev, 1997], a varredura, busca e captura de objetos no ambiente [McFarland, 1993], tal tarefa é interessante pois, (1) pode ser desempenhada por cada robô independentemente e (2) a tarefa é também interessante devido a motivações relacionadas a inspirações biológicas.

Em sistemas robóticos cooperativos, pode-se dividir a realização do comportamento cooperativo em: 1) Arquitetura de grupo, envolve o conceito de homogeneidade/heterogeneidade dos robôs, ou seja, a capacidade de um dado robô reconhecer e modelar outros robôs, além da estrutura de comunicação; 2) Conflito de recursos, relaciona-se a múltiplos robôs habitarem um ambiente compartilhado, manipular objetos e a possibilidade de comunicação entre eles; 3) Origem da cooperação, refere-se como o comportamento cooperativo é motivado e alcançado; 4) Aprendizado, a adaptabilidade e flexibilidade são itens essenciais na solução de tarefas de um grupo de robôs; e 5) Problemas geométricos, cobre a tentativa de incorporar tarefas aos robôs em um ambiente bidimensional e tridimensional, incluindo o planejamento de trajetória multi-agente, movimentação em formação e geração de padrões.

Durante os estudos na área de robótica, inicialmente, as abordagens cooperativas relacionavam-se com ao planejamento de trajetória para a exploração de uma região [Brumitt, 1992], os artigos tratando de abordagens cooperativas para diversas tarefas teve seu maior salto a partir do ano de 2000, com trabalhos que utilizam múltiplos robôs móveis na realização de diversas tarefas de manipulação e sensoriamento decididas pelos robôs móveis de acordo com uma unidade de planejamento de tarefas baseadas em modelos de prioridade pré-definidos. O foco desta proposta é o estudo do processo de manipulação e controle

aproveitando múltiplos robôs móveis em ambientes abertos, na qual o processo de sensoriamento detecta mudanças na geometria ao redor de cada robô. Há um planejamento de movimento para cada tarefa a ser realizada e os robôs são divididos em grupos de acordo com a habilidade de agir e sentir [Borenstein, 1993] ou abordando a utilização de robôs autônomos em tarefas de transporte [Samah-Abou, 2001], outros utilizando agentes para negociação entre robôs [Jahanbin, 1988]. Os problemas abordados por estes artigos utilizavam a composição de heurísticas e algoritmos de inteligência artificial na condução de uma determinada tarefa.

Na proposta apresentada neste trabalho, utilizou-se um sistema cooperativo de acordo com [Crowley, 1985], contudo aprimorou-se o monitoramento por sensores de deslocamento transversal e rotacional e a passagem de liderança é realizada somente pela comunicação implícita entre os robôs. Agregando a hierarquia líder-seguidor e baseado em [Pereira, 2002], procurou-se aperfeiçoar o planejamento de trajetória modificando-se o critério de retorno ao algoritmo de campo potencial artificial. Nas próximas seções é fornecido um conjunto de conceitos inerentes ao estudo realizado neste trabalho, a fim de fornecer um entendimento dos aspectos matemáticos, de controle e computacionais envolvidos.

3 TÓPICOS DE CONTROLE E PLANEJAMENTO DE TRAJETÓRIA

Nesta seção serão abordados assuntos relativos a sistemas de controle do trabalho realizado e os principais algoritmos utilizados em planejamento de trajetória.

3.1. SISTEMAS DE CONTROLE NÃO LINEARES

Mais tarde neste artigo nós consideraremos um robô não holonômico como um sistema de controle não-linear. Nesta seção nós recordaremos importantes resultados no controle não-linear. Veja [Hermann, 1977] para mais detalhe.

3.1.1. CONTROLABILIDADE

Seja Ω um subconjunto mensurável de \mathbb{R}^m e C uma distribuição conectada de dimensão n . Considera-se um sistema de controle da forma:

$$\dot{q} = f(q, u) \quad (3.1)$$

onde $u \in \Omega$, $q \in C$, e f são suaves como uma função de q . Ω representa o espaço de controle do sistema, i.e., o conjunto de valores de controle admissíveis. C representa o espaço de estado, ou espaço de configuração, i.e., o conjunto de estados distinguíveis que o sistema pode captar a qualquer momento.

Dado um subconjunto $U \subset C$, a configuração $q_1 \in U$ é dita ser U -acessível de $q_0 \in U$ se existir um controle mensurável limitado $u(t)$, $t \in [t_0, t_1]$, tal que a solução $q(t)$ do sistema (3.1) satisfaz: $q(t_0) = q_0$, $q(t_1) = q_1$ e $q(t) \in U$, $\forall t \in [t_0, t_1]$. Escreve-se $q_1 A_U q_0$. O conjunto de pontos U -acessível de q_0 é indicado por $A_U(q_0)$.

O sistema (3.1) é controlável em q_0 se $A_C(q_0) = C$. Se isto é verdade para qualquer estado $q_0 \in C$, então o sistema é controlável.

O sistema (3.1) é localmente controlável em q_0 se para toda vizinhança U de q_0 , $A_U(q_0)$ é também uma vizinhança de q_0 . É localmente controlável se isto é verdade para todo $q_0 \in C$.

A acessibilidade é uma relação reflexiva e transitiva. O fechamento simétrico desta relação é chamado acessibilidade fraca. q' é fracamente U-acessível de q se existe uma seqüência q_0, \dots, q_k tal que $q = q_0$, $q' = q_k$, e também $q_i A_U q_{i-1}$ ou $q_{i-1} A_U q_i$ para todo $i \in [1, k]$. O conjunto de pontos fracamente U-acessível de q é indicado por $WA_U(q)$.

O sistema (3.1) é fracamente controlável em q_0 se $WA_C(q_0) = C$. Se este for o caso, o sistema é necessariamente fracamente controlável em qualquer outra configuração, por causa da acessibilidade fraca é uma relação de equivalência; conseqüentemente, o sistema é fracamente controlável.

O sistema (3.1) é localmente fracamente controlável em q_0 se para toda vizinhança U de q_0 , $WA_U(q_0)$ é também uma vizinhança de q_0 . É localmente fracamente controlável se isto é verdade em todo $q_0 \in C$.

A controlabilidade local (fraca) implica controlabilidade (fraca). Para sistemas simétricos (sistemas cuja relação de acessibilidade é simétrica), Controlabilidade (local) fraca é equivalente a Controlabilidade (local). Portanto, para um sistema simétrico, controlabilidade local fraca implica controlabilidade. abilis1,01(n)32(n)344(TEOREMA DA INTE

Algebra associada com F , $CLA(F)$, é o subespaço de $X(C)$ gerado por todas as combinações lineares dos campos vetoriais em F e todos os seus Lie brackets calculadas recursivamente.

Para todo $q_0 \in C$, seja $CLA(F)(q_0)$ representa o subespaço dos vetores tangentes espalhados pelos campos vetoriais de $CLA(F)$ em q_0 . Uma subdistribuição conectada C' de C é uma subdistribuição integral de $CLA(F)$ se em cada $q \in C'$ o espaço tangente de C' está contido em $CLA(F)(q)$. C' é uma subdistribuição integral máxima de $CLA(F)$ se não é propriamente incluída em qualquer outra distribuição integral.

O Teorema de Frobenius pode ser indicado como segue:

Teorema 1: Se $CLA(F)(q)$ tem dimensão constante k para todo $q \in C$, existe uma partição de C em subdistribuições integrais máximas de todos $CLA(F)$ de dimensão k .

3.1.3. CONDIÇÃO DE CONTROLABILIDADE

O sistema (3.1) é dito satisfazer a condição do posto de controlabilidade em q_0 se a dimensão de $CLA(F)(q_0)$ é exatamente a dimensão n de C . Se isto é verdade para todo $q_0 \in C$, então o sistema é dito satisfazer a condição do posto de controlabilidade.

Os seguintes resultados derivam-se do trabalho de Chow [Chow, 1939]:

Teorema 2: Se o sistema (3.1) satisfizer a condição do posto de controlabilidade em q_0 , é localmente fracamente controlável em q_0 .

Teorema 3: Se o sistema (3.1) é localmente fracamente controlável, a condição do posto de controlabilidade é satisfeito subconjunto aberto e denso de C .

Estes dois teoremas considerados juntos são conhecidos como teorema condição do posto de controlabilidade. Em particular, se considerarmos somente sistemas simétricos para o qual a dimensão de $CLA(F)(q)$ não depende de q , pode-se inferir que um sistema de controle é localmente controlável (consequentemente, controlável) se satisfaz à condição do posto de controlabilidade.

Outra apresentação do teorema da condição do posto de controlabilidade baseado no conceito de distribuição é dado em [Isidori, 1985]. Sua relação para robôs não holonômicos é analisada em [Li, 1989] e [Barraquand, 1989]. Ao contrário desta apresentação, a qual se

aplica somente a confinamentos de igualdade linear, a formulação usada acima permite tratar com confinamentos de igualdade e desigualdade não-linear, como mostrado na seção seguinte.

3.1.4. LIE ALGEBRA

É bem conhecido que conceitos de álgebra linear, bem como teoria de matrizes são essenciais na abordagem no problema de linearização do controle de sistemas não lineares, especificamente quando seu grau relativo não é bem definido. No entanto, a relação entre o controle de sistemas não lineares (isto é, robôs móveis) e a teoria de matrizes não é sempre fácil de entender. Uma das dificuldades inerentes ao planejamento de trajetória de robôs são os algoritmos de controle. Outra dificuldade é relativa ao controle de seguir uma trajetória. Serão abordados os tópicos chave relativos a uma dinâmica não holonômica do robô e sua representação espaço-estado, a linearização controlada de realimentação de entrada e saída, o modelo cinemático, as equações de confinamento do robô móvel, as equações de saída que são essenciais para o algoritmo de controle e a determinação ou existência de graus relativos, além dos conceitos de “Lie brackets” e derivadas de posição [Coelho, 2003].

EQUAÇÕES DINÂMICAS E FORMULAÇÃO TEÓRICA

Considere um robô móvel não holonômico com n coordenadas generalizadas q sujeitas a m confinamentos (assumindo que $m < n$) cujas equações dinâmicas de movimento são descritas por

$$M(q)\ddot{q} + V(q, \dot{q}) = B(q)\tau - A^T(q)\lambda \quad (3.3)$$

onde $V(q, \dot{q}) = C(q, \dot{q})\dot{q}$, $M(q) \in \mathfrak{R}^{n \times n}$ é a matriz inércia, $C(q, \dot{q}) \in \mathfrak{R}^{n \times n}$ é a matriz das forças centrípeta e de Coriolis, $B(q) \in \mathfrak{R}^{n \times (n-m)}$ é a matriz de transformação de entrada, $A^T(q)$ é uma matriz Jacobiana, $\tau \in \mathfrak{R}^{(n-m)}$ é o vetor de entrada e $\lambda \in \mathfrak{R}^m$ é o vetor de forças confinadas. As m equações de confinamento do sistema mecânico pode ser escrita na forma

$$w(q, \dot{q}) = 0 \quad (3.4)$$

Se uma equação de confinamento está na forma $W_i(q) = 0$ é dita holonômica, caso contrário é um confinamento cinemático rotulada não holonômica.

3.2. HOLONOMICIDADE

Holonomicidade é o termo usado para descrever as propriedades locomotivas de um veículo em relação a seu espaço de configuração. Introduzir-se-á uma definição matemática do termo, mas começaremos estabelecendo, em palavras, uma definição: "Um veículo é holonômico se o número de graus de liberdade de movimentos locais igualem o número de graus de liberdade globais."

Nós podemos tornar um pouco mais claro com alguns exemplos:

1. Um triciclo é não-holonômico: os graus de liberdade globais são movimentos em x, y e a orientação, no entanto localmente, um carro pode somente mover-se em uma das direções (x ou y) ou rotacionar em torno de si. Ele não pode deslocar-se lateralmente, até mesmo o giro é acoplado ao movimento, figura 3.1.
2. Um veículo de roda esférica é holonômico: ele pode girar sobre o ponto e transladar instantaneamente em qualquer direção sem ter que rotacionar primeiro.
3. Um trem é holonômico: pode mover-se para frente ou para trás ao longo do trilho o qual é parametrizado por um único grau de liberdade global - à distância ao longo do trilho.

Deve ser óbvio a você que o controle de movimento para um veículo holonômico é muito mais fácil do que para um veículo não-holonômico. Se isto não for óbvio considere a complexidade relativa de estacionar um carro em um espaço apertado comparado a dirigir um veículo que pode simplesmente deslocar-se lateralmente no espaço (um hovercraft, por exemplo).

Nós podemos agora colocar algum formalismo em nossa noção de holonomicidade. No próximo tópico será mostrada em termos matemáticos a identificação de veículos holonômicos e não-holonômicos [Newman, já citado anteriormente].

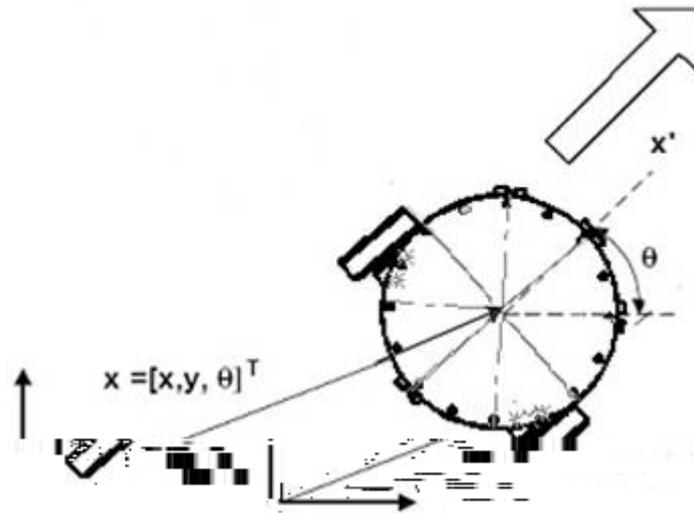


FIG. 3.1. Um veículo não holonômico

3.2.1. CONFINAMENTOS NÃO HOLONÔMICOS

Indicaremos por A o robô composto de corpos rígidos movendo-se em um espaço de trabalho \mathcal{W} . Uma configuração A é uma especificação de posição de todos os pontos em A em relação ao eixo Cartesiano em \mathcal{W} . O espaço de configuração de A é o espaço C de todas as possíveis configurações de A . O espaço de configuração de um sistema mecânico composto de corpos rígidos em uma variação suave. Virtualmente algumas situações práticas, o intervalo das posições alcançáveis pelos corpos do robô pode ser limitado, fazendo C uma distribuição compacta. Seja n a dimensão de C . Nós representamos uma configuração q como uma lista (q_1, \dots, q_n) de n coordenadas generalizadas com módulo apropriado em coordenadas angulares.

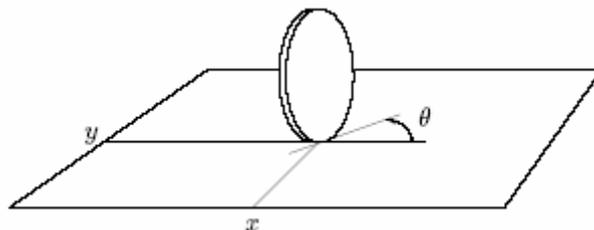


FIG. 3.2. Exemplo canônico de não holonomicidade (disco rolante). Extraído do livro Mobile Robots de J.C. Latombe

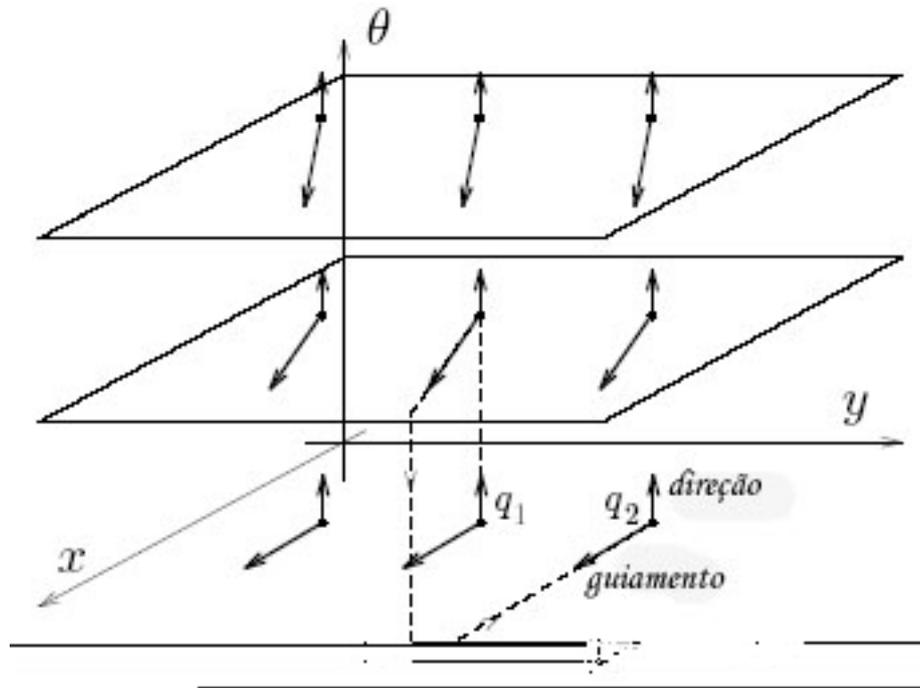


FIG. 3.3. Espaço de configuração do disco rolante.

Suponha que um confinamento escalar da forma:

$$F(q, t) = 0 \quad (3.5)$$

com $q \in C$ e t representando o tempo, aplicam-se ao movimento de A . Suponha além disso que F é suave com derivadas não nulas. Então, na teoria, poderia usar a equação para solução de uma das coordenadas generalizadas em termos das outras coordenadas e do tempo. Assim, a equação (3.3) define uma subdistribuição de dimensão $(n-1)$ de C . Esta subdistribuição está no fato que o real espaço de configuração de A e as $n - 1$ coordenadas remanescentes são suas reais coordenadas generalizadas. Confinamento (3.5) é um confinamento de igualdade holonômico. Mais geral, pode haver k confinamentos da forma (3.5). Se forem independentes, determinam uma subdistribuição de dimensão $(n - k)$ de C , a qual é o real espaço de configuração A .

Um confinamento da forma $F(q, t) < 0$ (ou ≤ 0) age como um obstáculo. Simplesmente determina um subconjunto de C que tem mesma dimensão de C .

Um confinamento da forma (3.5) é somente um confinamento cinemático de um tipo. Outro é um confinamento escalar da forma:

$$G(q, \dot{q}, t) = 0 \quad (3.6)$$

com $\dot{q} \in T_q(C)$, o espaço tangente de C em q . O espaço tangente, o qual representa o espaço das velocidades de A , é um vetor do espaço de dimensão n . Um confinamento da forma (3.6) é holonômico se for integrável, i.e., se \dot{q} puder ser eliminado e a equação (3.1) for reescrita na forma (3.5). Se não, o confinamento é não holonômico. Como visto abaixo, um confinamento não holonômico restringe espaço de velocidades alcançáveis por A em qualquer configuração q em um subespaço linear de dimensão $(n - 1)$ de $T_q(C)$ sem afetar a dimensão do espaço de configuração. Se houver k independentes confinamentos não holonômicos da forma (3.6), o espaço das velocidades alcançáveis é um subespaço de $T_q(C)$ de dimensão $n - k$.

Um confinamento da forma (3.5) restringe o conjunto de velocidades alcançáveis em algumas configurações q para um subconjunto $T_q(C)$ tendo a mesma dimensão de $T_q(C)$. O confinamento que limita o ângulo de direção de um carro é um típico confinamento de desigualdade cinemática.

Quando se trata de confinamentos da forma (3.6), duas questões importantes são levantadas:

1. A questão da integrabilidade: Elas são não integráveis?, isto é, estamos certos que elas são realmente não holonômicas?
2. A questão da controlabilidade: Elas restringem o conjunto de configurações alcançáveis de qualquer configuração dada?

Nós investigamos estas questões nas próximas duas subseções. Por simplicidade, assumiremos que os confinamentos cinemáticos não dependam do tempo. Entretanto, todos os resultados permanecem válidos quando os confinamentos são dependentes do tempo.

3.2.2. CLASSIFICAÇÃO DE SISTEMAS NÃO HOLONÔMICOS

É importante conhecer o tipo de confinamentos de movimento do sistema. Alguns conceitos e formulações matemáticas que permitem alcançar esta proposta serão apresentados. Suponha que há k confinamentos holonômicos e $m-k$ não holonômicos, todos podem ser escritos na forma de

$$A(q)\dot{q} = 0 \tag{3.7}$$

onde $A(q) \in \mathfrak{R}^{m \times m}$ é uma matriz de posto cheio. Seja s_1, \dots, s_{n-m} o conjunto de campos vetoriais suaves (continuamente diferenciáveis) e linearmente independentes no espaço nulo de $A(q)$, $\mathfrak{N}(A)$, i.e.

$$A(q)s_i(q)=0 \quad i = 1, \dots, n - m \quad (3.8)$$

Seja $S(q)$ a matriz de posto cheio composta destes vetores

$$S(q) = [s_1(q) \dots s_{n-m}(q)] \quad (3.9)$$

e Δ a distribuição espalhada por estes campos vetoriais

$$\Delta(q) = \text{span}\{s_1(q), \dots, s_{n-m}(q)\} \quad (3.10)$$

Portanto, segue que $\dim\Delta(q) = \text{posto } S(q)$ e qualquer \dot{q} satisfazendo a equação (3.7) pertence a Δ .

Definição 1: Para dois campos vetoriais f e g , o “Lie bracket” é um terceiro campo vetorial definido por:

$$[f, g](q) = \frac{\partial g}{\partial q} f(q) - \frac{\partial f}{\partial q} g(q) \quad (3.11)$$

É óbvio que $[f, g] = -[g, f]$ e $[f, g] = 0$ para campos vetoriais constantes f e g . Também a identidade Jacobiana,

$$[h, [f, g]] + [f, [g, h]] + [g, [h, f]] = 0$$

A seguinte notação é comumente usada na representação “Lie bracket”:

$$\begin{aligned} ad_f^0 g(q) &= g(q) \\ ad_f g(q) &= [f, g](q) \\ ad_f^k g(q) &= [f, ad_f^{k-1} g](q), \quad k > 1 \end{aligned} \quad (3.12)$$

Definição 2: Uma distribuição Δ é involutiva se é fechada sobre a operação “Lie bracket”, isto é, se

$$g_1 \in \Delta \text{ e } g_2 \in \Delta \Rightarrow [g_1, g_2] \in \Delta$$

Então, analisando se a distribuição Δ é ou não involutiva. Seja Δ^* a menor distribuição involutiva contendo Δ , neste caso $\dim(\Delta) \leq \dim(\Delta^*)$. De acordo com Campion et al, há três possíveis casos: (1) para $k = m$, isto é, todos os confinamentos são holonômicos, Δ é involutivo; (2) para $k = 0$, ou seja, todos os confinamentos são não holonômicos, Δ^* espalha-se pelo espaço inteiro; (3) para $0 < k < m$, os k confinamentos são integráveis e k componentes das coordenadas generalizadas podem ser eliminadas das equações de movimento, no caso anterior $\dim(\Delta^*) = n - k$.

No entanto, pode-se ser mais preciso, distinguindo entre confinamentos holonômicos e não holonômicos. Para verificar o tipo de confinamento é necessário calcular repetidos “Lie bracket” dos campos vetoriais s_1, \dots, s_{n-m} de Δ (ou do sistema $\dot{q}(t) = \sum_{i=1}^{n-m} s_i v_i(t) = S(q)v(t)$).

Como observado por Luca[De Luca, 1995], “O nível de “bracketing” necessário para espalhar R^n está relacionada à complexidade do problema de planejamento do movimento. Por esta razão, dá-se uma classificação de sistemas não holonômicos baseado na sequência e ordem dos “Lie brackets” na correspondente álgebra de acessibilidade.”

Definição 3: A filtragem gerada pela distribuição Δ (3) é definida como a sequência $\{\Delta_i\}$ com

$$\Delta_i = \Delta_{i-1} + [\Delta_1, \Delta_{i-1}], i \geq 2 \quad (3.13)$$

onde

$$\Delta_1 = \Delta$$

e

$$[\Delta_1, \Delta_{i-1}] = \text{span} \{[s_j, \gamma] \mid s_j \in \Delta_1, \gamma \in \Delta_{i-1}\}, j = 1, \dots, n-m$$

Observa-se que $\Delta_i \subseteq \Delta_{i+1}$. Também, da identidade Jacobiana segue que

$$[\Delta_i, \Delta_j] \subseteq [\Delta_1, \Delta_{i+j-1}] \subseteq \Delta_{i+j}$$

Uma filtragem é regular em uma dada vizinhança V de q_0 se $\dim \Delta_i(q) = \dim \Delta_i(q_0)$, $\forall q \in V$.

Para uma filtragem regular, se $\dim \Delta_{i+1} = \dim \Delta_i$, então Δ_i é involutiva e $\Delta_{i+j} = \Delta_i$ para todo $j \geq 0$. Desde que $\dim \Delta_1 = n-m$ e $\dim \Delta_i \leq n$, a condição de término ocorre após m passos, isto é, concorda com o número de confinamentos cinemáticos originais.

Se a filtragem gerada por uma distribuição Δ é regular, é possível definir o grau de não holonomicidade de Δ como o menor inteiro k que verifica a condição $\dim \Delta_{k-1} = \dim \Delta_k$. Observe que a verificação desta condição implica que $k \leq m+1$.

As condições anteriores para holonomicidade, não holonomicidade parcial e não holonomicidade completa pode se reescrita como segue: (1) para $k = 1$, isto é, $\dim \Delta_k = n-m$, todos os confinamentos são holonômicos; (2) para $2 \leq k \leq m$ e se $\dim \Delta_k = n$, todos os confinamentos são não holonômicos; (3) para $2 \leq k \leq m$ e se $(n-m)+1 \leq \dim \Delta_k \leq n$, os confinamentos são parcialmente não holonômicos.

3.2.3. REPRESENTAÇÃO ESPAÇO DE ESTADOS

Considere o sistema mecânico dado por (3.1) e (3.7) e seja k de m confinamentos holonômicos. Desde que a velocidade confinada está sempre no espaço nulo de $A(q)$, é possível definir $n-m$ velocidades $v(t) = [v_1 v_2 \dots v_{n-m}]^T$ tal que para todo t

$$\dot{q} = S(q)v(t) \quad (3.14)$$

A equação anterior representa a cinemática de um sistema mecânico (neste caso um robô móvel), onde $S(q)$ é basicamente uma matriz Jacobiana que converte velocidades de um sistema de coordenadas móveis para velocidades em um sistema de coordenadas cartesianas.

Equações diferenciais (3.7) em relação a t , depois substituindo o resultado (\ddot{q}) na equação (3.1), e finalmente multiplicando o resultado por S^T fornece

$$S^T (MS\ddot{v}(t) + M\dot{S}v(t) + V) = S^T B\tau \quad (3.15)$$

levando em consideração que $S^T A^T \lambda = 0$, desde que a matriz S “span” $\mathcal{N}(A)$. Considerando o vetor de estado

$$x = \begin{bmatrix} q \\ v \end{bmatrix} \quad (3.16)$$

e baseado nas equações (11) e (12) é obtido a equação de estado:

$$\dot{x} = \begin{bmatrix} Sv \\ f_2 \end{bmatrix} + \begin{bmatrix} 0 \\ (S^T MS)^{-1} S^T B \end{bmatrix} \tau \quad (3.17)$$

onde $f_2 = (S^T MS)^{-1} (-S^T M\dot{S}v - S^T V)$. Assumindo que o número de entradas do sistema é maior ou igual a diferença entre o número de coordenadas generalizadas e o número de confinamentos independentes do sistema mecânico ($r \geq n - m$) e que $(S^T MS)^{-1} S^T B$ tem posto $n-m$, a seguinte realimentação não linear pode ser aplicada:

$$\tau = \left((S^T MS)^{-1} S^T B \right)^+ (u - f_2) \quad (3.18)$$

onde $(.)^+$ representa um inversa generalizada de $(.)$. A equação de estado pode ser reescrita da forma:

$$\dot{x} = f(x) + g(x)u \quad (3.19)$$

$$\text{onde } f(x) = \begin{bmatrix} S(q)v \\ 0 \end{bmatrix} \text{ e } g(x) = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

3.2.4. LINEARIZAÇÃO DA REALIMENTAÇÃO ENTRADA-SAÍDA

Considere o seguinte sistema SISO (“single-input single-output”):

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{3.20}$$

onde $x \in \mathbb{R}^n$ representa o vetor de estado, $u \in \mathbb{R}$ é a entrada de controle e $y \in \mathbb{R}$ é a saída. f e g são campos vetoriais (daqui em diante chamadas funções), h é uma função e todas são não lineares e assumidas diferenciáveis.

No problema de linearização da realimentação entrada-saída a questão é encontrar, se existir, uma lei de controle de realimentação de estado $u = \alpha(x) + \beta(x)v$ e uma transformação de variáveis de estado $Z = T(x)$ que transforma o sistema não linear em um equivalente linear. A variável v é uma entrada externa, $\alpha(x)$ e $\beta(x)$ são funções algébricas escalares com $\beta(x) \neq 0$.

A matriz de transformação $T(x)$ deve ser inversível, tal que $x = T^{-1}(z)$ pode ser plausível, e desde que as derivadas de z e x deveria ser contínuo, ambos $T(\cdot)$ e $T^{-1}(\cdot)$ deve ser continuamente diferenciável. Um mapa continuamente diferenciável com uma inversa continuamente diferenciável é conhecida como um difeomorfismo.

Sistemas não holonômicos tem propriedades únicas. Portanto, o sistema (3.9) é controlável se todos seus confinamentos são não holonômicos e seu ponto de equilíbrio $x = 0$ pode ter uma estabilidade Lagrangiana, mas não pode ser assintoticamente estável por uma realimentação de estado suave.

Pode-se estabelecer que um sistema com confinamentos não holonômicos não é de estado de entrada linearizável, e pode ser entrada-saída linearizável se um conjunto apropriado de equações de saída é escolhido. Vale a pena observar que na linearização do estado de entrada, a equação de estado é completamente linearizado, enquanto na linearização entrada-saída, onde o mapeamento entrada-saída é linearizado, a equação de estado pode ser somente parcialmente linearizada.

O conhecimento do grau relativo, ρ , é uma das condições para aplicação dos métodos de linearização de realimentação. O grau relativo do sistema é a menor ordem das derivadas de saída, y , que explicitamente dependem da entrada, u .

Portanto, pela diferenciação de (10), é encontrado que

$$\dot{y} = \frac{\partial h}{\partial x} \dot{x} = \frac{\partial h}{\partial x} [f(x) + g(x)u] \stackrel{def}{=} L_f h(x) + L_g h(x)u \quad (3.21)$$

onde $L_f h(x) = \frac{\partial h}{\partial x} f(x)$ é definida como a derivada de posição de uma função escalar $h(x)$ em relação a uma função vetorial $f(x)$ ou ao longo de $f(x)$. Esta é a noção comum da derivada de h ao longo da trajetória do sistema $\dot{x} = f(x)$. O procedimento com o cálculo do grau relativo é necessário para conhecer o seguinte conjunto de expressões de álgebra de posição:

$$\begin{aligned} L_g L_f h(x) &= \frac{\partial (L_f h)}{\partial x} g(x) \\ L_f^2 h(x) &= L_f L_f h(x) = \frac{\partial (L_f h)}{\partial x} f(x) \\ L_f^k h(x) &= L_f L_f^{k-1} h(x) = \frac{\partial (L_f^{k-1} h)}{\partial x} f(x) \\ L_f^0 h(x) &= h(x) \end{aligned} \quad (3.22)$$

Se $L_g h(x) \neq 0$, então $\rho = 1$. Se $L_g h(x) = 0$, então $\dot{y} = L_f h(x)$ é independente de u .

Calculando a segunda derivada de y , representada por $y^{(2)}$, segue:

$$y^{(2)} = \frac{\partial (L_f h)}{\partial x} [f(x) + g(x)u] = L_f^2 h(x) + L_g L_f h(x)u \quad (3.23)$$

Novamente, se $L_g L_f h(x) \neq 0$, então $\rho = 2$. Se $L_g L_f h(x) = 0$ então $y^{(2)} = L_f^2 h(x)$ que é independente de u . Procedendo o cálculo, a definição do grau relativo ρ é obtido, desde que se $h(x)$ é satisfeito

$$\begin{aligned} L_g L_f^{i-1} h(x) &= 0, \quad i = 1, 2, \dots, r-1 \\ L_g L_f^{r-1} h(x) &\neq 0 \end{aligned} \quad (3.24)$$

então u não aparece nas equações de $y, \dot{y}, \dots, y^{r-1}$, e aparece com um coeficiente não nulo na equação de $y^{(r)}$, isto é, $\rho = r$ e

$$y^{(\rho)} = L_f^\rho h(x) + L_g L_f^{\rho-1} h(x)u \quad (3.25)$$

A equação anterior mostra que o sistema é entrada-saída linearizável e descrito pela equação

$$y(\rho) = v, \quad (3.26)$$

se a seguinte lei de controle é escolhida

$$u = \frac{1}{L_g L_f^{\rho-1} h(x)} [v - L_f^\rho h(x)] \quad (3.27)$$

De (3.26) conclui-se que o sistema linearizado resultante, ou seja, um sistema com entrada v e saída y , é uma cadeia de ρ integradores.

Uma característica importante do método de linearização da realimentação entrada-saída é o fato que decompõe dinâmicas de sistemas não lineares em uma parte externa e interna. Desde que a parte externa consiste em uma relação linear entre y e v (ou equivalentemente, a forma canônica de controlabilidade entre y e u) é fácil projetar a entrada v tal que a saída y comporte-se como desejado. Então a questão é se as dinâmicas internas também comportar-se-ão bem ou não, ou seja se os estados internos permanecerão limitados. Desde que o projeto de controle deve levar em conta a dinâmica completa (e portanto não pode tolerar a instabilidade de dinâmicas internas), o comportamento interno foi endereçado cuidadosamente. O problema de instabilidade é geralmente superado por meio do uso da dinâmica zero do sistema. A dinâmica zero pode ser alcançada pela consideração que as entradas e condições iniciais do sistema são escolhidas de tal maneira que a saída do sistema é identicamente zero.

A linearização de entrada-saída, é baseada na aplicação da transformação ($Z = T(x)$), que permite o sistema transformado ser apresentado como:

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= z_3 \\ &\vdots \\ \dot{z}_{\rho-1} &= z_\rho \\ \dot{z}_\rho &= v \\ \dot{z}_{\rho+1} &= q_{\rho+1}(z) \\ &\vdots \\ \dot{z}_n &= q_n(z) \\ y &= z_1 \end{aligned} \quad (3.28)$$

onde

$$z = T(x) = \begin{bmatrix} T_1(x) \\ T_2(x) \\ \vdots \\ T_\rho(x) \\ T_{\rho+1}(x) \\ \vdots \\ T_n(x) \end{bmatrix} = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{\rho-1} h(x) \\ T_{\rho+1}(x) \\ \vdots \\ T_n(x) \end{bmatrix} \quad (3.29)$$

De (3.28) é verificada que as últimas $n - \rho$ equações não são observáveis por meio da saída; eles são, no entanto, necessário para garantir que eles são estáveis. As funções $T_{\rho+1}(x)$, ..., $T_n(x)$ pode ser escolhido arbitrariamente, contanto que é garantido que $T(x)$ é um difeomorfismo. Particularmente, baseado em (3.28), é possível alcançar as funções $T_{\rho+1}(x)$, ..., $T_n(x)$, tais que as últimas $n - \rho$ equações são independentes da entrada u . Do sistema (3.28) pode-se verificar que as primeiras ρ equações exibem um comportamento entrada-saída de um integrador de ordem ρ .

Duas situações devem ser distinguidas: (1) quando $\rho = n$, tem-se uma linearização do estado de entrada e o estado e equações de saída representam um sistema linear; (2) quando $\rho < n$, tem-se uma linearização entrada-saída, e algumas variáveis de estado exibirão uma dinâmica não linear.

3.2.5. CONFINAMENTOS CINEMÁTICOS E SISTEMAS DE CONTROLE

Vamos considerar um conjunto de $k < n$ confinamentos cinemáticos independentes da forma (3.30):

$$G(q, \dot{q}) = (G^1(q, \dot{q}), \dots, G^k(q, \dot{q})) = (0, \dots, 0) \quad (3.30)$$

Para cada q , $G_q = G(q, \cdot)$ define uma função de $T_q(C)$ em \mathbb{R}^k . Como os k confinamentos são independentes, o Jacobiano desta função tem posto cheio. O subconjunto do espaço tangente comprova que os confinamentos são simplesmente $G_q^{-1}(0, \mathbb{K}, 0)$. De acordo com o teorema da função implícita, este subespaço é uma subdistribuição de $T_q(C)$ de dimensão $n - k$. Mostramos um mapa destas distribuições por $u = (u_{k+1}, \dots, u_n)$, e definimos $f_q = u^{-1}$. Obtêm-se:

$$\dot{q} = f_q(u) = f(q, u) \quad (3.31)$$

De acordo com a consideração adicional que f é suave como uma função de q , esta relação localmente define um sistema de controle não linear com $n - k$ variáveis de controle (u_{k+1}, \dots, u_n). Assumimos que fixaremos confinamentos desiguais em adição aos confinamentos iguais mencionados acima. Estes novos confinamentos são transformados em confinamentos desiguais aplicando-se as variáveis de controle por meio da inversa da lista u . Eles definem a forma do conjunto Ω de variáveis de controle admissíveis.

Reciprocamente, se considerarmos qualquer sistema de controle do tipo (3.1) tal que $f(q, \dot{q}) = f_q$ tem posto cheio como uma função da variável de controle $u = (u_{k+1}, \dots, u_n)$, então podemos aplicar novamente o Teorema da Função Implícita para todo q e obtermos uma lista $G_q = (G_q^1, \dots, G_q^n)$ verificando:

$$\begin{cases} \forall i \in [1, k]: & G_q^i(f_q(u)) = G_q^i(\dot{q}) = G^i(q, \dot{q}) = 0 \\ \forall i \in [k+1, n]: & G_q^i(f(u)) = G_q^i(\dot{q}) = u_i \end{cases} \quad (3.32)$$

As primeiras k igualdades precisamente definem os k confinamentos cinemáticos independentes. Além disso, as desigualdades nas variáveis de controle que definem a forma do conjunto Ω são transformadas em desigualdades em velocidade por meio de $G_q^i, i \in [k+1, n]$.

Portanto, em geral, um robô sujeito a k confinamentos cinemáticos independentes é localmente equivalente a um sistema de controle com $n - k$ variáveis de controle para as quais a função f tem posto cheio em u . Além disso, qualquer confinamento desigual adicional na velocidade é equivalente a um confinamento desigual na variável de controle.

3.2.6. NÃO HOLOMICIDADE E CONTROLABILIDADE

Considere um robô sujeito a k confinamentos cinemáticos iguais independentes da forma (3.6). Para responder a questão da integrabilidade, primeiramente, calcula-se o sistema de controle equivalente, que é a função $f(q, u)$, como indicado acima.

Podemos caracterizar a integrabilidade dos confinamentos usando o Teorema de Frobenius. Para cada configuração q , a dimensão r de $CLA(F)(q)$ é claramente maior ou igual a $n - k$. Se r toma valores maiores que $n - k$, então as distribuições de integral máxima tem uma dimensão maior que $n - k$, e os confinamentos são não integráveis, conseqüentemente, não holonômicos.

Por outro lado, se r é igual a $n - k$ para todo q , então o Teorema de Frobenius requer as distribuições de integral máxima de $CLA(F)$ tenham dimensão $n - k$. Logo, as configurações admissíveis do robô giram em uma subdistribuição de C de dimensão $n - k$. Como uma consequência, as velocidades sempre pertencem ao espaço tangente desta subdistribuição, a qual é precisamente $CLA(F)(q)$. Mas, estas mesmas velocidades também pertencem a uma subdistribuição S de dimensão $n - k$ do espaço tangente de C em q definido pelos confinamentos. Por conseguinte, S é necessariamente igual a $CLA(F)(q)$, consequentemente, linear. Isto implica que os k confinamentos de igualdade são lineares em \dot{q} . Portanto, caracterizam confinamentos holonômicos, podemos limitá-los por si só os quais são lineares nos parâmetros de velocidade. Neste caso, pode-se substituir o espaço de configuração C pela subdistribuição de integral máxima passando pela configuração inicial do robô e, livrar-se dos confinamentos. As equações definindo esta subdistribuição podem ser escritas localmente na forma (3.4), isto é, $F(q) = 0$, a qual é a forma integral dos confinamentos (3.30). Pela derivada desta última equação como uma função do tempo, encontram-se novamente os confinamentos em velocidade $dF(q)\dot{q} = 0$, os quais dão uma explicação mais intuitiva do fato de os confinamentos holonômicos serem necessariamente lineares em parâmetros de velocidade [Barraquand, já citado anteriormente].

Em suma:

Proposição 1: Confinamentos cinemáticos que são propriamente não lineares como função da velocidade são necessariamente não holonômicos.

Proposição 2 (Caracterização de holonomicidade): Um robô sujeito a k confinamentos iguais independentes da forma (3.6) é holonômico se a codimensão $n - r$ da Lie Algebra é igual ao número k de confinamentos. Em tal caso, os confinamentos cinemáticos são necessariamente lineares em parâmetros de velocidade.

A resposta à questão de controlabilidade para robôs sujeitos a confinamentos cinemáticos é uma consequência direta do Teorema de Controlabilidade pela Condição do Posto. Como delineado acima, dado k confinamentos independentes, considera-se o sistema de controle equivalente com $n - k$ variáveis de controle. Então se pode analisar a dimensão r da Lie Algebra. Se este número r é constante e igual a n , logo o sistema (isto é, o robô) é localmente fracamente controlável, então r igual a n em um subconjunto denso aberto de C .

Proposição 3 (Caracterização de controlabilidade): Um robô sujeito a confinamentos cinemáticos em velocidade – os quais podem ser lineares ou não lineares, iguais ou desiguais – é, localmente, fracamente controlável se a dimensão r da Lie Algebra é máxima, isto é, igual a dimensão n do espaço de configuração.

Após o estudo prévio, será descrito, brevemente, a conceituação dos métodos de navegação mais comuns no desenvolvimento de sistemas robóticos.

3.3. MÉTODOS DE NAVEGAÇÃO PARA ROBÔS

Conforme descrito anteriormente, a navegação permite aos robôs móveis uma livre desenvoltura pelo seu ambiente de trabalho, ora alcançando metas, ora desviando de obstáculos. Isto é obtido através de sistemas de navegação, presentes em todos os robôs deste tipo, algumas vezes diferentes quanto à forma de executar o seu trabalho, mas chegando objetivamente a um dos dois pontos: na geração de uma trajetória, ou no rastreamento de uma trajetória. Autores [Spence e Hutchinson, já citados anteriormente] ainda admitem que esta maneira de dividir o problema é acarretado pela diferença entre os pesquisadores em planejamento de movimento e em teoria de controle. Segundo os autores existem aqui duas situações que podem ser devidamente separadas. A primeira é o planejamento da trajetória a ser seguida pelo robô, baseado principalmente na leitura dos obstáculos presentes no local. Já a segunda é o controle desta trajetória evitando o choque e preservando a integridade do robô.

Com isto, a geração de uma trajetória está mais voltada ao mundo externo, não considerando características cinemáticas ou dinâmicas do robô, o que não ocorre com os modelos apresentados para o rastreamento de trajetória adotado.

Sob outro aspecto, a navegação também é dividida por alguns autores em duas tarefas básicas: a de se localizar e a de evitar obstáculos [Dam, já citado anteriormente]. Pode-se perceber que a semântica da localização é referenciada como um controle de retorno para o operador, além de servir como um medidor de desempenho no alcance de uma meta preestabelecida, assim, o robô saberá o quão próximo ou distante está do seu objetivo.

É inegável que muitas das técnicas adotadas no planejamento de um sistema de navegação têm elevado embasamento em sistemas biológicos, sendo estes últimos responsáveis por várias heurísticas empregadas neste campo. Isto é plenamente aceitável pois os sistemas biológicos têm demonstrado que através dos anos de evolução das espécies, têm encontrado modos extremamente eficientes de resolver problemas [Dev, já citado anteriormente]. Basear as ações dos robôs em ações típicas de seres vivos é uma excelente técnica no sentido de implementar comportamentos inteligentes [Mcfarland, já citado anteriormente].

No caso de robôs que operam em locais abertos, como o proposto neste trabalho, alguns pontos na determinação do modelo de navegação devem ser levados em consideração, como [Brumitt, já citado anteriormente]:

- a) a incerteza do meio ambiente – o sensoriamento deve ser simultâneo ao translado do robô, uma vez que é impossível ter um caminho pré-gravado;
- b) a segurança do robô – se acontece uma falha do sistema de controle, o robô deve parar imediatamente, evitando qualquer choque;
- c) o tempo computacional na resposta à situação – deve ser rápido o suficiente para que, com o aumento da velocidade do robô, seja possível ainda ter um sistema de navegação confiável;
- d) a complexidade do terreno – a capacidade de trabalhar com as imperfeições do terreno sem acarretar em elevação do custo computacional é de primordial importância, e
- e) a dinamicidade – se a velocidade é razoável, variáveis dinâmicas e cinemáticas devem ser levadas em consideração.

Dentre os modelos de navegação existentes para robôs móveis podem-se definir três grandes grupos, baseados, cada um: em cálculo de posição, em balizas e por último, em mapas e modelos do ambiente [Mckerrow e Borenstein, já citados anteriormente].

A navegação por cálculo de posição é aquela em que o robô recebe uma trajetória para ser executada e, por meio de odometria (sensores que demonstram a rotação das rodas) ou acelerômetros, calcula a sua posição instantânea, avaliando o erro existente. Como em um sistema de otimização, a minimização deste erro aproximará o robô de trajetória desejado.

A baseada em balizas é aquela que, por meio de sensores, possibilita dizer ao robô se ele está no caminho certo. No caso do uso de balizas utilizando emissores de infravermelho ou ondas de rádio, a triangulação entre dois sensores e o robô permite o cálculo da sua posição, conforme a figura 2.6. Também existem aplicações hoje empregando o Sistema de Posicionamento Global (*Global Positioning System*) – GPS que, por meio do cálculo da distância entre o robô em uma série de satélites geo-estacionários, é capaz de localizar o robô na Terra com precisão de metros.

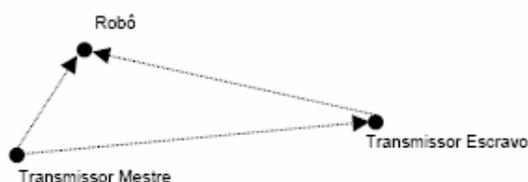


FIG. 3.4. Triangulação entre balizas e robô

Estes dois primeiros modelos de navegação, os quais não impedem do robô de se chocar com algum obstáculo, não serão abordados nesta dissertação, sendo uma excelente referência

a obra *Navigating mobile robots: systems and techniques*. Os métodos baseados em mapas e modelos do ambiente hoje estão divididos basicamente em dois subgrupos, os de planejamento de trajetória e os de uso de sensores [Aurenhammer, 1991], conforme descrito a seguir.

3.3.1. PLANEJAMENTO DE TRAJETÓRIA – HEURÍSTICAS RELATIVAS AO PROBLEMA

O problema de planejamento de trajetória pode ser definido como a busca de um percurso a ser seguido, ou a seqüência de ações a serem tomadas para que o robô possa, saindo de um ponto de partida, chegar a um ponto meta, evitando a colisão com um conjunto de obstáculos conhecidos [Schalkoff e Jahanbin, já citados anteriormente].

O método de planejamento de trajetória é dividido em duas abordagens: global e local, que devem ser utilizados simultaneamente para que se possa ter um melhor desempenho do robô em um ambiente real.

O planejamento global é responsável pelo mapeamento do ambiente onde está inserido o robô em um modelo simplificado, estático, pré-gravado [Crowley, já citado anteriormente], que lhe permite traçar um caminho mesmo por um local que não consiga perceber com seus sensores, seja por estar fora do alcance ou por estar obstruído por algum obstáculo [Firby, já citado anteriormente]. Já o local é responsável pela navegação curta, baseada nos valores coletados pelos seus sensores. Uma vez modelado o ambiente, o planejamento local será o responsável pela localização do robô neste mundo simplificado e dinâmico, devido à constante atualização dos dados não fornecidos pelo modelo global. Uma maneira de diminuir o custo computacional requerido no tratamento destes modelos é utilizar polígonos na representação dos objetos. Esta representação é aceitável pois, como se deseja que o robô não se aproxime muito do objeto, os seus pequenos detalhes são irrelevantes, sendo as aproximações geométricas, por meio de linhas retas, uma excelente solução.

Existem diferentes técnicas matemáticas para se alcançar o planejamento de trajetória, sendo os *Roadmaps*, as Decomposições em Células e o Campo Potencial Artificial as principais. Entre os desenvolvimentos mais recentes na área de Campos Potenciais estão a Navegação por Campo de Força Virtual (VFF) e a Navegação por Histograma de Campo

Vetor (VFH), ambos desenvolvidos na Universidade de Michigan [Koren e Borenstein, já citados anteriormente].

A Navegação por Campo de Força Virtual é baseada em duas etapas. A primeira é definir um histograma bidimensional, onde cada ponto - chamado de célula – contém uma certeza de existir um obstáculo c_{ij} . Este histograma passa então a ser atualizado a partir das informações fornecidas pelos sensores, que verificam a certeza ou não da existência de obstáculo em cada célula. Assim, as que representarem os obstáculos, terão valores maiores de certeza do que as que não representarem.

O próximo passo é aplicar o conceito de campo potencial, onde as células que recebem valores de certeza de um obstáculo exercem forças repulsivas sobre o robô, ao mesmo tempo em que forças atrativas são exercidas pelas células que representam a meta a ser alcançada. A resultante destas forças fornece a direção a ser tomada pelo robô.

O método de Navegação por Histograma de Campo Vetor tenta resgatar uma falha do modelo anteriormente exposto: a perda de informação quando se passa do histograma para o campo de forças. O método consegue evitar este problema por meio de uma redução de dados em dois estágios. Assim, utilizam-se três níveis para se determinar a posição a ser seguida, ao invés de apenas dois como no modelo de Campo de Força Virtual. O primeiro nível é o histograma utilizado e atualizado tal como no método anterior; o segundo é a transformação do histograma cartesiano em polar, centrado no próprio robô, dividido em n setores angulares, onde cada qual tem uma densidade polar de obstáculos, calculada a partir da disposição dos dados no histograma cartesiano. O terceiro nível é a definição da direção a ser tomada pelo robô, que passa a ser calculada através do mapeamento das densidades polares de obstáculos.

3.3.1.1. GRAFO DE VISIBILIDADE

Esta técnica é um dos métodos de planejamento pioneiros, aplica-se a espaços de configuração bidimensional com uma região C de obstáculo poligonal. O grafo de visibilidade é um grafo G não direcional cujos nós são as configurações inicial e final e os vértices da região de obstáculos C . As conexões de G são todas as linhas retas que não interseccionam a região de obstáculos C . O subgrafo de G o qual é restrito aos vértices da região de obstáculo C determina o roadmap R . Outras conexões em G são as que conectam as configurações

inicial e objetivo do roadmap. Um algoritmo complexo compara todos os pares de vértices de G e, verifica se uma linha ligando-os interseccionam com qualquer das bordas dos obstáculos. Se a linha não interseccionam qualquer uma das bordas e o par em consideração não pertence à região de obstáculo C , então eles são bordas visíveis. Para um conjunto de obstáculos com n vértices no total, cada vértice pode fazer par com $(n - 1)$ vértices. Consequentemente, a complexidade é $O(n^2)$. Cada par de pontos deve ser comparado com pelo menos n bordas (um conjunto de obstáculos com n vértices tem n bordas). Isto dará uma complexidade $O(n^3)$ do tempo total. Usando árvores de busca binária, esta complexidade pode ser reduzida a $O(n^2 \log n)$.

Dado um grafo de visibilidade, nós podemos encontrar o caminho mais curto entre as configurações inicial e objetivo usando o algoritmo de caminho mais de Dijkstra. Este método não é aplicável a três dimensões e não é adequado a sistemas não-holonômicos. Um empecilho principal deste método é que a conformidade para limitações, tais como trajetória não suave, limitação da velocidade, aceleração e limites dinâmicos, são ignorados. Consequentemente, o caminho gerado deve ser modificado em um último estágio para considerar estas limitações. No entanto, o caminho gerado pode ser ótimo em casos bidimensionais.

Portanto o método do grafo de visibilidade é aplicável somente para ambientes bidimensionais com obstáculos poligonais. Opera-se em um espaço livre ou em um conjunto de operação segura, se ele é descrito por expressões semi-algébricas lineares bidimensionais. Em espaços com dimensões maiores do que 2, uma linha visível pode ser qualquer segmento de linha reta entre as bordas dos obstáculos que não interseccionam os obstáculos. Como o número de tais segmentos pode ser arbitrariamente grande, o método pode exigir primeiro selecionar algumas linhas visíveis do conjunto de todos os segmentos de linha. Neste caso, portanto, a otimização de trajetória solução não será garantida [Aurenhammer, já citado anteriormente].

3.3.1.2. CAMPO POTENCIAL ARTIFICIAL (CPA)

Esta abordagem foi introduzida por Khatib, para braços de manipuladores robóticos e mais tarde sugerida para plataformas de robôs móveis por J.C. Latombe [Latombe, 1991]. Extraído do conceito da teoria de campo da física, este método modela obstáculos como

emissores de uma força repulsiva e o ponto objetivo como emissor de uma força atrativa em nosso robô. A navegação é desempenhada pelo movimento do robô de maneira a minimizar a energia potencial. Os aspectos principais da abordagem são:

1. Ambiente: Esta abordagem supõe o conhecimento do tipo de obstáculos no ambiente e na fase de planejamento estes obstáculos conhecidos são aproximados por polígonos ou esferas. O ambiente para a formulação original desta idéia foi suposto para ser estática, porém houve algumas adaptações do mesmo para usar esta abordagem para ambientes dinâmicos. Os potenciais são associados aos objetos no ambiente tal como ele é encontrado.

2. Modelo do sistema: A abordagem é quase independente do modelo de sistema e de fato não faz realmente nenhum planejamento inteligente dado o conhecimento dos confinamentos do veículo. Para exemplo: No caso de um robô não holonômico, esta abordagem nunca recomenda realmente que o robô retorne para alcançar o objetivo ao invés de tentar preferivelmente exercer uma força lateral no robô que assintoticamente deveria levar o robô ao objetivo. Isto pode conduzir ao comportamento tal como o robô espirala no objetivo se a orientação do robô quando começa aproximar-se não for correta.

3. Objetivo: Este é idealmente projetado para trabalhar com somente um único objetivo. Se mais do que um objetivo for requerido existirá uma pilha de objetivos que é esvaziada a cada vez que o objetivo atual seja alcançado. Não há mudança alguma no comportamento se uma seqüência dos marcos for especificada.

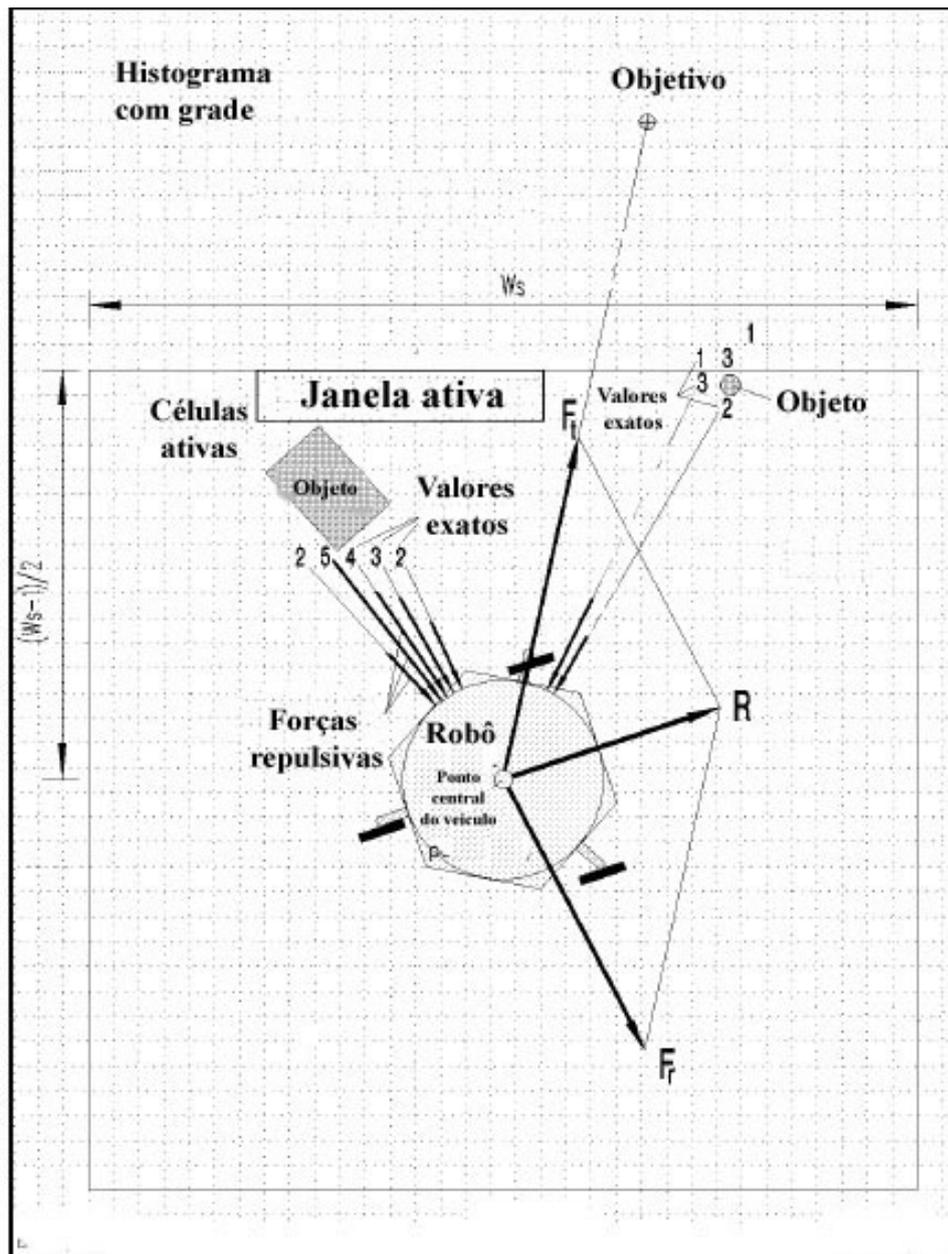


FIG. 3.5. Representação do campo potencial artificial utilizado. Extraído do artigo Potential Field Methods de J. Borenstein e Y. Koren.

Para este tipo de geração de ação de controle, o inteiro conhecimento do ambiente e os confinamentos necessitam ser incorporados ao projeto do sistema. As várias variantes da abordagem do campo potencial artificial foram desenvolvidas desde o tempo que o pensamento original foi publicado, nas tentativas de fazer a abordagem mais útil em ambientes dinâmicos ou desordenados ou parcialmente conhecidos. Um tutorial excelente em campos potenciais é apresentado por Goodrich [Goodrich, 2004]. Uma abordagem clássica de planejamento de trajetória usando campos potenciais é apresentada em Beard et al [Beard,

2003]. A abordagem mais recente do CPA incorpora a realimentação dinâmica de sensores no controle do robô e conseqüentemente supera as limitações de ser computacionalmente pesada e incapaz de reagir aos obstáculos inesperados no ambiente que as abordagens baseadas na otimização sofrem. A teoria do CPA indica que para qualquer robô dirigido ao objetivo em um ambiente que contenha obstáculos estacionários ou móveis, um mapa do CPA pode ser formulado e calculado, levando em conta um pólo atrativo na posição do objetivo do robô e superfícies repulsivas dos obstáculos no ambiente. O campo potencial experimentado pelo robô pode ser expresso como segue:

$$U_{cpa}(x) = U_{objetivo}(x) + U_{obs}(x)$$

onde $U_{cpa}(x)$, $U_{objetivo}(x)$ e $U_{obs}(x)$ denotam o campo potencial artificial, o potencial atrativo do objetivo, e o potencial repulsivo dos obstáculos respectivamente. x indica as coordenadas operacionais que descrevem a posição e a orientação do robô.

Tipicamente, os obstáculos são tratados como corpos exponencialmente repulsivos, que é dito experimentar repulsão pelas ascensões do agente enquanto aproxima da fronteira do obstáculo no qual aponta a força essencialmente torna-se infinita. Para a coesão das equipes permitindo que as equipes inteiras movam-se para frente contudo não funcionando em se um campo potencial típico usado por

$$U_{obs}(x) = \log|r_j| + 1/r_j^2$$

onde r_j é a distância entre o robô e j -ésimo vizinho. O objetivo é escolhido tipicamente ter uma força atrativa de forma parabólica tal como:

$$U_{objetivo}(x) = K|x - x_{objetivo}|^2$$

O potencial é calculado em cada ponto ao longo da trajetória e o robô move-se em direção ao gradiente descendente do campo potencial até alcançar um mínimo local ou global. O método foi aplicado à navegação de robôs móveis especificamente em [Rimon, 1992]. Esta abordagem é simples de estender a espaços de trabalho de dimensões mais elevadas e mais novas idéias permitem adaptar dinamicamente o campo aos ambientes mutáveis. O projeto da função potencial é significativo quando o método é incompleto se existir mínimos locais no ambiente. A maioria de projetos são heurísticos, porém as funções resultantes são frequentemente funções de Lyapunov que dão uns mínimos globais originais.

3.3.1.3. MÉTODOS DE ROADMAP

Os métodos de roadmap baseiam-se na teoria de grafos objetivando criar caminhos no espaço de trabalho livre de obstáculos e, dentre os métodos de roadmap, os descritos nas seções seguintes são comumente implementados em Robótica.

3.3.1.3.1. DIAGRAMA VORONOI

O diagrama de Voronoi é uma versátil estrutura geométrica. O diagrama de Voronoi tem aplicações em física, astronomia, robótica e em muitos outros campos. É também intimamente ligada à outra importante estrutura geométrica, a conhecida triangulação Delaunay. Desde a década de 80, o planejamento de trajetória tem sido uma área de intensos estudos em robótica e geometria computacional. O problema do planejamento de trajetória é encontrar o caminho livre de colisão para um robô em um ambiente repleto de obstáculos. O diagrama de Voronoi é baseado na noção topológica de retração, correspondente a uma estrutura esqueleto que é obtida por redução do espaço livre. Essa redução ocorre pela remoção repetida de uma fina faixa da fronteira do espaço livre. As faixas são removidas até que reste uma estrutura unidimensional.

O mapa do caminho resultante tem a seguinte propriedade, para um dado ponto no espaço livre, considere o ponto mais próximo ou os pontos na região de obstáculos. Para qualquer ponto no mapa do caminho de Voronoi há dois ou mais pontos próximos à região de obstáculos. Em outras palavras, o mapa do caminho de Voronoi obtém soluções para as quais o caminho que o robô percorre tem o máximo espaço livre [Fortune, 1987].

Planejamento de movimento é um problema fundamental em robótica. Aparece em uma variedade de formas, mas a versão mais simples é a seguinte: dado um sistema robótico B , o qual pode consistir em alguns objetos rígidos conectados entre si através de diversas juntas, ligas e conexões ou mover-se independentemente em um ambiente bidimensional ou tridimensional V repleto de obstáculos. Assume-se que a forma e a localização dos obstáculos e as dimensões de B são conhecidas pelo sistema de planejamento. Dada uma configuração inicial Z_1 e uma final Z_2 de B , deseja-se determinar se existe um movimento para evitar

colisão de Z_1 a Z_2 , e se existir, planejar tal movimento. Nesta configuração simplificada e puramente geométrica, ignoram-se tópicos tais como informação incompleta, confinamentos não-holonômicos, tópicos de controle relacionados à inexatidão no sensoriamento e movimento, obstáculos não estacionários, otimização do movimento planejado, etc.

Desde a década de 80, o planejamento de trajetória tem sido uma área de intensos estudos em robótica e geometria computacional. O planejamento de trajetória, com um pequeno número de graus de liberdade, dois ou três, tem métodos de solução mais eficientes tais com:

Curva ou arco Jordan: a imagem de um intervalo unitário fechado sobre um mapeamento bijetor contínuo em um plano. Uma curva fechada de Jordan é a imagem do círculo unitário sobre um mapeamento similar, e uma curva não limitada de Jordan é uma imagem de um intervalo unitário aberto que separa o plano;

Algoritmo aleatório: um algoritmo considere algoritmos que sempre terminam e produzem uma saída correta, mas cujo tempo de execução é uma variável aleatória que depende de aleatoriedade internas;

Soma de Minkowski: para dois conjuntos planos (ou espaciais) A e B, sua soma de Minkowski é o conjunto $A + B = \{x + y | x \in A, y \in B\}$;

Posição geral: a entrada a um problema geométrico é dita ser de posição geral se nenhuma identidade algébrica não trivial com coeficientes inteiros permanece entre os parâmetros que especificação a entrada. Por exemplo, nenhum dos três pontos de entrada deveria ser colinear, nenhum dos quatro pontos cocircular, nenhuma das três linhas concorrente, etc;

Função de distância convexa: uma região convexa B que contenha a origem em seu interior induz uma função de distância convexa d_B definida por $d_B(p,q) = \min\{\lambda | q \in p + \lambda B\}$;

O diagrama de Voronoi de um conjunto de locais particiona o espaço em regiões, um por local. A região de um local s consiste em todos os pontos próximos a s do que qualquer outro ponto. O dual do diagrama de Voronoi, a triangulação Delaunay, é a única triangulação tal que a circunferência circunscrita aos triângulos não contém locais em seu interior. Os diagramas de Voronoi e a triangulação de Delaunay são os tópicos centrais em geometria computacional. Na próxima seção discutiremos a definição e as propriedades básicas no caso de locais em R^d com métrica Euclidiana.

3.3.1.3.1.1. DEFINIÇÕES E PROPRIEDADES BÁSICAS

Define-se a distância euclidiana entre dois pontos p e q por $\text{dist}(p,q)$. No plano tem-se:

$$\text{dist}(p,q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \quad (3.33)$$

Seja $P = \{p_1, p_2, \dots, p_n\}$ o conjunto de n pontos distintos no plano, estes pontos são ditos locais. Define-se o Diagrama de Voronoi de P como a subdivisão do plano em n células, uma para cada local em P , com a seguinte propriedade, um ponto q pertence à célula correspondente ao local p_i se e somente se $\text{dist}(q,p_i) < \text{dist}(q,p_j)$ para cada $p_j \in P$ com $j \neq i$. Sendo designado o Diagrama de Voronoi por $\text{Vor}(P)$ e abusando da terminologia, às vezes utiliza-se $\text{Vor}(P)$ ou Diagrama de Voronoi para indicar somente as bordas e vértices da subdivisão. Por exemplo, quando se diz que um Diagrama de Voronoi é conectado, significa que a união de suas bordas e vértices formam um conjunto conectado. A célula de $\text{Vor}(P)$ que corresponde ao local p_i é representado por $C(p_i)$ (célula de Voronoi).

Aprofundando nosso estudo, primeiramente, viu-se a estrutura de uma simples célula de Voronoi. Para dois pontos p e q no plano, define-se o bissetor de p e q como o bissetor perpendicular ao segmento \overline{pq} . Este bissetor divide o plano em dois semi-planos, um que contém p , caracterizado por $h(p,q)$ e o semi-plano que contém q , caracterizado por $h(q,p)$. Observa-se que $r \in h(p,q)$ se e somente se $\text{dist}(r,p) < \text{dist}(r,q)$, desta observação deriva-se que $C(p_i) = \bigcap_{1 \leq j \leq n, j \neq i} h(p_i, p_j)$.

Portanto, $C(p_i)$ é a intersecção de $n - 1$ semi-planos e, conseqüentemente, uma região poligonal convexa aberta (possivelmente não limitada) limitada por no máximo $n - 1$ vértices e no máximo $n - 1$ bordas.

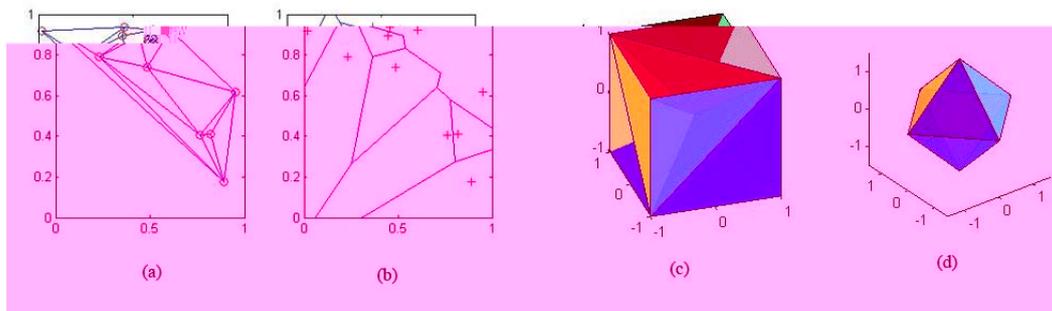


FIG. 3.6. Triangulação Delaunay e Diagrama de Voronoi do mesmo conjunto de locais em duas dimensões (a e b) e três dimensões (c e d).

Definições: Seja S um conjunto finito de pontos em \mathbb{R}^d , ditos locais. A face Voronoi de um local $s \in S$ é o conjunto de todos os pontos de \mathbb{R}^d estritamente mais próximos de s do que qualquer outro local em S . A face Voronoi de um local é sempre subconjunto não vazio, aberto, convexo e de dimensão completa de \mathbb{R}^d . Mais genericamente, para T um subconjunto não vazio de S , a face Voronoi $V(T)$ é o conjunto de pontos de \mathbb{R}^d equidistantes de todos os membros de T e mais próximos a qualquer membro de T do que qualquer membro de $S \setminus T$. O Diagrama de Voronoi de S é a coleção de todas as faces não vazias $V(T)$, para $T \subseteq S$.

Em duas dimensões (figura 3.6 (a)), a face Voronoi de um local é o interior de um polígono convexo possivelmente infinito; seus limites consistem em bordas Voronoi equidistantes de dois locais e vértices Voronoi equidistantes de pelo menos três locais. A figura 3.6 (c) mostra um diagrama de Voronoi em três dimensões.

Para T um subconjunto de S , uma face Delaunay $D(T)$ é definida sempre que houver uma esfera com todos os locais de T com todos os outros locais exteriores, então $D(T)$ é o interior da estrutura convexa de T . Por exemplo, em duas dimensões (figura 3.6 (b)), um triângulo Delaunay é formado por três locais cuja circunferência circunscrita está vazia e uma borda Delaunay conecta dois locais que tem uma circunferência circunscrita vazia. A triangulação Delaunay forma uma célula complexa particionando a estrutura convexa de S .

Há uma correspondência um para um entre o Diagrama de Voronoi e a triangulação Delaunay (a face Voronoi é mapeada na face Delaunay). Esta correspondência tem a propriedade que a soma das dimensões de $V(T)$ e $D(T)$ é sempre d . Portanto, em duas dimensões $V(T)$ é um vértice Voronoi se $D(T)$ é uma região poligonal aberta; $V(T)$ é uma borda se $D(T)$ também o é; $V(T)$ é uma região poligonal aberta se $D(T)$ é um vértice, isto é, um local.

Um diagrama Voronoi é o conjunto de todas as configurações cuja distância mínima para uma região de obstáculo C é alcançada com pelo menos dois pontos na fronteira da região de obstáculo C . Quando os obstáculos são polígonos, o diagrama Voronoi consiste em segmentos retos e parabólicos. As configurações iniciais e objetivo são retratados pelo diagrama Voronoi e o caminho resultante é uma concatenação da linha reta do ponto inicial ao objetivo, o caminho ao longo do diagrama e a linha reta do diagrama ao objetivo. A abordagem é mais bem entendida como um algoritmo, visto abaixo:

1. Aproxime as fronteiras dos obstáculos poligonais com um grande número de pontos que resultam da subdivisão de cada lado do polígono original em pequenos segmentos.
2. Calcule o diagrama Voronoi para esta coleção de pontos aproximados.

3. O diagrama Voronoi é construído, eliminam-se as bordas Voronoi as quais tem um ou ambos os pontos finais recaem em de qualquer dos obstáculos.
4. A borda Voronoi restante forma uma boa aproximação do diagrama Voronoi generalizado para obstáculos originais no mapa.
5. Uma vez o diagrama Voronoi esteja construído, as configurações iniciais e objetivo são retraídos ao diagrama Voronoi e o caminho resultante é uma concatenação de trajetória direto do ponto inicial para o diagrama, o caminho ao longo do diagrama e a linha reta do diagrama ao objetivo.

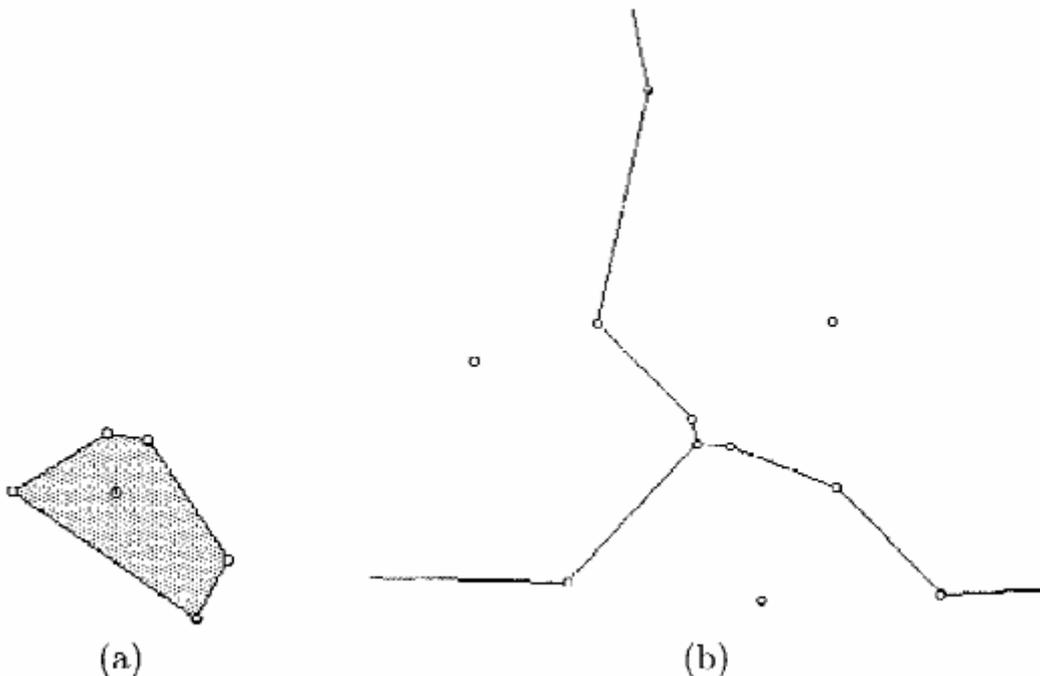


FIG. 3.7. Diagrama de Voronoi dos pontos próximos. (a) Polígono subjacente e (b) diagrama dos três pontos.

Algoritmos computacionais eficientes baseados nesta abordagem são calculados em $O(n^2)$ ou até mesmo $O(n \log n)$. O diagrama Voronoi precisa ser calculado somente uma vez em relação ao q_{ini} e q_{obj} . O mais utilizado diagrama Voronoi baseado em algoritmo de planejamento é chamado retração. Retração em topologia é definida como um mapa “subjetivo” de um espaço topológico de seu subconjunto Y se for contínuo e sua restrição para Y é mapa de identidade. Dado uma configuração inicial e objetivo existe um caminho livre se houver um caminho em R entre q_{ini} e q_{obj} onde: $C_{livre} \rightarrow R$ é a retração. Isto leva o tempo de $O(n)$. A vantagem da abordagem baseada no diagrama Voronoi é que ela alcança os

caminhos livres que tendem a maximizar a clareza entre o robô e os obstáculos. Para um estudo aprofundado vide [Barequet, 2000].

3.3.1.3.2. ÁRVORES ALEATÓRIAS DE EXPLORAÇÃO RÁPIDA

A estrutura de dados usada pelo método baseado em árvore aleatória de exploração rápida (AAER) e os correspondentes algoritmos são projetados para a busca eficiente em espaços de dimensões elevadas não convexos. As AAER são construídas incrementalmente de uma maneira que rapidamente reduz a distância esperada de um ponto aleatoriamente escolhido pela árvore. As AAER são particularmente adequadas para problemas de planejamento de trajetória que envolvem obstáculos e limitações diferenciais (não holonomicidade), isto é, quando a dimensão é elevada. As AAER podem ser consideradas como uma técnica para gerar trajetórias em malha aberta para sistemas não lineares com limitação de estados. Uma AAER pode ser intuitivamente considerada com um modo Monte-Carlo de busca tendenciosa em grandes regiões Voronoi. Algumas variações podem ser consideradas como fractais estocásticos. Geralmente, uma AAER sozinha é insuficiente para resolver um problema de planejamento. Portanto, pode ser considerado como uma componente que pode ser incorporada no desenvolvimento de uma variedade de diferentes algoritmos de planejamento. Uma das melhores abordagens de AAER é o planejamento cinemático dinâmico e aleatório sugerido por Lavalle. O primeiro passo é escolher pontos aleatoriamente no espaço de configuração e rejeitá-lo se ele não estiver no espaço de configuração C_{livre} . Então selecionamos os pontos marcados e os conectamos se a distância entre dois pontos seja menor que D e a linha reta conectando-os recaem em C_{livre} . Este algoritmo é bem apropriado a aplicações em tempo real porém sem garantia de ser completo ou otimizado [Cheng, 2001; Kuffner, 2000].

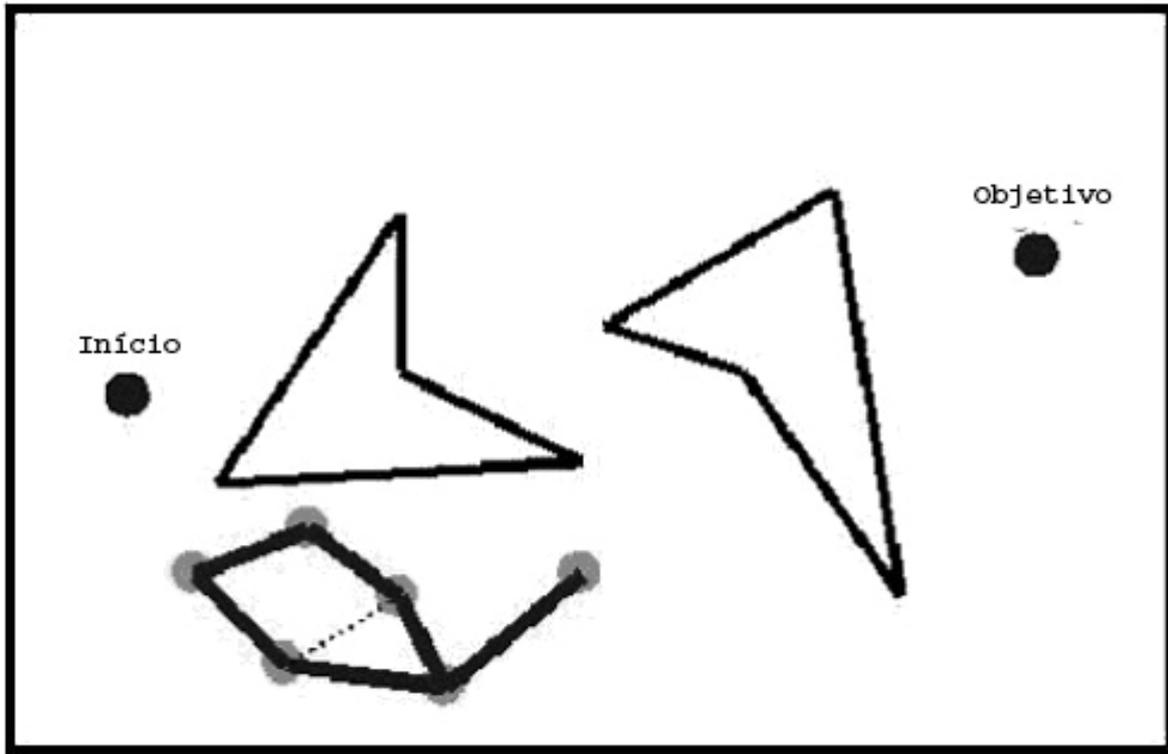


FIG. 3.8. Exemplo de utilização de árvores aleatórias de busca.

3.3.1.3.3. ÁRVORES DE NAVEGAÇÃO POR SALTOS

Embora este método não está estritamente incluído na categoria de métodos Roadmap como descrito anteriormente, visto que o roadmap calculado pelo robô não é um roadmap geometricamente verdadeiro no ambiente. As árvores de navegação por saltos essencialmente representam a área visível de todos os pontos por um grafo representando as profundas discontinuidades em torno do robô, portanto compartilhando algumas características de um grafo de visibilidade. O robô precisa explorar o ambiente com um sensor omni-direcional tal como um anel de sonares ou lasers. O robô retém informações do ambiente na forma de árvore, na qual cada nó representa uma discontinuidade brusca ou canto, a árvore cresce quando uma discontinuidade aparece, por exemplo, se o robô move-se ao longo de um corredor entre dois cômodos, quando se move em direção a um dos cômodos o outro cômodo vem a ser representada pela abertura da porta, o robô sabe que o cômodo está atrás da porta. O robô está primeiro no modo de exploração, varrendo ao longo a procura de bordas do grafo e adicionando novas bordas quando necessário. Para localizar o objetivo, nós somente

colorimos o apropriado nó da árvore e uma vez o ambiente esteja completamente explorado a navegação é feita percorrendo a árvore por um nó colorido os quais são vistos anteriormente pelo robô. A navegação do robô é somente percorrida pelas linhas bi-tangentes as quais são codificadas na estrutura da árvore de navegação por saltos. A abordagem não exige

uma mudança súbita nas limitações aplicadas ao movimento de um robô. As células por si só são regiões de várias formas as quais exigem despesas adicionais. Uma vez o espaço seja dividido em células o grafo de adjacência resultante dá todas as regiões que necessitaria atravessar para ligar uma configuração inicial a uma final. Um exemplo de algoritmo é conhecido como algoritmo de varredura de linha (para espaços de obstáculos poligonais convexos) é apresentado aqui.

Selecionam-se vértices do espaço de obstáculos na coordenada x .

2. “Varre-se” com uma linha vertical L da esquerda para a direita parando em cada vértice V no espaço de obstáculo.

3. Três casos são agora possíveis em cada parada ou evento:(a) Uma borda do vértice está à esquerda e uma borda do vértice está à direita de L . Neste caso, o final da célula atual e o início da nova célula. (b) Ambas as bordas do vértice estão à direita da linha de varredura. Neste caso, o final de ambas as células atuais e o início de uma nova célula. (c) Ambas as bordas do vértice estão à esquerda da linha de varredura. Neste caso, o final da célula atual e o início de 3 novas células.

O algoritmo executa em tempo $O(n \log n)$ e é completo, portanto sempre encontra um trajeto se existir. O grafo de conectividade resultante é usado então como uma linha guia para o movimento do robô no ambiente. Tais algoritmos foram explorados com profundidade na geometria computacional e suas aplicações para planejamento de trajetória apresentadas em [Sleumer, 1999].

3.3.1.3.4.2. MÉTODOS APROXIMADOS

Os métodos aproximados criam um conjunto de células que cobrem aproximadamente C_{livre} . Estas células têm limites regulares e assim é mais fácil calcular transversais, mas muitos detalhes sobre o ambiente são perdidos. Especificamente, detalhes que são menores do que a metade da menor dimensão da célula não seria capturada no todo.

Um exemplo para aproximar decomposição em célula é o algoritmo Quad-Tree [Goodrich e Rimon, já citados anteriormente]. Este algoritmo representa o espaço de configuração como uma estrutura quadtree pela execução das seguintes operações:

1. Colocar o espaço de configuração como o nó raiz e a célula atual.
2. Dividir a célula atual em n células não sobrepostas.

3. Verificar cada célula para ver se há obstáculos, observando se os obstáculos não interseccionam acima de alguma ou de nenhuma das células.

4. Represente cada uma destas células sobrepostas como nós filhos.

5. Se uma célula estiver completamente livre, ou preenchida completamente por obstáculos ela não tem filhos, quando as características da célula inteira são conhecidas.

6. Se uma célula é mista (parcialmente livre, parcialmente preenchida por obstáculos) este processo retorna a etapa 2.

7. Uma vez que uma boa definição suficiente é encontrada, um trajeto pode ser calculado através das parcelas conhecidas do C_{livre} baseado no grafo de conectividade obtido.

Um planejamento de trajetória que usa o algoritmo Quad-tree é sugerido por Arimoto et al.[Naniwa, 1990]. As estruturas de dados conhecidas como grades de ocupação e seus algoritmos associados também pertencem à categoria dos métodos aproximados. O método da grade de ocupação cobre o espaço com uma grade regular e então essencialmente colore as células de preto se houver um obstáculo na região. As ligações entre as células são as células adjacentes que são desobstruídas. A eficiência computacional dos algoritmos que usam este método depende do refinamento da decomposição. A decomposição mais refinada, mais trajetos são criados, e mais próximo do trajeto mais curto na rede. Entretanto, a complexidade temporal para procurar tal rede aumenta exponencialmente. Também, a menos que a decomposição seja executada algebricamente, a conformidade dos confinamentos tais como trajetórias não suaves, limites de velocidade e de aceleração, e confinamentos dinâmicos são ignorados. Neste caso, o trajeto gerado necessita ser modificado em um estágio posterior para levar em conta estes confinamentos.

3.3.2. TÓPICOS EM INTELIGÊNCIA COMPUTACIONAL

Nesta seção são apresentadas os conceitos envolvidos em inteligência computacional utilizados neste trabalho.

3.3.2.1. INTRODUÇÃO À LÓGICA FUZZY

Nesta seção serão apresentados conceitos básicos sobre a lógica *Fuzzy*, sua metodologia e formas de aplicação mais usuais.

De forma simples, pode-se caracterizar a lógica *Fuzzy* como "um tipo de lógica que reconhece mais do que simples valores de verdadeiro ou falso". Com a lógica fuzzy, proposições podem ser representadas com graus de verdade e falsidade. Por exemplo, a frase, hoje está ensolarado, pode ser 100% verdadeira se não há nuvens, 80% verdadeira se há poucas nuvens, 50% verdadeira se está nublado e 0% verdadeira se chove durante todo o dia."

A lógica *Fuzzy* provê um processamento lingüístico, ou seja, método de traduzir expressões verbais vagas, imprecisas e qualitativas em valores numéricos, permitindo que computadores sejam utilizados como sistemas inteligentes baseados na experiência humana [Shaw, 1999].

3.3.2.1.1. ASPECTOS QUALITATIVOS DA LÓGICA FUZZY

Vista de forma qualitativa, a lógica *fuzzy* pode ser apresentada da seguinte forma:

- Diferentemente da lógica aristotélica (bivalente - verdadeiro ou falso), a lógica *fuzzy* é multivalente, ou seja, a verdade é graduada, recebendo um valor no intervalo $[0,1]$, sendo 0 representando completamente falso e 1 completamente verdadeiro;
- Expressões verbais, imprecisas, qualitativas, inerentemente humanas, podem ser tratadas através do formalismo da lógica *fuzzy*;
- As implicações lógicas, ou inferência lógica, as entradas e saídas, ou antecedentes e conseqüentes, são associados a graus de verdade no intervalo $[0,1]$;
- A utilização da lógica *fuzzy* facilita a interface computador-homem, por permitir ao primeiro uma melhor compreensão da linguagem inerentemente imprecisa dos seus operadores.

3.3.2.1.2. CONJUNTOS FUZZY

Seja um conjunto universo E e x um elemento desse conjunto. O conjunto A é o conjunto de pares ordenados tais que:

$$\{[x, \mu_A(x)]\}, \forall x \in E$$

onde $\mu_A(x)$ é o grau de pertinência de x em A . O valor de $\mu_A(x)$, na lógica *fuzzy*, pode ser qualquer valor no intervalo $[0,1]$.

3.3.2.1.3. OPERAÇÕES COM CONJUNTOS

Assim como nos conjuntos convencionais, os conjuntos *fuzzy* possuem operadores de união, intersecção e complemento. A seguir serão definidas as principais operações utilizadas.

- União

A função de pertinência $\mu_{A \cup B}(u)$, $u \in U$, da união $A \cup B$, é definida como:

$$\mu_{A \cup B}(u) = \mu_A(u) \mathbf{s} \mu_B(u)$$

onde \mathbf{s} é a co-norma triangular de união generalizada. Diversas normas \mathbf{s} podem ser utilizadas, dentre as quais:

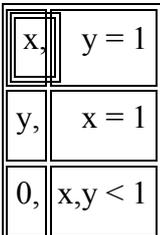
União	$x \mathbf{s} y = \max(x, y)$
Soma algébrica	$x \mathbf{s} y = x + y - xy$

- Intersecção

A função de pertinência $\mu_{A \cap B}(u)$, $u \in U$, da intersecção $A \cap B$, é definida como:

$$\mu_{A \cap B}(u) = \mu_A(u) \mathbf{t} \mu_B(u)$$

onde \mathbf{t} é a norma triangular de intersecção generalizada. Diversas normas \mathbf{t} podem ser utilizadas, dentre as quais:

Intersecção	$x \text{ t } y = \min(x,y)$
Produto algébrico	$x \text{ t } y = xy$
Produto drástico	$x \text{ t } y = \{$ 

- Complemento

A função de pertinência do complemento é dada como segue:

$$\mu_{A'}(x) = 1 - \mu_A(x)$$

3.3.2.1.4. FUNÇÕES DE PERTINÊNCIA

Uma função de pertinência *fuzzy* é uma função numérica gráfica ou tabulada que atribui valores de pertinência para valores discretos de uma variável, em de seu universo de discurso (o universo de discurso representa o intervalo numérico de todos os valores possíveis que uma variável lingüística pode assumir).

As funções de pertinência assumem diversas formas, sendo as mais utilizadas a triangular e a trapezoidal. A figura 3.9 apresenta as principais funções de pertinência geralmente utilizadas em projetos de controladores *fuzzy*.

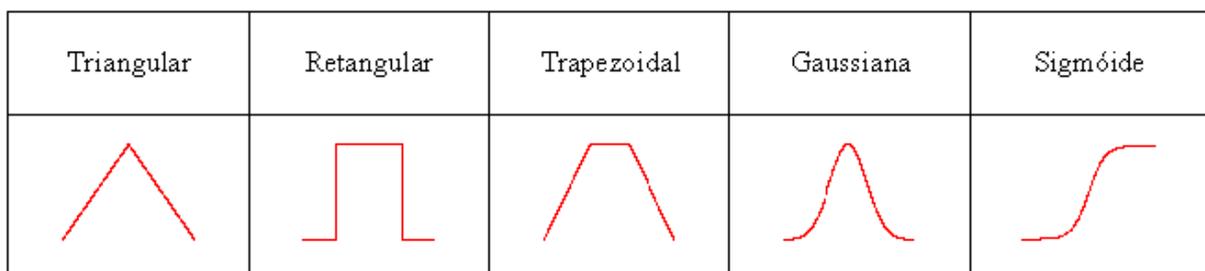


FIG. 3.9. Funções de pertinência *fuzzy*.

A quantidade de funções de pertinência a ser utilizada em um universo de discurso e seus formatos são geralmente baseados na experiência. Apesar disso, algumas dicas práticas podem ser utilizadas:

- o número de funções de pertinência utilizado varia geralmente entre 2 e 7. Quanto maior o seu número, maior a precisão mas também é maior o custo computacional;
- o formato das funções de pertinência é geralmente triangular ou trapezoidal, mas caso seja necessário um comportamento mais suave do sistema, devem ser utilizadas funções como a gaussiana ou a sigmóide;
- a precisão do sistema é dependente do grau de superposição entre as funções de pertinência. Geralmente aconselha-se para testes iniciais 50% de superposição, e para o sistema otimizado um mínimo de 25% e um máximo de 75% de superposição em sistemas de malha fechada;
- caso haja dados experimentais, é possível utilizar sistemas *neuro-fuzzy* para gerar automaticamente as funções de pertinência;
- as funções de pertinência devem preferivelmente abranger todo o universo de discurso da variável analisada.

Fuzzyficação

A *fuzzyficação* é o mapeamento do domínio de valores numéricos (VN) reais (como valores obtidos de um sensor) para valores *fuzzy* (VF), definidos pelas funções de pertinência. Como pode ser visto na figura 3.10, o valor *fuzzy* para a variável X é dado por 0.3 para a função M e por 0.7 para L.

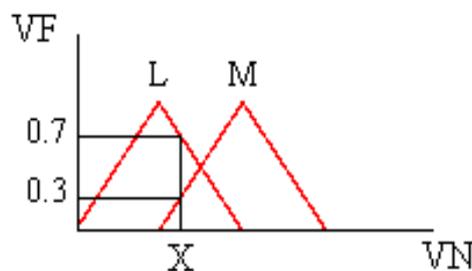


FIG. 3.10: *Fuzzyficação* de valores numéricos.

Regras

As regras *fuzzy* seguem dois tipos de implicações: *modus ponens* (modo afirmativo) e o *modus tollens* (modo negativo). A tabela 3.1 mostra a forma como essas implicações são utilizadas.

TAB.3.1: Métodos de implicações *fuzzy*.

Modo	<i>Modus ponens</i>	<i>Modus tollens</i>
Premissa 1	$u=A$	$y=\text{não-B}$
Premissa 2	se $u=A$ então $y=B$	se $u=A$ então $y=B$
Conseqüência	$y=B$	$y=\text{não-A}$

Uma regra *fuzzy* utilizando a implicação *modus ponens* toma a seguinte forma:

SE $X=A$ ENTÃO $Y=B$

Generalizando, as regras *fuzzy*, utilizando as variáveis e suas funções de pertinência, possuem a seguinte sintaxe:

SE < variavel_1 = fp_1 > E/OU <variavel_2 = fp_2> E/OU ...
 ENTÃO < variavel_S = fp_S >

onde *variavel_n* refere-se a uma variável de entrada, *fp_n* a uma função de pertinência desta variável n, e *variavel_S* e *fp_S* referem-se à variável de saída e uma de suas funções de pertinência.

Inferência

A inferência *fuzzy* baseia-se na transformação do espaço de entradas $A(x)$ para o espaço de saídas $B(y)$ através de uma relação matricial $R(x,y)$. Matematicamente:

$$B(y) = A(x) \circ R(x,y)$$

onde \circ é o operador composicional de inferência. Os operadores mais utilizados são os seguintes:

$$\text{Max-min} \quad \mu_B(y) = \max_{x \in A} \{ \min[\mu_A(x), \mu_R(x,y)] \}$$

$$\text{Max-produto} \quad \mu_B(y) = \max_{x \in A} \{ \mu_A(x) \cdot \mu_R(x,y) \}$$

Defuzzyficação

Na *defuzzyficação* o valor das variáveis lingüísticas de saída é traduzido para um valor discreto numérico, com o objetivo de obter-se o melhor valor que represente o valor desejado. Existem diversos métodos de *defuzzyficação*, cada um apropriado para uma aplicação determinada:

- Centro da Área ou Método do Centróide

No método do centróide, o valor numérico retornado é calculado a partir da seguinte expressão:

$$u^* = \frac{\sum_{i=1}^N \mu(x_i) x_i}{\sum_{i=1}^N \mu(x_i)}$$

- Centro do Máximo ou *Defuzzyficação* pelas Alturas

O método dos máximos realiza uma média ponderada dos máximos das funções de pertinência, segundo a seguinte fórmula:

$$u^* = \frac{\sum_{i=1}^N u_i \cdot \sum_{k=1}^n \mu_{O,k}(u_i)}{\sum_{i=1}^N \sum_{k=1}^n \mu_{O,k}(u_i)}$$

- Média dos Máximos

Esta abordagem utiliza o valor médio entre os máximos das funções de pertinência, segundo a seguinte expressão:

$$u^* = \sum_{m=1}^M \frac{u_m}{M}$$

onde u_m é o m -ésimo elemento no universo de discurso onde $\mu_{\text{sai}}(u_i)$ possui um máximo.

3.3.2.1.5. SISTEMAS DE CONTROLE FUZZY E NEURO-FUZZY

Considere o sistema dinâmico

$$\dot{x} = f(x, u), \quad y = g(x, u) \quad (3.34)$$

onde $x \in \mathbb{R}^n$ é um vetor de variáveis de estado e $u \in \mathbb{R}^r$, $y \in \mathbb{R}^m$ são os vetores de entrada e saídas do sistema, respectivamente, $f(*) \in \mathbb{R}^n$ e $g(*) \in \mathbb{R}^m$ são mapeamentos suaves, $f: \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^n$ e $g: \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^m$. Se $f(*)$ e $g(*)$ representam mapeamentos lineares então (3.34) recai na categoria de sistema linear, por exemplo,

$$\dot{x} = Ax + Bu, \quad y = Cx + Du \quad (3.35)$$

Definindo-se o erro de varredura, ou seja, a diferença entre x_d e x (se a realimentação de estados for utilizada) ou entre y_d e y (se a realimentação de saída for usada), na qual x_d e y_d são as trajetórias desejadas. Portanto a equação diferencial (3.34) pode formar um sistema de erro diferencial. Logo, o problema de controle pode ser equivalentemente expresso do ponto de vista matemático como: “Encontrar um mapeamento apropriado do domínio de erro para o domínio de entrada tal que a solução nula do sistema de erro diferencial é estável.”

O controlador fuzzy nada mais é que um mapeamento não linear do domínio de erro para o domínio de estados. Todas as informações de erro são amostradas no domínio E como valores “singleton” ou “crisp” (valor numérico), depois mapeadas (codificadas) em alguns conjuntos fuzzy por meio das funções de pertinência $\mu(e)$. Então os valores dos conjuntos fuzzy no domínio E são mapeados em valores de conjuntos fuzzy no domínio U através da associação de regras fuzzy R. Finalmente, os valores dos conjuntos fuzzy no domínio U serão mapeadas (decodificadas) para valores “crisp” de sinais de controle pela técnica de defuzzificação como mostra a figura 3.11. Para um completo mapeamento do domínio E para o domínio U ser contínuo, as funções membro fuzzy definidas no domínio E devem ser conjuntos fechados ou semi-fechados e contínuos, ou seja, qualquer elemento no domínio E deve pertencer a pelo menos um subconjunto fuzzy. O mesmo vale para os sinais de controle no domínio U, contudo a condição de continuidade do domínio U não é necessária se as regras forem apropriadamente configuradas [Mohammad, 1997].

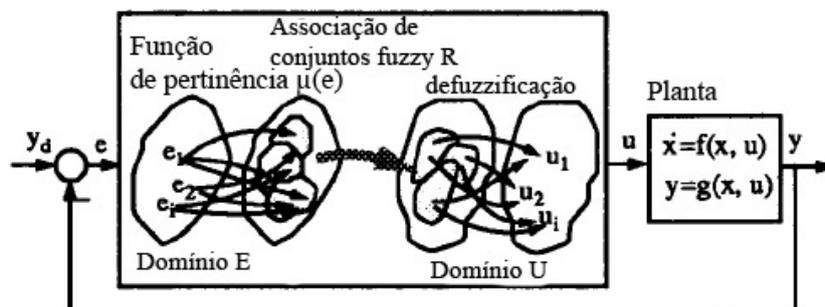


FIG. 3.11. Representação do controlador fuzzy.

Embora o controle convencional, baseado em controladores PID e funções de transferência, sejam adequados para a maior parte das situações, as implementações por lógica *fuzzy* freqüentemente são mais eficientes, devido às seguintes características:

- o controle *fuzzy* nasce da experiência ao invés de modelos matemáticos, portanto uma implementação lingüística é muito mais fácil e rápida de ser definida;
- condições raras ou excepcionais no controle podem ser incorporadas com pouco custo computacional, permanecendo o sistema ainda transparente e compreensível;
- a implementação da lógica *fuzzy* é freqüentemente mais eficiente em termos de codificação e tempo computacional de execução.

Um controlador *fuzzy* é geralmente composto dos seguintes blocos funcionais:

- interface de *fuzzyficação*;
- base de conhecimento (regras);
- sistema de inferência;
- interface de *defuzzyficação*.

3.3.2.1.6. APLICAÇÕES DA LÓGICA FUZZY

A lógica *fuzzy* é aplicada em diversas áreas, seja comercial ou de pesquisa. Entre algumas aplicações comerciais podem ser citadas:

- Câmeras de Vídeo

O controle de foco e abertura do diafragma de câmeras pode ser obtido através da lógica *fuzzy*, garantindo assim uma melhor qualidade das imagens obtidas e permitindo equipamentos mais simples e baratos.

- Máquinas de lavar roupa

O controle dos ciclos da máquina é obtido através do sensoriamento de diversas características para a lavagem, como a temperatura da água, peso das roupas e grau de sujeira. Aplicando-se o conhecimento empírico humano a um controlador *fuzzy* é possível obter a lavagem das roupas garantindo-se melhor utilização do equipamento.

- Sistemas de Ar-Condicionado

A utilização de um controlador *fuzzy* para sistemas de ar-condicionado permite economia energética de 10 a 30%, devido ao bom ajuste do sistema às diversas variáveis de entrada

(sensores de temperatura e umidade) e aos atuadores (válvulas de água fria, abertura das tubulações, pressão das bombas, entre outros).

No âmbito acadêmico, destacam-se como aplicações da lógica *fuzzy* o controle de motores, análise de sinais, diagnósticos médicos e o controle de robôs móveis (velocidade e trajetória, principalmente) [De Luca, já citado anteriormente].

3.4. ERROS DE ODOMETRIA SISTEMÁTICOS E NÃO-SISTEMÁTICOS

A odometria é baseada em equações simples que são facilmente implementadas e que utilizam dados de codificadores de roda incrementais baratos. Entretanto, a odometria é também baseada na suposição que as voltas da roda podem ser traduzidas em deslocamento linear relativo ao assoalho. Esta suposição é somente de validade limitada. Um exemplo extremo é o deslizamento da roda: se uma das rodas estiver deslizando, por exemplo, em um derramamento de óleo, então o codificador associado registraria voltas da roda mesmo que estas voltas não correspondessem a um deslocamento linear da roda.

Com o caso extremo de deslizamento total, há algumas outras razões mais sutis para inexatidão na tradução das leituras do codificador da roda em movimento linear. Todas estas fontes do erro são classificadas em uma das duas categorias: *erros sistemáticos* e *não-sistemáticos*.

Erros Sistemáticos:

- Ø Diâmetros desiguais da roda.
- Ø A média dos diâmetros reais da roda difere do seu diâmetro nominal.
- Ø A base da roda real difere da sua base nominal.
- Ø Desalinhamento das rodas.
- Ø Definição finita do codificador.
- Ø Taxa de amostragem do codificador finita.

Erros não-sistemáticos:

- Ø Trajeto sobre assoalhos desnivelados.

- Ø Trajeto sobre objetos inesperados no assoalho.
- Ø Deslizamento da roda devido a:
 - o assoalhos lisos.
 - o aceleração brusca.
 - o giro rápido (derrapagem).
 - o forças externas (interação com corpos externos).
 - o forças internas (transmissão das rodas).
 - o Nenhum ponto de contato da roda com o assoalho.

A distinção clara entre erros sistemáticos e não-sistemáticos é de grande importância para a efetiva redução dos erros de odometria. Por exemplo, os erros sistemáticos são particularmente graves porque se acumulam constantemente. Em muitas superfícies internas polidas erros sistemáticos contribuem muito mais para erros de odometria do que erros não-sistemáticos. Entretanto, em superfícies ásperas com irregularidades significativas, os erros não-sistemáticos são dominantes. O problema com erros não-sistemáticos é que podem aparecer inesperadamente (por exemplo, quando o robô atravessa um objeto inesperado no assoalho), e podem causar grandes erros de posição. Tipicamente, quando um sistema de robô móvel utiliza sistema de navegação híbrido odometria/marcos, a frequência dos marcos é determinado empiricamente e é baseada no pior caso dos erros sistemáticos. Tais sistemas provavelmente falham quando um ou maiores erros não-sistemático ocorrem.

É notório que muitos pesquisadores desenvolvem algoritmos que estimam a incerteza de posição de um robô inoperante (e.g., [Tonouchi, 1994;Komoriya, 1994].) Com esta abordagem cada posição do robô calculada é cercada por uma “elipse de erro” característica, a qual indica uma região de incerteza para a posição real do robô (veja figura 3.12) [Tonouchi, 1994;Adams, 1994]. Tipicamente, estas elipses crescem com distância do trajeto, até uma medida de posição absoluta reduz o crescimento da incerteza e desse modo "reconfigura" o tamanho da elipse de erro. Estas técnicas de estimação de erro devem contar com parâmetros de estimação de erro derivados das observações do desempenho de veículo. Claramente, estes parâmetros podem levar em conta somente erros sistemáticos, porque a magnitude dos erros não-sistemáticos é imprevisível.

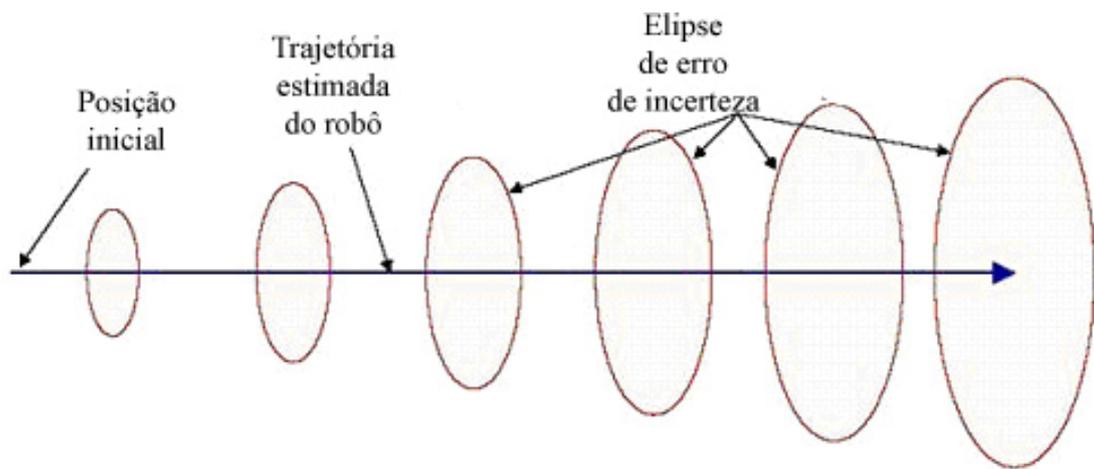


FIG. 3.12. Crescimento das “elipses de erro” indica o crescimento das incertezas de posição com odometria.

4 DESENVOLVIMENTO DO SISTEMA

O robô no qual se baseia este estudo é um típico sistema mecânico não holonômico. Na ausência de obstáculos, ele pode atingir qualquer posição no plano, com qualquer orientação. Conseqüentemente, o espaço de configuração é tridimensional. Assumindo nenhum deslizamento, a velocidade do ponto central entre as duas rodas traseiras do robô é sempre tangente à orientação dele.

O espaço de velocidades alcançáveis em qualquer configuração é, portanto, bidimensional. O planejamento de trajetória livre de colisão consiste em construir um caminho conectando duas configurações de entrada no subconjunto de configurações onde o robô não tenha contato ou intersecção com obstáculos. Confinamentos não holonômicos exigem que a tangente ao caminho em qualquer configuração esteja contido no subespaço de velocidades selecionadas inteiramente pelos confinamentos. Um caminho livre de colisão para um robô não holonômico, tipicamente, inclui "manobras", isto é, pontos de apoio no quais o robô pára e altera a velocidade.

Encontrar um caminho plausível entre duas configurações é um problema difícil. Outro é minimizar o número de manobras, ou pelo menos mantê-las em um número razoável, quando possível. Primeiramente, far-se-á uma análise matemática dos confinamentos não holonômicos. Utilizando resultados da geometria diferencial e da teoria de controle não linear, dar-se-á uma caracterização formal da holonomicidade em confinamentos lineares diferenciais e relacionaremos os estados resultantes com sua controlabilidade. Em particular, estabelecem-se dois resultados efetivos aplicáveis quando o robô está sujeito a um simples confinamento linear diferencial. O primeiro resultado permite-nos determinar através de um simples cálculo se este confinamento é holonômico ou não. O segundo resultado estabelece que qualquer robô sujeito a um simples confinamento linear não holonômico é completamente controlável, isto é, qualquer duas configurações recaem em um subconjunto aberto conectado de espaços de configuração que podem ser conectados por um caminho pertencente a este subconjunto considerando o confinamento não holonômico.

4.1. DESCRIÇÃO DO PROBLEMA

O protótipo é uma estrutura cilíndrica em alumínio de 36 cm de altura e 30 cm de diâmetro, dois motores de corrente contínua nos quais são acoplados em seu eixo caixas de redução e uma roda livre de suporte (figuras 4.1 e 4.2). Todo controle é baseado nos dados oriundos do conjunto de 3 sensores ultrasônicos dispostos na parte frontal do mesmo e defasados de 45 graus. Tais sensores têm lóbulo de abertura de 60 graus totalizando 180 graus de abertura frontal, o circuito destes sensores permite via software a variação do alcance máximo de cada módulo, portanto é definido nesta implementação o alcance de 6 m, no planejamento global – campo potencial, e 1 m, no planejamento local – controlador fuzzy. Essa característica otimiza o processamento dos dados de distância obtidos e reduz a sobreposição de lóbulos. Na parte superior, montou-se um suporte para a barra com um dispositivo de travamento transversal da barra, que mantém a barra travada no robô líder até o momento de encontrar um obstáculo côncavo, e um sensoriamento ótico deste deslocamento, como visto na figura 4.3. Nas rodas também é utilizado sensores óticos de rotação dos motores com o intuito de possíveis correções de trajetória e ajustes do modelo real dos robôs.

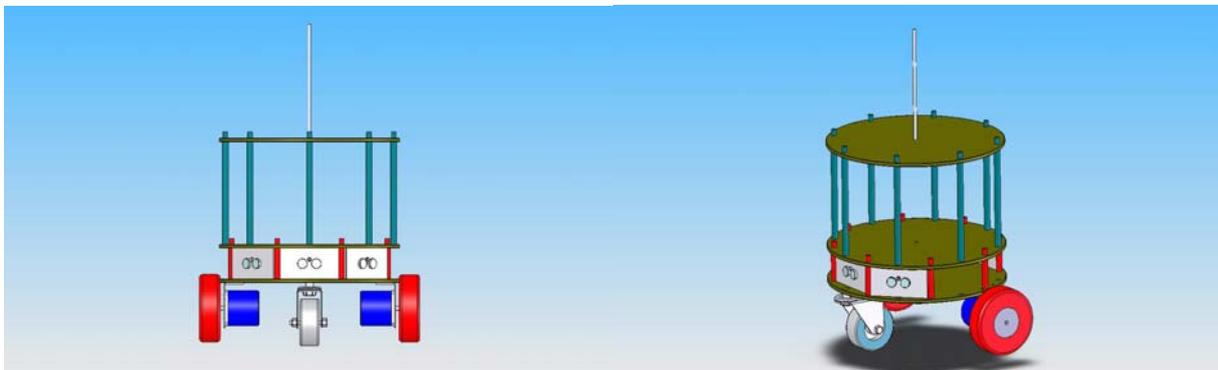


FIG. 4.1. Modelo em 3 dimensões dos robôs utilizados na simulação.



FIG. 4.2. Protótipo

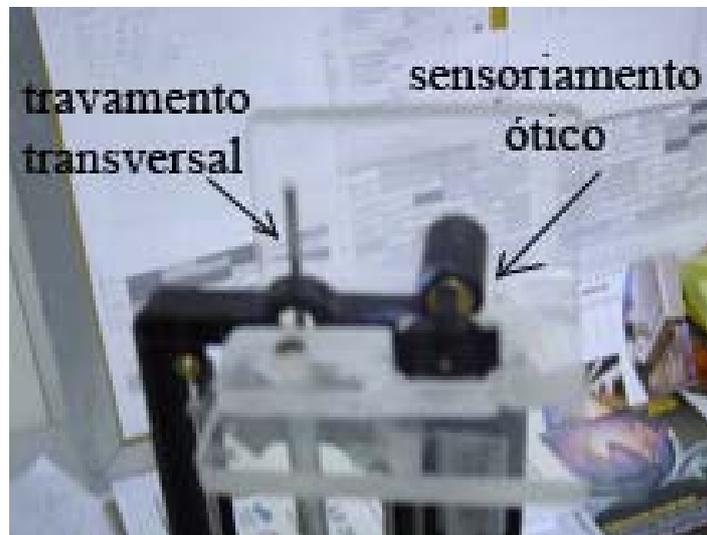


FIG. 4.3. Suporte para o transporte da barra.

Como dito inicialmente, o planejamento de trajetória é dividido em global e local, sendo este último subdividido em dois módulos, monitoramento da possibilidade de colisão e desvio de obstáculos, ambos implementados em fuzzy. A possibilidade de colisão tem a função de alterar o controle do CPA para o módulo de desvio de obstáculos, quando tal possibilidade ultrapassar um limiar, figura 4.4.



FIG. 4.4. Módulo de possibilidade de colisão.

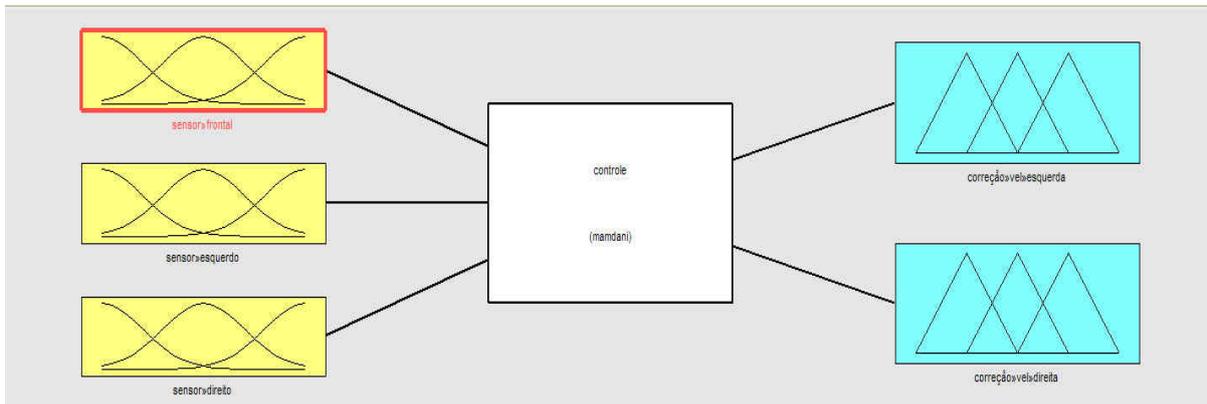


FIG. 4.5. Controlador Fuzzy.

O Sistema de controle baseia-se em uma estratégia global (campo potencial) e local (lógica fuzzy) de acordo com o fluxograma abaixo:

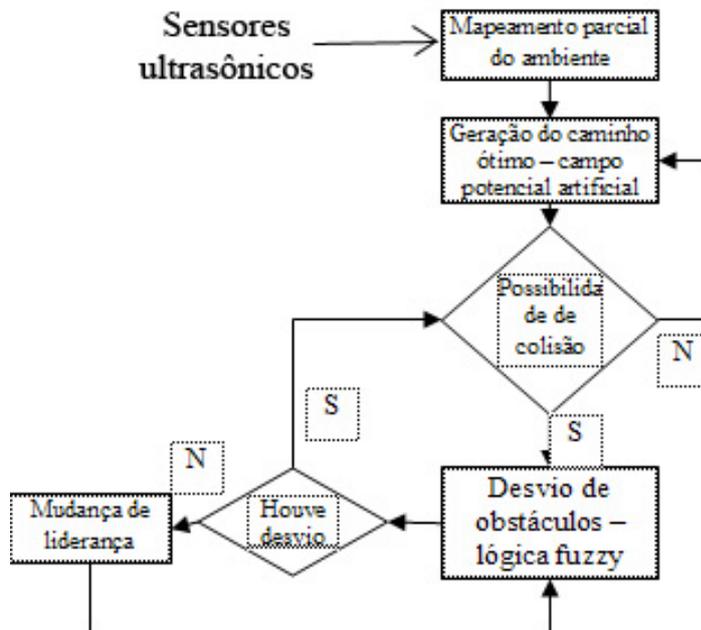


FIG. 4.6. Fluxograma do algoritmo de controle

Na arquitetura líder-seguidor é prevista a mudança de liderança no caso do líder encontrar algum obstáculo côncavo, no qual o controlador fuzzy pára o líder, o sensor de deslocamento da barra no robô seguidor percebe esta parada e trava a barra, após deslocar-se para retornar a posição original da barra, figura 4.7. Feito isso, trava-se a barra no robô seguidor e destrava-se no robô líder, ocorre a passagem de liderança, o seguidor retorna por alguns instantes e o líder o acompanha, ao finalizar ocorre o processo de passagem de liderança e, o robô líder, através do controlador fuzzy toma nova decisão.

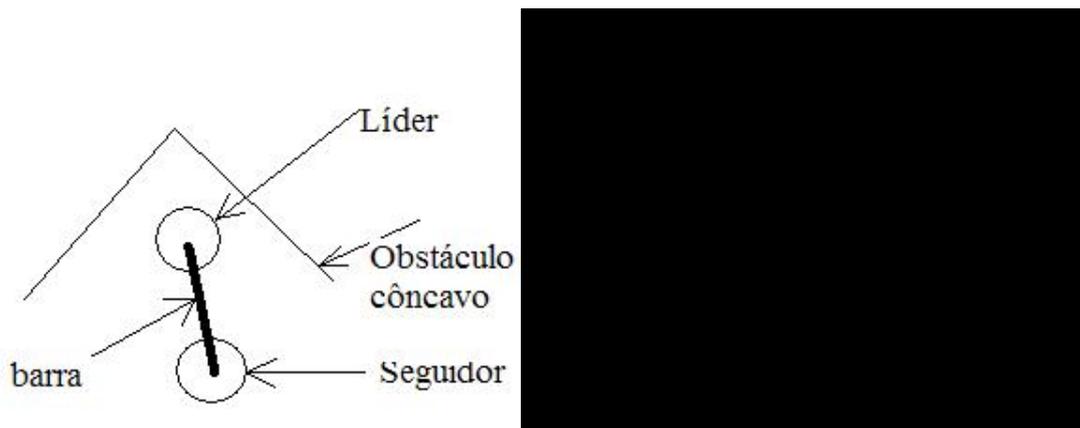


FIG. 4.7. Condição para mudança de liderança.

Na próxima seção é feita uma análise matemática dos robôs usados neste trabalho, abordando seus aspectos cinemáticos e dinâmicos.

4.2. MODELO CINEMÁTICO E EQUAÇÕES DE CONFINAMENTO

Este robô é dirigido por duas rodas, $2b$ é o comprimento dos eixos entre as rodas do robô móvel e r é o raio das rodas. $\{W\}$ é o sistema de coordenadas inerciais (ou sistema de coordenadas globais) e $\{R\}$ é o sistema de coordenadas fixos para o robô móvel. P_0 é a origem de $\{R\}$ e é colocado no meio do eixo das rodas. θ_d e θ_e representam os ângulos de rotação das duas rodas, direita e esquerda, respectivamente, (em relação aos estados iniciais arbitrários). A pose do robô é dado pela posição (x,y) de seu centro e o ângulo de direção \emptyset . P_c é o centro de massa do robô com coordenadas (x_c, y_c) , e é colocado no eixo X a uma distância d de P_0 e finalmente a é o comprimento do robô na direção perpendicular do eixo

das rodas, figura 4.8. O equilíbrio do robô é mantido por um pequeno castor cujo efeito será ignorado. Portanto, $q = (x_c, y_c, \phi, \theta_d, \theta_e)$ representam a configuração do sistema, ou seja, as cinco coordenadas generalizadas.

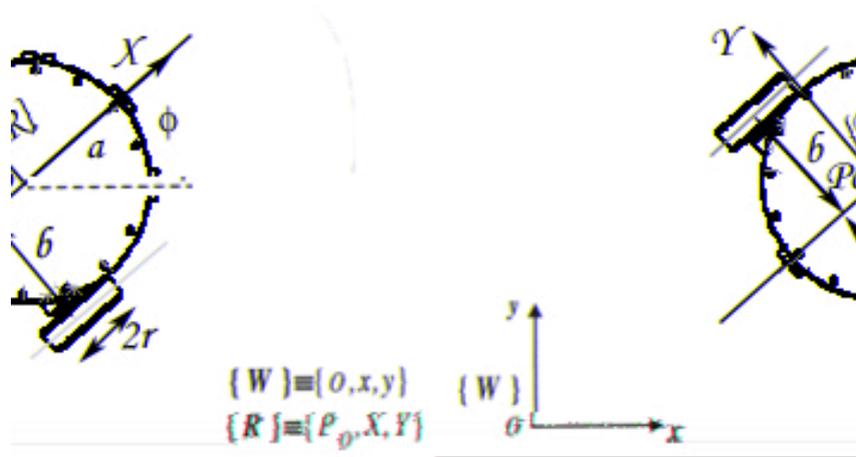


FIG. 4.8. Representação do robô móvel

No modelo cinemático supõe-se que em cada contato existe um movimento de rolamento puro, isto é, cada roda pode rolar na direção na qual aponta e girar sobre o seu eixo vertical, mas não pode deslizar. Assumindo que a velocidade de P_0 deve estar na direção do eixo de simetria (eixo X) e as rodas não devem deslizar, são obtidos, em relação a P_c , os seguintes conjuntos de confinamentos:

$$\begin{aligned} \dot{y}_c \cos \phi - \dot{x}_c \operatorname{sen} \phi - \dot{\phi} d &= 0 \\ \dot{x}_c \cos \phi + \dot{y}_c \operatorname{sen} \phi + b \dot{\phi} - r \dot{\theta}_r &= 0 \\ \dot{x}_c \cos \phi + \dot{y}_c \operatorname{sen} \phi - b \dot{\phi} - r \dot{\theta}_l &= 0 \end{aligned} \quad (4.1)$$

Estes confinamentos podem ser reescritos na forma:

$$A(q) \dot{q} = 0 \quad (4.2)$$

com

$$A(q) = \begin{bmatrix} -\operatorname{sen} \phi & \cos \phi & -d & 0 & 0 \\ \cos \phi & \operatorname{sen} \phi & b & r & 0 \\ \cos \phi & \operatorname{sen} \phi & b & 0 & r \end{bmatrix}$$

$$S(q) = [s_1(q), s_2(q)] = \begin{bmatrix} c(b \cos \phi - d \sin \phi) & c(b \cos \phi + d \sin \phi) \\ c(b \sin \phi + d \cos \phi) & c(b \sin \phi - d \cos \phi) \\ c & -c \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (4.4)$$

que satisfaz a equação $A(q)S(q) = 0$, e onde a constante $c = \frac{r}{2b}$. Então o modelo cinemático é dado por

$$\dot{q} = S(q)v(t), \quad \text{com } v = \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} \quad (4.5)$$

Neste estágio, o teste de não holonomicidade é completado com a proposta de determinar os tipos de confinamentos cinemáticos do robô. Usando conceito de filtragem, segue que $\Delta_1 = \text{span}\{s_1(q), s_2(q)\}$, onde $\dim \Delta_1 = \text{posto}[s_1(q), s_2(q)] = 2$.

Portanto, calculando a Lie Algebra de $s_1(q)$ e $s_2(q)$, tem-se:

$$s_3(q) = [s_1(q), s_2(q)] = \frac{\partial s_2}{\partial q} s_1 - \frac{\partial s_1}{\partial q} s_2 = \dots = \begin{bmatrix} -rc \sin \phi \\ rc \cos \phi \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.6)$$

Desde que $s_3(q)$ não é uma combinação linear de $s_1(q)$ e $s_2(q)$, então $s_3(q)$ não é uma distribuição Δ_1 “spanned” por $s_1(q)$ e $s_2(q)$, e então ao menos um dos confinamentos é não holonômico. Então a distribuição Δ_2 assume a forma $\Delta_2 = \text{span}\{s_1(q), s_2(q), s_3(q)\}$ com $\dim \Delta_2 = \text{posto}[s_1(q), s_2(q), s_3(q)] = 3$. Usando o mesmo processo de cálculo apresentado para $s_3(q)$, os “Lie brackets” são calculados para obter $s_4(q)$ e $s_5(q)$:

$$s_4(q) = [s_1(q), s_3(q)] = \begin{bmatrix} -rc^2 \cos \phi \\ -rc^2 \sin \phi \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.7)$$

e $s_5(q) = [s_2(q), s_3(q)] = -s_4(q)$, como $s_5(q)$ é múltiplo de $s_4(q)$ então Δ_3 é unicamente $\Delta_3 = \text{span}\{s_1(q), s_2(q), s_3(q), s_4(q)\}$ e, $\dim \Delta_3 = \text{posto}[s_1(q), s_2(q), s_3(q), s_4(q)] = 4$.

Da mesma maneira, as próximas combinações “Lie brackets” são:

$$\begin{aligned}
s_6(q) &= [s_1(q), s_4(q)] = [rc^3 \text{sen}\phi \quad -rc^3 \text{cos}\phi \quad 0 \quad 0 \quad 0]^T = -c^2 s_3(q) \in \Delta_3; \\
s_7(q) &= [s_2(q), s_4(q)] = c^2 s_3(q) \in \Delta_3; \\
s_8(q) &= [s_3(q), s_4(q)] = 0 \in \Delta_3.
\end{aligned} \tag{4.8}$$

Como $s_6(q)$ e $s_7(q)$ são múltiplos de $s_3(q)$, $s_8(q)$ é uma combinação linear de $s_3(q)$ e $s_4(q)$ e todos estão na distribuição Δ_3 , logo $\Delta_4 = \text{span}\{s_1(q), s_2(q), s_3(q), s_4(q)\} = \Delta_3$ e $\dim\Delta_4 = 4$.

Como $\dim\Delta_4 = \dim\Delta_3$, logo Δ_3 é involutiva e $\Delta_{3+j} = \Delta_3$ para todo $j \geq 0$. Deste resultado, pode-se concluir que a distribuição “spanned” por $s_1(q)$, $s_2(q)$, $s_3(q)$ e $s_4(q)$ é involutiva sendo todos os vetores linearmente independentes e, conseqüentemente, $\Delta^* = \text{span}\{s_1(q), s_2(q), s_3(q), s_4(q)\}$.

O grau de não holonomicidade de Δ^* é $k = 3$, é o sistema é parcialmente não holonômico. Da definição 3, pode-se concluir que entre os três confinamentos, dois deles são não holonômicos.

Subtraindo a terceira da segunda equações de 4.1, o confinamento holonômico é obtido,

$$\dot{\phi} = \frac{r}{2b}(\dot{\theta}_r - \dot{\theta}_l) \tag{4.9}$$

Concluindo, temos um confinamento holonômico (4.9), e dois confinamentos não holonômicos:

$$\begin{aligned}
\dot{y}_c \cos\phi - \dot{x}_c \text{sen}\phi - \dot{\phi}d &= 0 \\
\dot{x}_c \cos\phi + \dot{y}_c \text{sen}\phi - \frac{r}{2}(\dot{\theta}_r + \dot{\theta}_l) &= 0
\end{aligned} \tag{4.10}$$

4.3. MODELO DINÂMICO DOS ROBÔS PROPOSTOS

Considere um robô móvel não holonômico com n coordenadas generalizadas q sujeitas a m confinamentos (assumindo que $m < n$) cujas equações dinâmicas do movimento são descritas por:

$$M(q)\ddot{q} + V(q, \dot{q}) = B(q)\tau - A^T(q)\lambda \tag{4.11}$$

onde $V(q, \dot{q}) = C(q, \dot{q})\dot{q}$, $M(q) \in R^{n \times n}$ é a matriz de inércia, $C(q, \dot{q}) \in R^{n \times n}$ é a matriz de forças centrípeta e de Coriolis, $B(q) \in R^{n \times (n-m)}$ é a matriz de transformação da entrada, $A^T(q)$ é a matriz Jacobiana, $\tau \in R^{(n-m)}$ é o vetor de entrada e $\lambda \in R^m$ é o vetor de forças dos confinamentos. As m equações de confinamento do sistema mecânico pode ser escrito na forma:

$$w(q, \dot{q}) = 0 \quad (4.12)$$

Se uma equação de confinamento está na forma $W_i(q) = 0$, é dita holonômica, caso contrário, trata-se de um confinamento cinemático dito não holonômico.

Para alcançar as equações dinâmicas do robô móvel é necessário calcular o Lagrangiano tal como para obter as equações de Lagrange do movimento e, conseqüentemente, as matrizes M , V e B de forma a tornar a representação espaço-estado mais fácil.

As equações Lagrangianas de movimento para o robô em estudo podem ser escritas como:

$$\begin{aligned} m\ddot{x}_c &= md(\dot{\phi} \sin \phi - \phi^2 \cos \phi) - \lambda_1 \sin \phi - \cos \phi (\lambda_2 - \lambda_3) = 0 \\ m\ddot{y}_c &= md(\dot{\phi} \cos \phi + \phi^2 \sin \phi) - \lambda_1 \cos \phi - \sin \phi (\lambda_2 - \lambda_3) = 0 \\ I \ddot{\phi} &= mdy_c \cos \phi - mdx_c \sin \phi - d_1 \lambda_1 + b(\lambda_3 - \lambda_2) = 0 \\ I_w \ddot{\phi} &= 2r \ddot{\phi} \\ I_w \ddot{\phi} &= 3r \ddot{\phi} \end{aligned}$$

$$M(q) = \begin{bmatrix} m & 0 & md \operatorname{sen} \phi & 0 & 0 \\ 0 & m & -md \operatorname{cos} \phi & 0 & 0 \\ md \operatorname{sen} \phi & -md \operatorname{cos} \phi & I & 0 & 0 \\ 0 & 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & 0 & I_w \end{bmatrix} \quad (4.14)$$

$$V(q, \dot{q}) = \begin{bmatrix} md \dot{\phi}^2 \operatorname{cos} \phi \\ md \dot{\phi}^2 \operatorname{sen} \phi \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad B(q) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \tau = \begin{bmatrix} \tau_d \\ \tau_e \end{bmatrix}$$

e, finalmente, utilizando a variável de estado $x = [x_c \ y_c \ \phi \ \theta_d \ \theta_e \ \dot{\theta}_d \ \dot{\theta}_e]^T$, a dinâmica do robô móvel pode ser representada na forma espaço-estado:

$$\dot{x} = \begin{bmatrix} S_v \\ f_2 \end{bmatrix} + \begin{bmatrix} 0 \\ (S^T M S)^{-1} S^T B \end{bmatrix} \tau \quad (4.15)$$

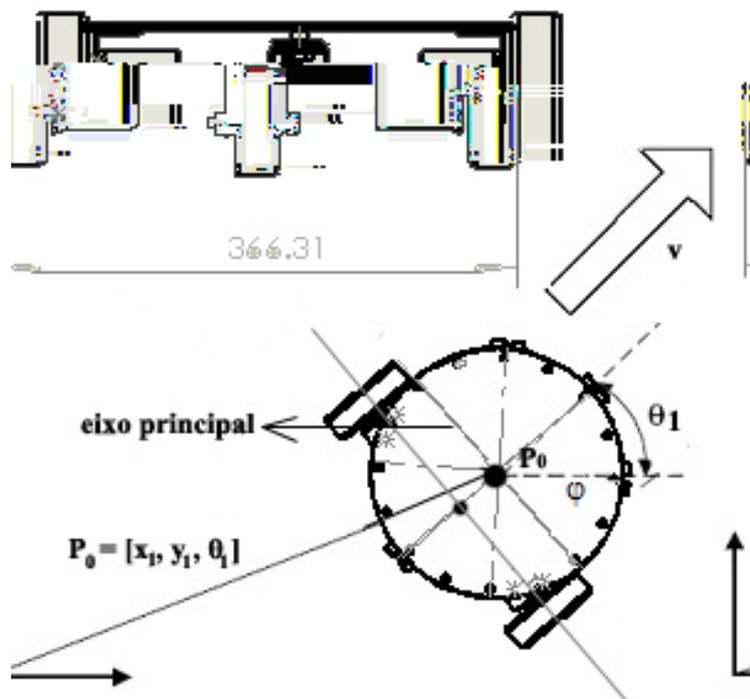


FIG. 4.9. Análise cinemática

Considere uma estrutura cilíndrica com 3 rodas, sendo uma que fornece o apoio e as outras duas a tração(robô móvel de estrutura única) e a direção (diferencial de velocidades)

mostrada na figura 4.9. Seu espaço de configuração é $D \times S^1$, na qual D é um domínio compacto de \mathbb{R}^2 . Parametriza-se cada configuração pelas coordenadas X_1 e Y_1 do centro geométrico do cilindro e o ângulo θ_1 entre o eixo x do plano Cartesiano fixado no plano do chão e o eixo que passa pelo ponto central entre as duas rodas e o seu centro geométrico. Os parâmetros de velocidade são \dot{X}_1 , \dot{Y}_1 e $\dot{\theta}_1$. Os parâmetros de controle do robô são a velocidade v pertencente a \mathbb{R} do centro geométrico P_0 (se $v > 0$, o robô move-se para frente) e o ângulo de direção ϕ medindo a orientação da velocidade de P_0 em relação ao eixo principal do robô (se $\pi/2 < \phi < \pi$ e $v > 0$, o robô está virando para esquerda). Assumir-se-á que os contatos entre as rodas e o chão será puramente de contato por rolamento entre corpos rígidos. A velocidade dos pontos P_1 ao longo do eixo principal são representados pelo seguinte confinamento cinemático:

$$-\dot{X}_1 \text{sen} \theta_1 + \dot{Y}_1 \text{cos} \theta_1 = 0 \quad (4.16)$$

O sistema de controle equivalente é calculado como:

$$\left. \begin{aligned} \dot{X}_1 &= v \text{cos} \phi \text{cos} \theta_1 \\ \dot{Y}_1 &= v \text{cos} \phi \text{sen} \theta_1 \\ L_1 \dot{\theta}_1 &= v \text{sen} \phi \end{aligned} \right\} \quad (4.17)$$

Como v toma valores tanto positivos quanto negativos, o sistema é simétrico. Neste caso, fraca controlabilidade é equivalente a controlabilidade, e os resultados para confinamentos não holonômicos é aplicável.

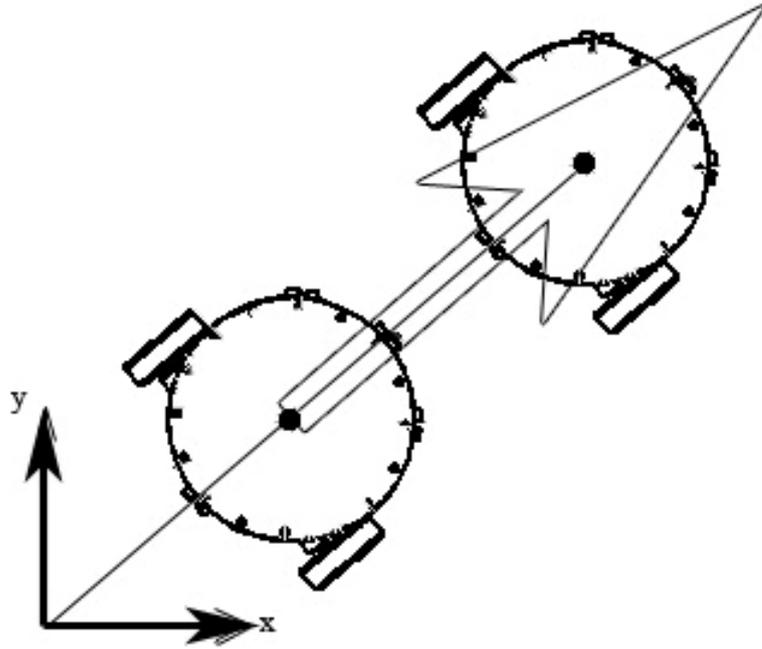


FIG. 4.10. Estrutura líder-seguidor.

Em um sistema de robôs móveis de dois corpos (sistema cooperativo proposto) figura 4.10, há dois confinamentos cinemáticos. A velocidade do centro geométrico de ambos os robôs é tangente à orientação do conjunto. Mais genericamente, um pode ser considerado um robô móvel p rebocando $(p - 1)$ trailers seqüencialmente atrelados. O centro geométrico do primeiro corpo (o trator) é indicado por P_1 . O centro geométrico do k -ésimo corpo é designado P_k . Consequentemente, tem-se pontos P_1, \dots, P_p , cujas coordenadas são expressas por $(X_1, Y_1), \dots, (X_p, Y_p)$. A orientação do k -ésimo corpo em relação ao eixo x do plano Cartesiano no plano é expresso por θ_k . O espaço de configuração do robô de corpo p é $D \times (S^1)^p$, onde D é um domínio compacto de \mathbb{R}^2 . Sua dimensão é $n = p + 2$. Parametrizaremos cada configuração por $(X_1, Y_1, \theta_1, \dots, \theta_p)$. Os parâmetros de velocidade são $\dot{X}_1, \dot{Y}_1, \dot{\theta}_1, \dots, \dot{\theta}_p$. Os parâmetros de controle são os mesmos do robô, isto é, a velocidade v e o ângulo de direção ϕ .

Há p confinamentos cinemáticos de igualdade, um para cada corpo. Para determiná-los, representaremos os pontos P_1, \dots, P_p no plano complexo, isto é, $P_k = X_k + iY_k$. L_k representa o comprimento do k -ésimo corpo, escrever-se-á os confinamentos geométricos entre os $k-1$ e k corpos como $P_k = P_{k-1} - L_k \exp(i\theta_k)$, levando :

$$P_k = P_1 - \sum_{l=2}^k L_l \exp(i\theta_l) \quad (4.18)$$

O confinamento cinemático do k-ésimo corpo é:

$$\dot{P}_k = \lambda_k \exp(i\theta_k) \quad (4.19)$$

o qual equivale a:

$$I(\exp(-i\theta_k) \dot{P}_k) = 0 \quad (4.20)$$

onde $I(z)$ expressa a parte imaginária do número complexo z . Combinando esta caracterização com a equação da derivada e utilizando a linearidade do operador I , obtém-se o k-ésimo confinamento cinemático:

$$-\dot{X}_1 \text{sen} \theta_k + \dot{Y}_1 \cos \theta_k - \sum_{l=2}^k L_l \dot{\theta}_l \cos(\theta_l - \theta_k) = 0 \quad (4.21)$$

Em particular, para $k = 1$ e $k = 2$, obtém-se:

$$\begin{cases} -\dot{X}_1 \text{sen} \theta_1 + \dot{Y}_1 \cos \theta_1 = 0 \\ -\dot{X}_1 \text{sen} \theta_2 + \dot{Y}_1 \cos \theta_2 - L_2 \dot{\theta}_2 = 0 \end{cases} \quad (4.22)$$

As equações acima são os confinamentos cinemáticos aplicados a conjunto dos dois robôs.

Similarmente, pela combinação de $\dot{P}_k = \lambda_k \exp(i\theta_k)$ com a derivada de

$$X(v, \phi) = \cos(\phi)X(v, 0) + \text{sen}(\phi)X\left(v, \frac{\pi}{2}\right) \quad (4.26)$$

Vamos selecionar qualquer dois campos correspondente a dois diferentes valores ϕ_1 e ϕ_2 do ângulo de direção. A Lie Algebra gerada por $X(v, \phi_1)$ e $X(v, \phi_2)$ são as mesmas que as geradas por $X(v, 0)$ e $X\left(v, \frac{\pi}{2}\right)$, por causa da bilinearidade da operação de Lie Brackets:

$$\left[X(v, \phi_1), X(v, \phi_2)\right] = \text{sen}(\phi_1 - \phi_2) \left[X(v, 0), X\left(v, \frac{\pi}{2}\right)\right] \quad (4.27)$$

Portanto, a dimensão da Lie Algebra do sistema robótico multi-corpo não é afetado pelos confinamentos desiguais no ângulo de direção. Foi mostrado em [Murray, 1993] que esta dimensão é máxima para robôs móveis de um e dois corpos. Abaixo estendemos este resultado a um robô de três corpos.

O espaço do campo vetorial de controle constante é gerado pelos dois seguintes campos vetoriais:

$$\begin{aligned} X_1 &= X(1, 0) = \left(\cos \theta_1, \text{sen} \theta_1, 0, \frac{\text{sen}(\theta_1 - \theta_2)}{L_2}, \cos(\theta_1 - \theta_2) \frac{\text{sen}(\theta_2 - \theta_3)}{L_3} \right) \\ X_2 &= L_1 X\left(1, \frac{\pi}{2}\right) = (0, 0, 1, 0, 0) \end{aligned} \quad (4.28)$$

cujo Lie Bracket é:

$$X_3 = [X_1, X_2] = (-\text{sen} \theta_1, \cos \theta_1, 0, \cos(\theta_1 - \theta_2)/L_2, -\text{sen}(\theta_1 - \theta_2) \text{sen}(\theta_2 - \theta_3)/L_3) \quad (4.29)$$

Calcula-se depois:

$$\begin{aligned} X_4 &= L_2 [X_1, X_3] = (0, 0, 0, 1/L_2, -\cos(\theta_2 - \theta_3)/L_3) \\ X_5 &= \cos(\theta_1 - \theta_2) [X_1, X_4] - \text{sen}(\theta_1 - \theta_2) [X_3, X_4] = \\ &= (0, 0, 0, 1/L_2^2, -1/L_3^2 - \cos(\theta_2 - \theta_3)/(L_2 L_3)) \end{aligned} \quad (4.30)$$

Finalmente:

$$\det(X_1, X_2, X_3, X_4, X_5) = -\frac{1}{L_2 L_3^2} \neq 0 \quad (4.31)$$

Utilizando uma prova de recorrência, Laumond [Lamiriaux, 2001] recentemente mostrou que este resultado estende-se a robôs móveis multi-corpos com um número arbitrário de robôs, isto é, que a Lie Algebra de um robô móvel de corpo p tem dimensão p + 2. Portanto:

“Um robô móvel multi-corpo é sempre controlável que o ângulo de direção do trator possa ter ao menos dois diferentes valores.”

A fim de ter um controle da pose do robô será utilizado um codificador óptico o qual fornece a rotação de cada roda sendo assim, integrada ao sistema de controle com a finalidade de corrigir o deslocamento do robô. Os parâmetros utilizados para o cálculo da distância percorrida são c_m , o fator de conversão dos pulsos do codificador para deslocamento linear, D_n o diâmetro nominal da roda em milímetros, C_e a resolução do codificador e n o fator de redução da caixa de redução utilizada. A distância percorrida é calculada levando-se em conta o movimento relativo das rodas esquerda e direita, ΔU_{Ei} e ΔU_{Di} , respectivamente, então:

$$\Delta U_{E/Di} = c_m N_{E/Di} \quad (4.32)$$

onde N representa o incremento de pulso com uma taxa de amostragem I . O deslocamento linear ΔU_i do centro do robô é dado por:

$$\Delta U_i = (\Delta U_D - \Delta U_E) / 2 \quad (4.33)$$

e a orientação é calculada com:

$$\Delta \theta_i = (\Delta U_D - \Delta U_E) / b \quad (4.34)$$

onde b é o diâmetro da base do robô e idealmente é a distância entre os dois pontos de contato das rodas com o chão.

A nova orientação do robô θ_i pode ser calculada de:

$$\theta_i = \theta_{i-1} + \Delta \theta_i \quad (4.35)$$

e a nova posição relativa do centro do robô é obtida por:

$$\begin{cases} x_i = x_{i-1} + \Delta U_i \cos \theta_i \\ y_i = y_{i-1} + \Delta U_i \sin \theta_i \end{cases} \quad (4.36)$$

onde x_i e y_i são as coordenadas relativas, que representam o centro C do robô no instante i .

4.4. CONTROLADOR FUZZY

No módulo de desvio de obstáculos do controlador fuzzy, figura 4.11, são considerados as possíveis situações encontradas em ambientes fechados tais como, escritórios ou laboratórios. Na tabela 4.2 é apresentado o conjunto de regras fuzzy deste módulo. Os conjuntos de entrada: muito perto (MP), perto (P) e longe (L) foram escolhidos,

primeiramente, visando a redução das regras e, conseqüentemente, o custo computacional, além disso, os conjuntos utilizam funções suaves com a finalidade de evitar grandes variações entre os conjuntos, implicando em um melhor mapeamento dos conjuntos de saída. Acrescentando-se a estas características, como visto na figura 4.12, os conjuntos de entrada estão mais concentrados em distâncias próximas ao robô líder, posto que, com a redução do alcance para 1 (um) metro reduz-se a sobreposição dos lóbulos de sensores ultrassônicos diminuindo assim, os erros de leitura dos mesmos. Os conjuntos de saída, também, apresentam funções suaves a fim de evitar erros de odometria devido ao deslizamento das rodas.

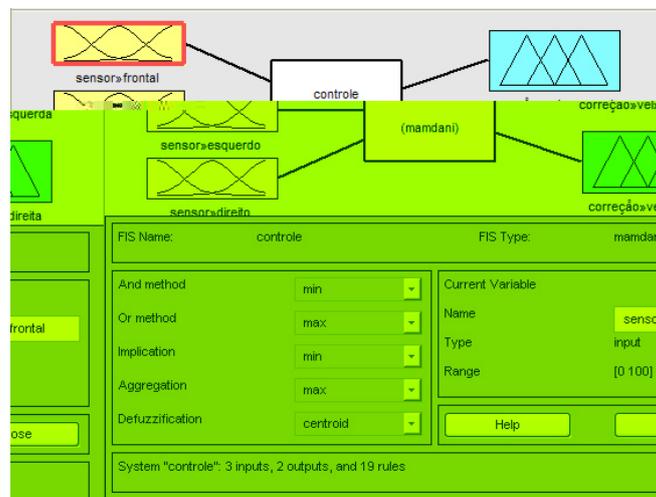
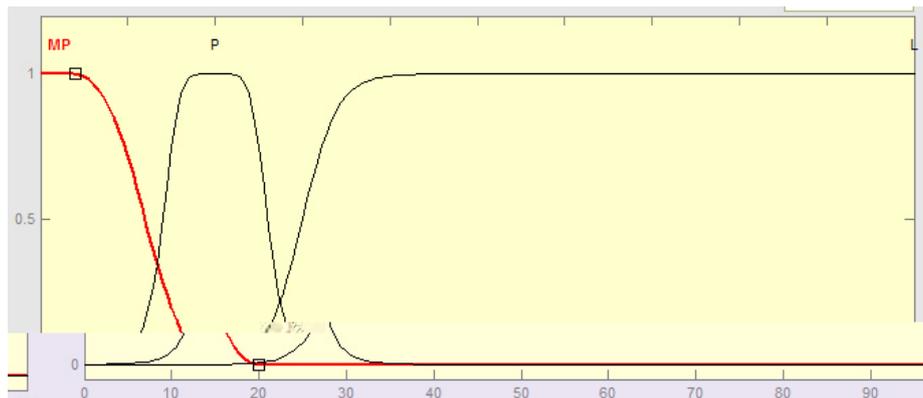
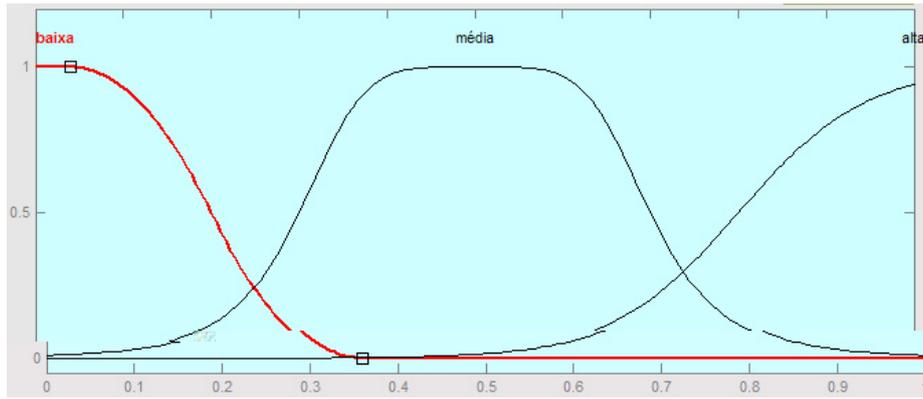


FIG. 4.11. Variáveis de entrada e saída do controlador fuzzy.



(a)



(b)

FIG. 4.12. Conjuntos fuzzy (a) Sensores, (b) Velocidades.

TAB. 4.1. Regras do controlador fuzzy.

SE			ENTÃO	
Sensor direito	Sensor frontal	Sensor esquerdo	Velocidade esquerda	Velocidade direita
longe	longe	longe	alta	Alta
perto	longe	longe	baixa	Média
muito perto	longe	longe	baixa	Alta
longe	longe	perto	média	Baixa
longe	longe	muito perto	alta	Baixa
longe	perto	longe	média	Baixa
longe	muito perto	longe	alta	Baixa
longe	perto	perto	média	Baixa
perto	perto	perto	média	Baixa
muito perto	perto	perto	alta	Baixa
perto	longe	perto	média	Média

perto	muito perto	perto	baixa	Média
perto	perto	longe	baixa	Média
perto	perto	muito perto	alta	Baixa
muito perto	longe	muito perto	parar	Parar
longe	muito perto	muito perto	alta	Baixa
perto	muito perto	muito perto	alta	Baixa
muito perto	muito perto	muito perto	parar	Parar
muito perto	perto	muito perto	parar	Parar
muito perto	muito perto	longe	baixa	Alta
muito perto	muito perto	perto	baixa	Alta

Neste módulo são criadas regras prevendo a possibilidade do robô líder encontrar obstáculos côncavos implementando-se a arquitetura líder-seguidor. O outro módulo de monitoramento da possibilidade de colisão em cada um dos sensores, figura 4.13, é acionado ao iniciar-se o movimento do conjunto robótico e, caso haja possibilidade de colisão despreza o controle por CPA e cria-se uma nova trajetória através do módulo de desvio de obstáculos. Na próxima seção é descrito as características do algoritmo de CPA adotado neste trabalho.

Possibilidade de Colisão

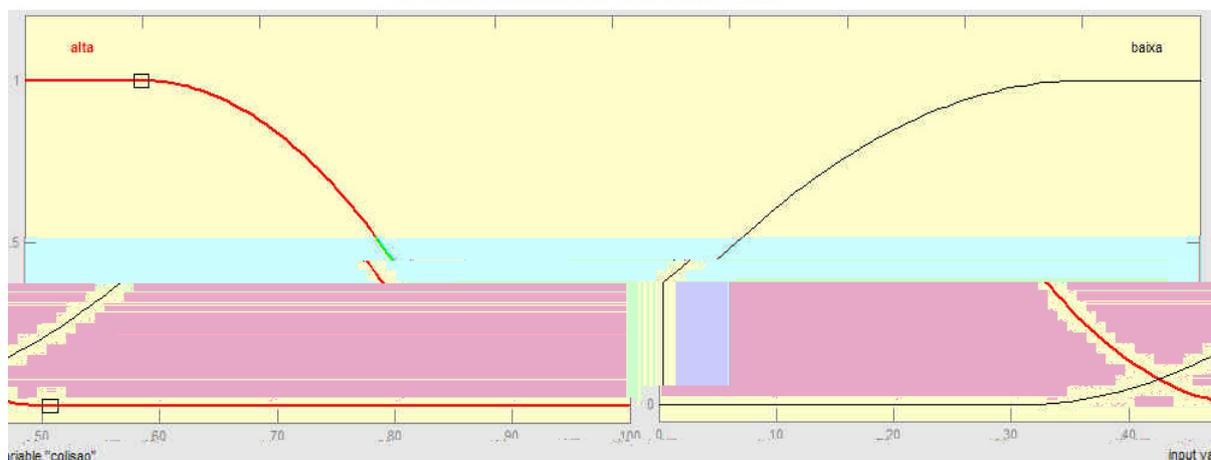


FIG. 4.13. Conjunto fuzzy de possibilidade de colisão.

4.5. CAMPO POTENCIAL ARTIFICIAL

Como descrito nas seções anteriores, o CPA é calculado no início do movimento do conjunto robótico (líder + seguidor), após alterar-se o alcance dos sensores para 6 (seis) metros e, depois do desvio de algum obstáculo não detectado no primeiro mapeamento pelo CPA. Utilizou-se a função de Campo Potencial tradicional, figura 4.14, visto que os possíveis mínimos locais são resolvidos pelo controlador fuzzy. A única modificação feita no algoritmo é a associação de pesos ao potencial de cada obstáculo proporcional a sua dimensão. Na seção seguinte, será descrito a estrutura eletrônica adotada para a realização deste trabalho.

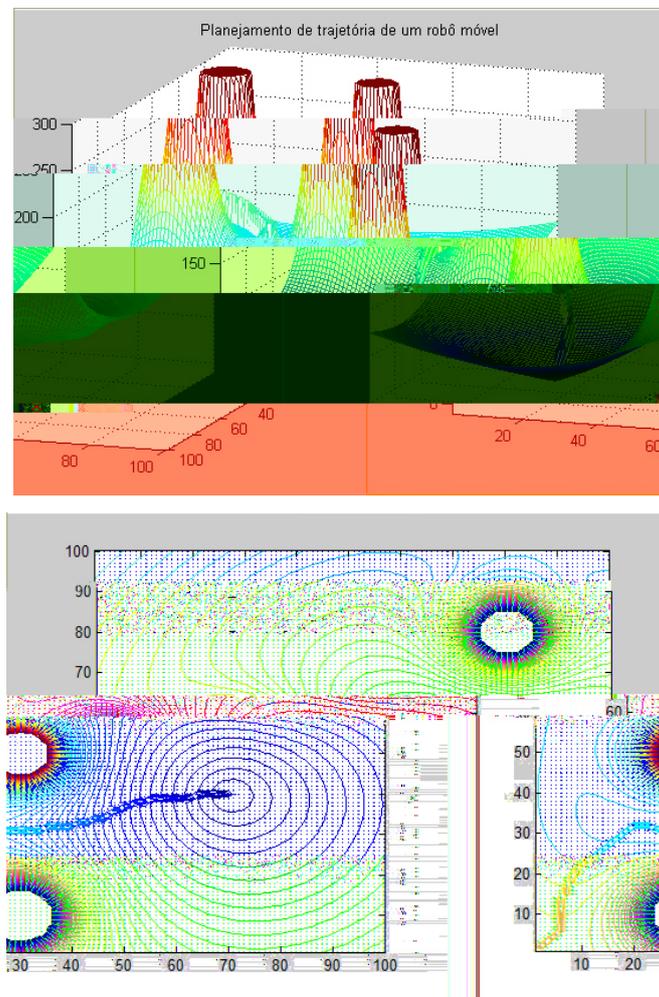


FIG. 4.14. Representações em duas e três dimensões do campo potencial artificial calculado.

4.6. ELETRÔNICA EMBARCADA

Na montagem do protótipo composto de dois robôs cilíndricos foi implementado o sistema de controle para o planejamento de trajetória em um ambiente semi-estruturado, além do monitoramento da barra, através do uso de um microcontrolador da Microchip modelo PIC16F877, o qual monitora os 3 (três) pares de sensores ultrasônicos, os sensores óticos e resistivos da barra. O circuito implementado no robô líder concentra todas as funções, ao passo que, no robô seguidor, somente o monitoramento da barra e controle de velocidade e posição no acompanhamento do líder. Para isso, o circuito foi dividido nos seguintes módulos, figura 4.15:

1. Transmissão do pulso ultrasônico;
2. Recepção deste pulso;
3. Controle de velocidade – PWM;
4. Comunicação – serial e I²C;
5. Sensoriamento e travamento da barra; e
6. Odometria.

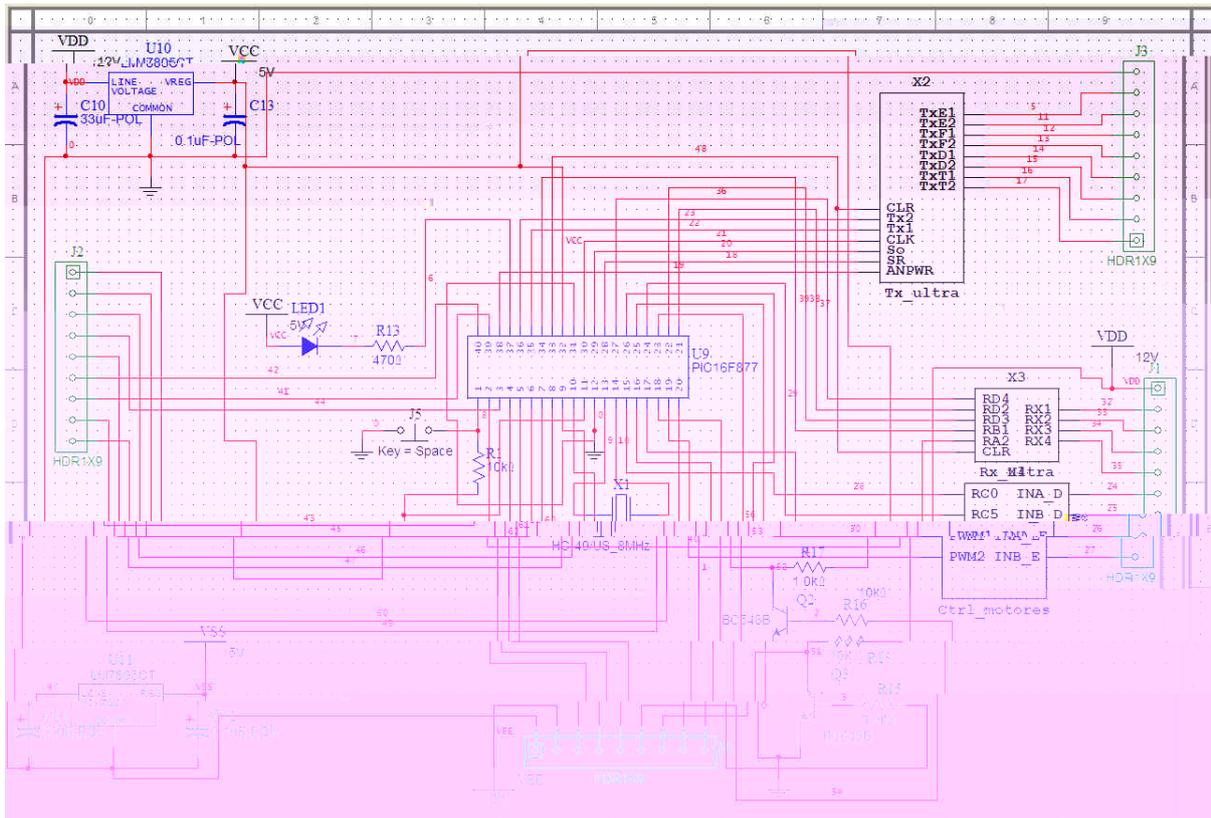


FIG. 4.15. Esquemático do circuito utilizado no líder.

No robô líder, o módulo de transmissão e recepção ultrasônica funciona em anel, da seguinte forma: faz-se a leitura do sensor esquerdo, frontal e direito nesta ordem, armazenando os valores em um buffer, figura 4.16. Após processamento do algoritmo de controle, o módulo de controle de velocidade envia as velocidades a cada uma das rodas, logo em seguida faz-se a leitura da odometria. Este ciclo é repetido durante o movimento dos robôs, concomitantemente, a barra mantém-se travada. No robô seguidor, os módulos de controle de velocidade e de sensoriamento e travamento da barra é utilizado, com isso o seguidor pode acompanhar o líder e, no caso de mudança de liderança realiza o travamento da barra e retorno para que o líder realize uma nova trajetória.

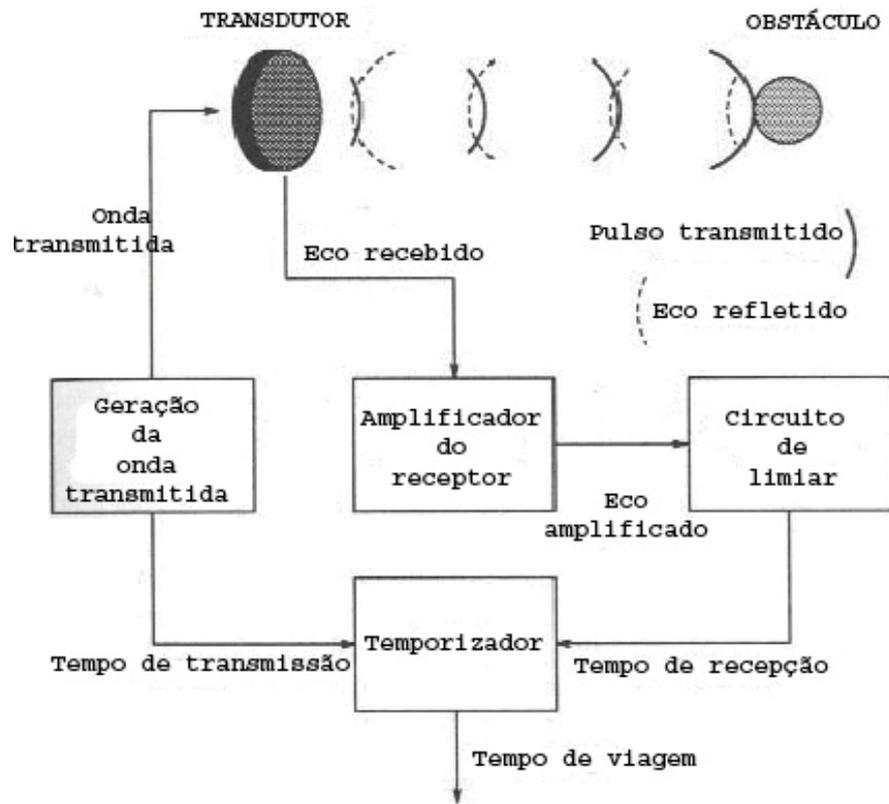


FIG. 4.16. Fluxograma do sonar.

5 RESULTADOS

Após a apresentação nos capítulos anteriores de como foi estruturada a proposta desta tese, neste capítulo é apresentado a implementação e os resultados. Primeiramente, o ambiente de simulação para validação e posterior implementação em hardware é composta por uma parte gráfica representando o ambiente no qual serão realizados o planejamento de trajetória e o controle do(s) robô(s) móveis, uma modelagem dos sensores ultrasônicos e uma modelagem do(s) robô(s).

O ambiente gráfico apresenta uma visão em duas dimensões do ambiente a ser estudado, a modelagem dos sensores reúne os atributos inerentes a sensores ultrasônicos, tais como: alcance, ângulo de abertura do lóbulo principal, resolução, posição e orientação em relação ao robô. No algoritmo utilizado, cada sensor retorna os valores de distância mínima a um obstáculo e as coordenadas dos pontos de intersecção dos feixes (representando a resolução) contidos no setor, o qual representa a abertura do lóbulo principal, com os pontos contidos na borda dos obstáculos, figura 5.1. A modelagem do(s) robô(s) considera a forma, posição e orientação em relação ao ambiente e sua cinemática direta e inversa mostrada nas figuras ?, além destes os módulos anteriormente citados de planejamento e controle dos movimentos, o campo potencial artificial e a lógica fuzzy.

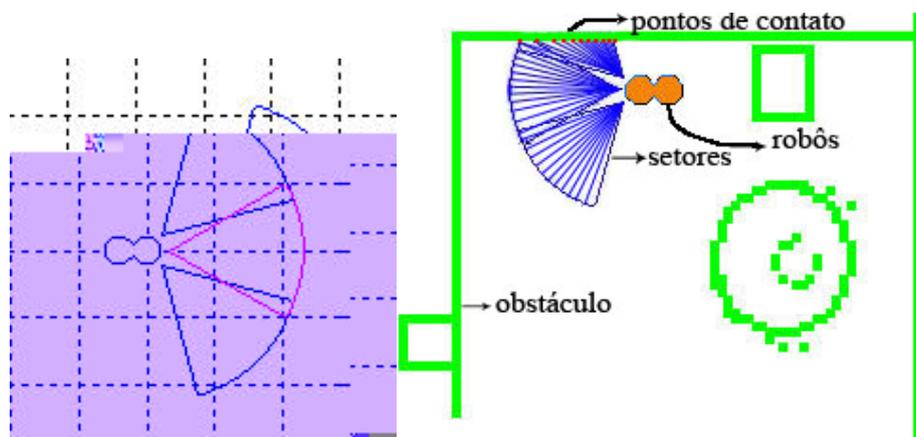


FIG. 5.1. Representação dos lóbulos do sonar e os pontos de contato.

Com campo potencial artificial, no qual evidencia o bom algoritmo para planejamento de trajetória, com o empecilho dos mínimos locais, os quais desvia os robôs da trajetória correta.

Na figura 5.2, a implementação somente da lógica fuzzy, mostrando um maior custo computacional, visto que os ajuste de orientação nem sempre convergem diretamente para o ponto objetivo, ao passo que o sistema híbrido, campo potencial mais lógica, otimiza o tempo de execução da tarefa, visto que elimina a falta de convergência da lógica fuzzy e o problema de mínimos locais do campo potencial, resultando em um algoritmo de planejamento de trajetória reduzido possibilitando sua utilização em microcontroladores embarcados, também, reduzindo custo de montagem.

Neste trabalho, foi implementado os algoritmos de campo potencial artificial e controlador fuzzy, após isto foi construído dois robôs, sendo que somente em um deles foi implementado a estrutura eletrônica e mecânica relativas ao sistema descrito nos capítulos anteriores.

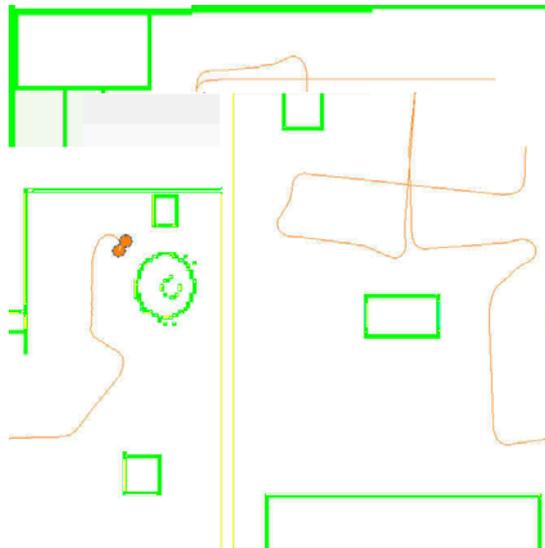
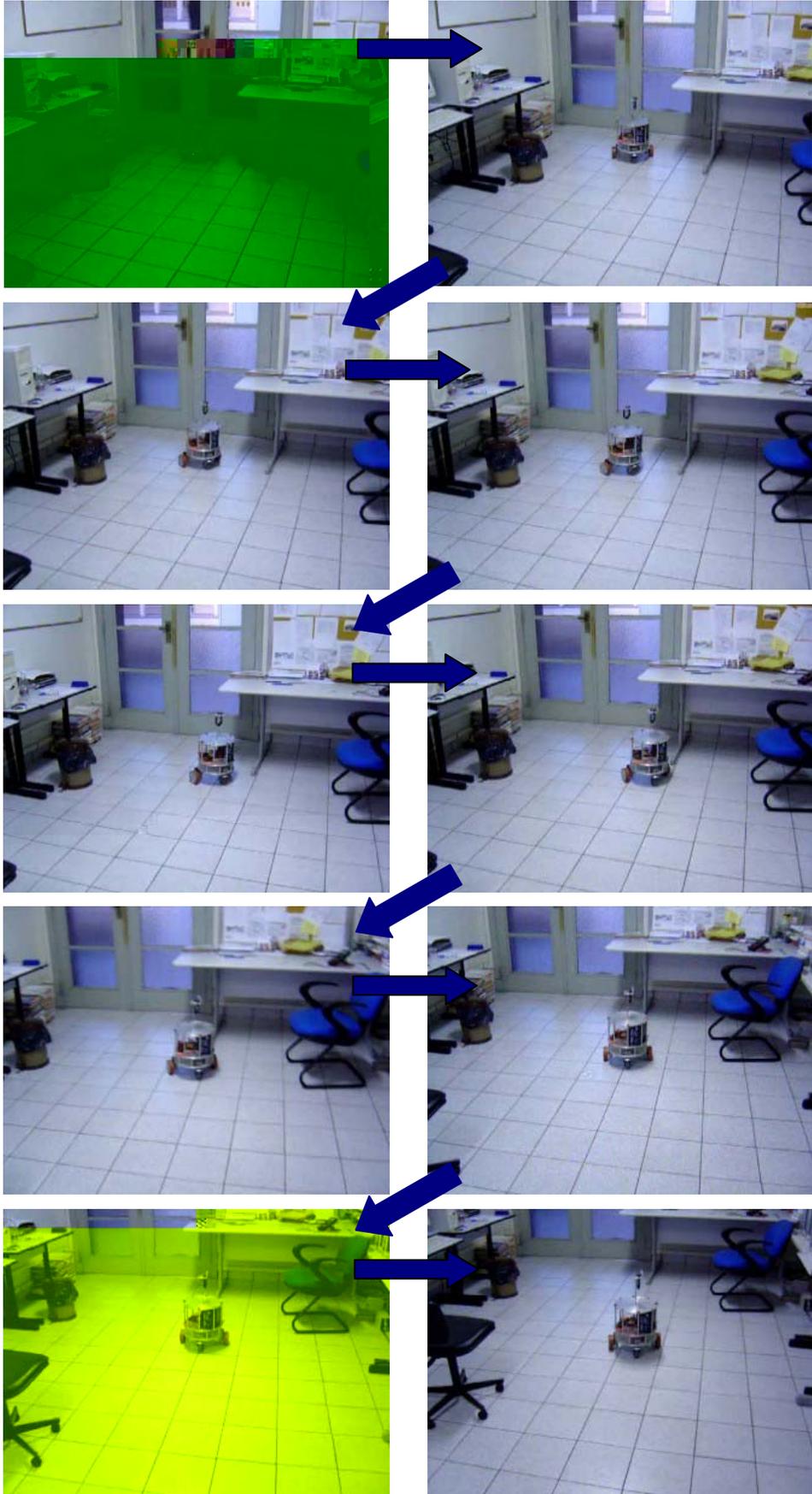


FIG. 5.2. Simulação somente com controlador fuzzy.

Após a simulação em duas dimensões, o protótipo foi implementado em uma estrutura de alumínio baseada no modelo do Nomad, contendo com quatro pares de sensores ultrasônicos na parte inferior, dois motores de corrente contínua acoplados a caixas de redução ligados as rodas e uma roda de apoio de giro livre. Sensores óticos acoplados aos eixos dos motores fornecem a realimentação odométrica a fim de corrigir possíveis mudanças de rota e, também utilizados no monitoramento da barra como a comunicação implícita entre líder e seguidor. Diferentemente de trabalhos relacionados, este somente faz uso de um único tipo de sensor em duas modalidades de algoritmo, capacitando um menor custo computacional embarcado. Na figura 5.3 é apresentado uma sequência de uma filmagem realizada somente o robô líder o qual está totalmente montado.



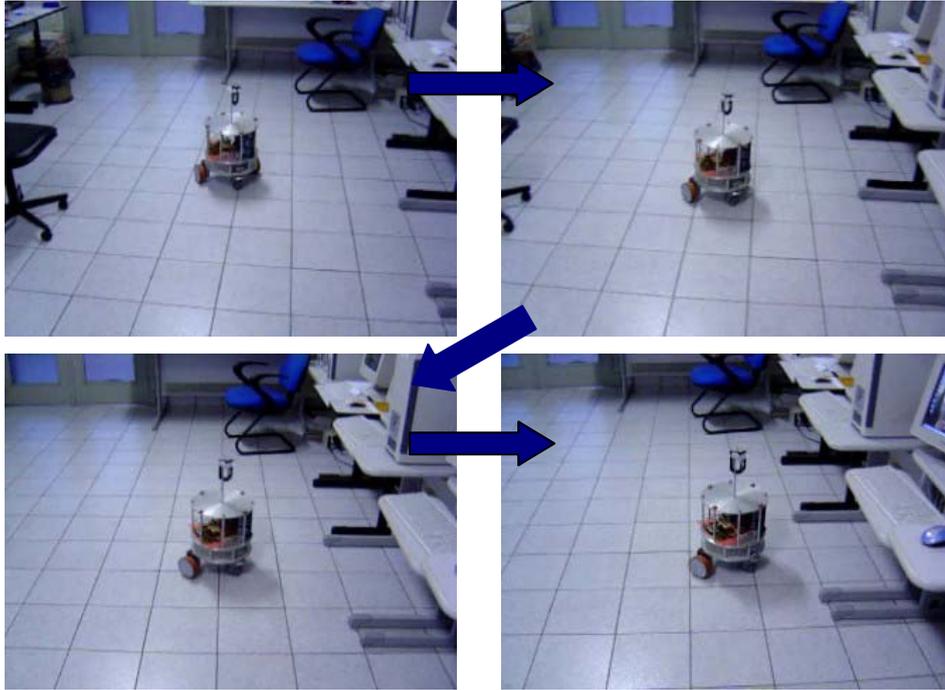


FIG. 5.3. Sequência de movimentos do robô líder em um ambiente desestruturado.

Na figura 5.4, são apresentados os diagramas esquemáticos do circuito contido no robô líder para testes dos algoritmos. Como afirmado anteriormente, foi utilizado somente um tipo de sensor para o monitoramento do ambiente, favorecendo também na redução do tempo gasto na execução do algoritmo de controle, além do hardware embarcado.

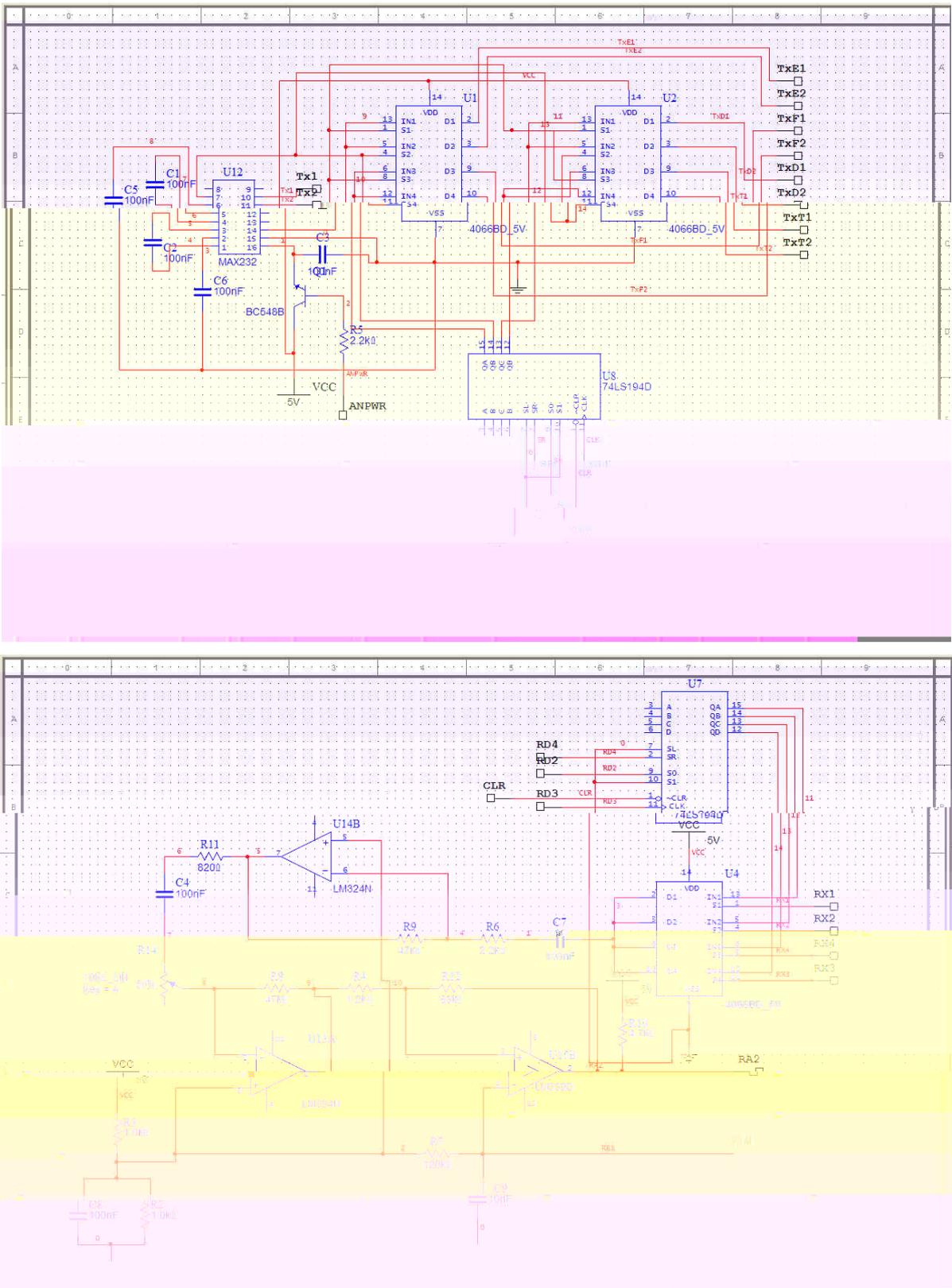


FIG. 5.4. Esquemático do circuito de controle do robô líder.

6 CONCLUSÃO

A abordagem utilizada neste trabalho focou, principalmente, o planejamento de trajetória e a cinemática dos robôs envolvida, através dos experimentos supracitados observou-se melhor eficiência do movimento e, maior autonomia da estrutura robótica, visto que somente um único tipo de sensor foi utilizado para dois algoritmos diferentes. O objetivo principal deste estudo evidenciou as dificuldades em combinar a fusão de sensores na percepção de obstáculos e a redução de custo computacional na realização de uma tarefa cooperativa. Procurou-se também, a criação de um protótipo viável para estudos futuros na área de Robótica do Instituto Militar de Engenharia.

6.1. TRABALHOS FUTUROS

Em princípio, há muitas linhas de pesquisa baseadas neste sistema, visto que o campo de inteligência computacional vem ocupando maior relevância nos sistemas robóticos. Algoritmos de inteligência computacional e as heurísticas podem ser combinadas tanto para a abordagem cinemática quanto para dinâmica, com a utilização de manipuladores com vários graus de liberdade. Além disso, fornece a possibilidade de implementação de diversos tipos de algoritmos para tarefas cooperativas mais complexas com a utilização de uma gama de manipuladores.

A partir de 2000, os trabalhos abordam a utilização de multi-agentes em hierarquias híbridas deliberativas/reativas com auto aprendizado [Newman, 2003] e o controle baseado em comportamento/emoções nas tarefas cooperativas [Dam, 1996]. Devido a flexibilidade e robustez da estrutura montada, também há a viabilidade de utilizá-los com estas abordagens, somente com poucas alterações, como colocação de um manipulador em cada um, e o emprego de métodos matemáticos de refinamento das medidas dos sonares (filtro de Kalman). Outra utilização, após a colocação de manipuladores com 3 ou 5 graus de liberdade é a coordenação de tarefas cooperativas com redes sem fio, como por exemplo, a montagem de prateleiras, visto que aumenta a maneabilidade do conjunto formado pelos dois robôs. Outro exemplo, é a implementação do sistema de controle híbrido apresentado nesta tese em uma

cadeira de rodas, já adquirida pelo Instituto Militar de Engenharia, a fim de fornecer a pessoas portadoras de deficiência maior autonomia na movimentação em ambientes semi-estruturados.

6.2. COMENTÁRIOS FINAIS

Este trabalho foi patrocinado em parte pelo CNPQ e pelo Ministério da Defesa. Resultados parciais desta pesquisa foram publicados nos anais do 18th International Congress of Mechanical Engineering, 2005, Ouro Preto – MG, pp 1 - 10.

7 REFERÊNCIAS BIBLIOGRÁFICAS

- ADAMS, M. ET AL., 1994, **Control and Localisation of a Post Distributing Mobile Robot**. *1994 International Conference on Intelligent Robots and Systems (IROS '94)*, Munich, Germany, Sept. 12-16, pp. 150-156.
- ALTAFINI, C., Speranzon, A.. **Backward line tracking control of a radio-controlled truck and trailer**. Proceedings of the IEEE International Conference on Robotics and Automation, pp. 169-174. 2001.
- AURENHAMMER, F.. **Voronoi diagrams: A survey of a fundamental geometric data structure**. *ACM Computing Surveys*, 23(3):345-405, 1991.
- BALCH, T., Arkin, R. C.. **Communication in reactive multiagent robotic systems**. *Autonomous Robots*, vol. 1, n° 1, pp. 1-25. 1994.
- BAREQUET, G., Dickerson, M. T., Goodrich, M. T.. **Voronoi Diagrams for Convex Polygon-Offset Distance Functions**. *Discrete and Computational Geometry*. Springer-Verlag. 2000.
- BARRAQUAND, J., Latombe, J.C.. **On Nonholonomic Mobile Robots and Optimal Maneuvering**. IEEE International Symposium on Intelligent Control. 1989.
- BEARD, R., McClain, T.. **Motion planning using potential fields**. 2003
- BENI, G., Wang, J.. **Theoretical problems for the realization of distributed robotic systems**. Proceedings of the IEEE International Conference on Robotics and Autonomous Systems, pp. 1914-1920. 1991.
- BLANCO, A., Delgado, M., Requena, I.. **Improved fuzzy neural networks for solving relational equations**. *Fuzzy Sets and Systems*, vol. 72, pp. 311-322. 1995.
- BORENSTEIN, J., Koren, Y.. **Motion Control Analysis of a Mobile Robot**. *Transactions of ASME, Journal of Dynamics, Measurement and Control*, vol. 109, n° 2, pp. 73-79. 1987.
- BORENSTEIN, J.. **Multi-layered control of a four-degree-of-freedom mobile robot with compliant linkage**. in: *iee international conference on robotics and automation*, Atlanta, p.3.7- 3.12, 1993.

- BRUMITT, B. L., Coulter, R. C. , Stentz, A.. **Dynamic trajectory planning for a cross-country navigator**. Spie mobile robots, 7, 1992, boston. proceedings... [s. 1.]
- BUCKLEY, J. J., Hayashi, Y.. **Fuzzy neural networks: A survey**. Fuzzy Sets and Systems, vol. 66, pp. 1-13.1994.
- BYRNE, R. H., Klarer, P. R., Pletta, J. B.. **Techniques for Autonomous Navigation**. Sandia Report SAND92-0457, Sandia National Laboratories. 1992.
- CHAIMOWICZ, L., Sugar, T., Kumar, V., Campos, M.F. M.. **An architecture for tightly-coupled multi-robot cooperation**. Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2292-2297. 2001.
- CHENAVIER, F., Crowley, J.. **Position estimation for a Mobile Robot using Vision and Odometry**. Proceedings of IEEE International Conference on Robotics and Automation, pp. 2588-2593. 1992.
- CHENG, P., Shen, Z. and LaValle, S. M.. **RRT-based trajectory design for autonomous automobiles and spacecraft**. *Archives of Control Sciences*, 11(3-4):167-194, 2001.
- CHOW, W.L.. **Über Systeme von Linearen Partiellen Differentialgleichungen ester Ordnung**. Math. Ann., vol. 117, pp. 98-105. 1939.
- COELHO, P., Nunes, U.. **Lie Algebra application to mobile robot control: a tutorial**. Institute of Systems and Robotics. University of Coimbra, Coimbra, 2003.
- COX, I. J.. **Blanche – An Experiment in Guidance and Navigation of an Autonomous Mobile Robot**. IEEE Transactions Robotics and Automation, vol. 7, pp. 193-204. 1991.
- CROWLEY, J. L.. **Navigation for an intelligent mobile robot**. IEEE journal of robotics and automation, v.ra-1, n.1, p.31-41, mar./1985.
- DAM, J. W. M. V., Kröse, B. J.A., Groen, F. C.A. **Adaptive sensor models**. in: ieee/sice/rsj int. conf. on multisensor fusion and integration for intelligent systems, Washington., p.705-712, 1996.
- DE LUCA, G. O.. **Modelling and control of nonholonomic mechanical systems in Kinematics and Dynamics of Multi-Body Systems** (J.Angeles, A.Kecsk emethy Eds.), Springer-Verlag, 1995.

- DEV, A., Kröse, B., Groen, F.. **Navigation of a mobile robot on the temporal development of the optic flow**. IEEE/RSJ/GI Int. Conf. on Intelligent Robots and Systems, p.558-563. 1997.
- EVANS, J. M.. **HelpMate: An Autonomous Mobile Robot Courier for Hospitals**. International Conference on Intelligent Robots and Systems, pp. 1695-1700. 1994.
- FORTUNE, S.. **A sweepline algorithm for Voronoi diagrams**. Algorithmica, vol. 2, pp. 153–174. 1987.
- FUKUDA, T., Nakagawa,S..**Dynamically Reconfigurable Robotic System**. Proceedings of the IEEE Int Conf Robotics and Automation. pp. 1581 – 1586. 1988
- GEVARTER, W. B. **Artificial Intelligence, Expert Systems, Computer Vision and Natural Language Processing**. New Jersey: Noyes Publications, 1984.
- GONZALEZ, J., Stentz, A., Ollero, A.. **An Iconic Position Estimator for a 2D Laser RangeFinder**. Proceedings of IEEE International Conference on Robotics and Automation, pp. 2646-2651. 1992.
- GOODRICH, M.. **Potential fields tutorial**. Class Notes. 2004
- GOURLEY, C., Trivedi, M.. **Sensor Based Obstacles Avoidance and Mapping for Fast Mobile Robots**. Proceedings of IEEE International Conference on Robotics and Automation, pp. 1306-1311. 1994.
- HEBB, D. O.. **The Organization of Behavior**. Wiley. 1949.
- HERMANN, R., Krener, A.J. **Nonlinear Controllability and Observability**, IEEE Tr. on Automatic Control, vol. 22(5), pp. 728-740. 1977.
- HOLLINGUM, J.. **Caterpillar make the earth move: automatically**. The Industrial Robot, vol. 18, n° 2, pp. 15-18. 1991.
- ISHIDA, Y., Asama, H., Tomina, S., Ozaki, K., Matsumoto, A., Endo, I.. **Functional complement by cooperation of multiple autonomous robots**. Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2476-2481. 1994.
- ISIDORI, A.. **Nonlinear Control Systems: An Introduction**. Springer-Verlag. 1985.

- JAHANBIN, M.R., Fallside, F. **Path planning using a wave simulation technique in the configuration space**. Artificial intelligence in engineering: robotics and process. Amsterdam: Elsevier. 1988.
- JAMSHIDI, M. ET AL.. **Applications of fuzzy logic: Towards high machine intelligence quotient systems**, Prentice-Hall, NJ, 1997.
- JONES, J. L., Flynn, A. M.. **Mobile robots: inspiration to implementation**. Wellesley: A K Peters, 1993.
- KHATIB, O.. **Real-time obstacle avoidance for manipulators and mobile robots**. International Journal of Robotics Research. 1986.
- KHATIB, O., Yokoi, K., Chang, K., Ruspini, D., Holmberg, R., Casal, A., Baader, A.. **Force strategies for cooperative tasks in multiple mobile manipulation systems**. Proceedings of the International Symposium on Robotics Research, pp. 333-342. 1995.
- KOMORIYA, K., Oyama, E.. **Position Estimation of a Mobile Robot Using Optical Fiber Gyroscope (OFG)**. International Conference on Intelligent Robots and Systems. Munich, Germany, pp. 143-149. 1994
- KOREN, Y.. **Robotics for engineers**. New York: McGraw- Hill, 1985.
- KOSUGE, K., Oosumi, T.. **Decentralized control of multiple robots handling an object**. Proceedings of the IEEE/RJS International Conference on Intelligent Robots and Systems, pp. 318-323. 1996.
- KUFFNER, J., LaValle, S. M.. **RRT-connect: An efficient approach to single-query path planning**. In IEEE Int'l Conf. on Robotics and Automation, pp. 995-1001, 2000.
- LAMIRAUX, F., Laumond, J. P., **Smooth motion planning for car-like vehicles**, IEEE Trans. on Robotics and Automation, vol. 17(4), pp. 498-502. 2001.
- LATOMBE, J.C.. **Robot Motion Planning**. Kluwer, 1991.
- LAVALLE, S. M.. **Planning Algorithms**. Online book, 1999-2003.
- LE PAPE, C.. **A Combination of Centralized and Distributed Methods for Multi-Agent Planning and Scheduling**. Proceedings of the IEEE International Conference on Robotics and Automation, pp. 488-493. 1990.

- LI, Z., Canny, J.F.. **Robot Motion Planning with Non-holonomic constraints**, Memo UCB/ERL M89/13, Electronics Research Lab., UC Berkeley. 1989.
- MCCOMB, G.. **The robot builder's bonanza: 99 inexpensive robotics projects**. New York: TAB Books, 1987.
- MCCULLOCH, W. S., Pitts, W. A.. **A logical calculus of the ideas imminent in nervous activity**. *Bulletins Mathematics Biophys.*, vol. 5, pp. 115-133. 1943.
- MCFARLAND, D., Bösser, T.. **Intelligent behavior in animals and robots**. Cambridge: Bradford book, 1993.
- MCKERROW, P. J.. **Introduction to robotics**. Sydney: Addison-Wesley, 1991.
- MINSKY, M., Papert, S.. **Perceptrons**. MIT Press. 1969.
- MURRAY, R., Sastry, S.. **Nonholonomic motion planning: steering using sinusoids**. *IEEE Transactions on Automatic Control*, vol.38, pp. 700-716. 1993.
- NANIWA, H. N. T., Arimoto, S.. **A quadtree-based path-planning algorithm for a mobile robot**. *Robotic Systems*. 1990.
- NEWMAN, P. M.. **Mobile Robotics. Lecture**. 2003.
- PARKER, L. E.. **Current state of the art in distributed autonomous mobile robotics**. *Proceedings of the International Symposium on Distributed Autonomous Robotics Systems*, pp. 3-12. 2000.
- PEREIRA, G. A. S., Pimentel, B. S., Chaimowicz, L., Campos, M. F. M.. **Coordination of multiple mobile robots in an object carrying task using implicit communication**. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 281-286. 2002.
- PEREIRA, G. A. S., Pimentel, Bruno S., Chaimowicz, Luiz, Campos, Mário F. M.. **Coordination of multiple mobile robots in an object carrying task using implicit communication**. *Proceedings of the IEEE International Conference on Robotics and Automation*. 2002.

- PREPARATA, F.; Shamos, M.. **Computational Geometry: An Introduction**. Springer-Verlag. 1985.
- RICH, E., Knight, K.. **Inteligência artificial**. 2. ed. São Paulo: Makron, 1994.
- ROSENBLATT, F.. **The perceptron: A probabilistic model for information storage and organization in the brain**. Physic. Rev., vol. 65, pp.386-408. 1958.
- RIMON, E., Koditschek, D. E.. **Exact robot navigation using artificial potential functions**. IEEE Transactions on Robotics and Automation, vol. 8(5). 1992.
- RUS, D., Donald, B., Jennings, J.. **Moving furniture with teams of autonomous robots**. Proceedings of the IEEE/RJS International Conference on Intelligent Robots and Systems, pp. 235-242. 1995.
- RUSSEL, R.A.. **Laying and sensing odor markings as a strategy for assisting mobile robot navigation tasks**. IEEE Robotics and Automation Magazine, vol. 2, n° 3, pp. 3-9. 1995.
- SAMAH-ABOU, M., K.V.. **Cooperative Frameworks for Multiple Mobile Robots**. Center of Intelligent Machines, McGill University, Canadá.
- SCHALKOFF, R. J.. **Artificial Intelligence: an engineering approach**. New York: McGraw Hill, 1990.
- SHAW, I. S., Simões, M.G. **Controle e Modelagem Fuzzy**. Edgard Blücher. São Paulo. 1999.
- SLEUMER, N. H., Tschichold-Gürman, N.. **Exact cell decomposition of arrangements used for path planning in robotics**. 1999.
- SPENCE, R., Hutchinson, S.. **An integrated architecture for robot motion planning and control in the presence of obstacles with unknown trajectories**. IEEE trans. on systems, man, and cybernetics, vol. 25, n° 1, pp.100-110. 1995.
- STIWEL, D. J., Bishop, B. E.. **A framework for decentralized control of autonomous vehicles**. Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2358-2363. 2001.

- SUGAR, T., Kumar, V.. **Decentralized control of cooperating mobile manipulators.** Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2916-2921. 1998.
- SUGAR, T., Kumar, V.. **Design and control of a compliant parallel manipulator for a mobile platform.** Proceedings of the ASME Design Engineering Technical Conference. 1998.
- TAKUSE, K., Inoue, H., Sato, K., Hagiwara, S.. **The design of the articulated manipulator with torque control ability.** In *Fourth International Symposium on Industrial Robots*, 1974.
- TONOUCHI, Y., Tsubouchi, T., and Arimoto, S.. **Fusion of Dead-reckoning Positions with a workspace model for a mobile robot by bayesian inference.** *International Conference on Intelligent Robots and Systems (IROS '94)*. Munich, Germany, pp. 1347-1354. 1994.
- VAUGHAN, R., Stoey, K., Sukthame, G., Mataric, M. J.. **Whistling in the dark: Cooperative trail following in uncertain localization space.** Proceedings of the Autonomous Agents. 2000.
- ZURADA, J. M.. **Introduction to Artificial Neural Systems.** West Publishing Company. 1992.

8 ANEXOS

9 ANEXO A – DESCRIÇÃO DOS CIRCUITOS, ESTRUTURA MECÂNICA DOS ROBÔS E MICROCONTROLADOR PIC16F877

Plataforma do Microcontrolador PIC

Um PIC16F877A-40P é responsável pelo controle direto do veículo, traduzindo os comandos vindos de sua programação em sinais elétricos que podem ser aplicados diretamente para controlar os motores ligados as rodas. A figura 1 mostra um diagrama de pinagem do processador PIC. Este PIC é um processador RISC de baixo consumo que opera em 20 MHz. Há somente 35 instruções no repertório do PIC16F877A, fazendo-o um simples, mas poderoso microcontrolador. O processador PIC é comercializado como uma plataforma versátil de desenvolvimento, permitindo projetos rápidos e desenvolvimento de soluções microcontroladas personalizáveis.

O diagrama esquemático do circuito para a placa do processador PIC é mostrado na figura 2

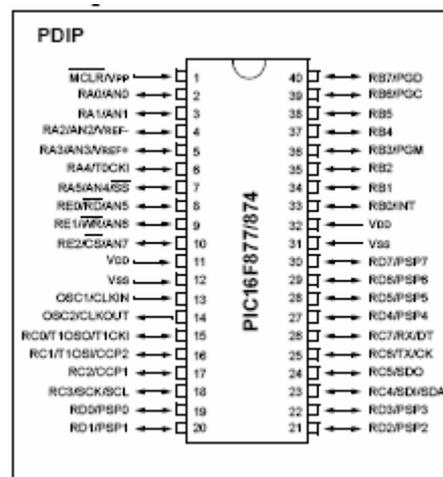


FIG. 1 Diagrama dos pinos do microprocessador PIC da Microchip

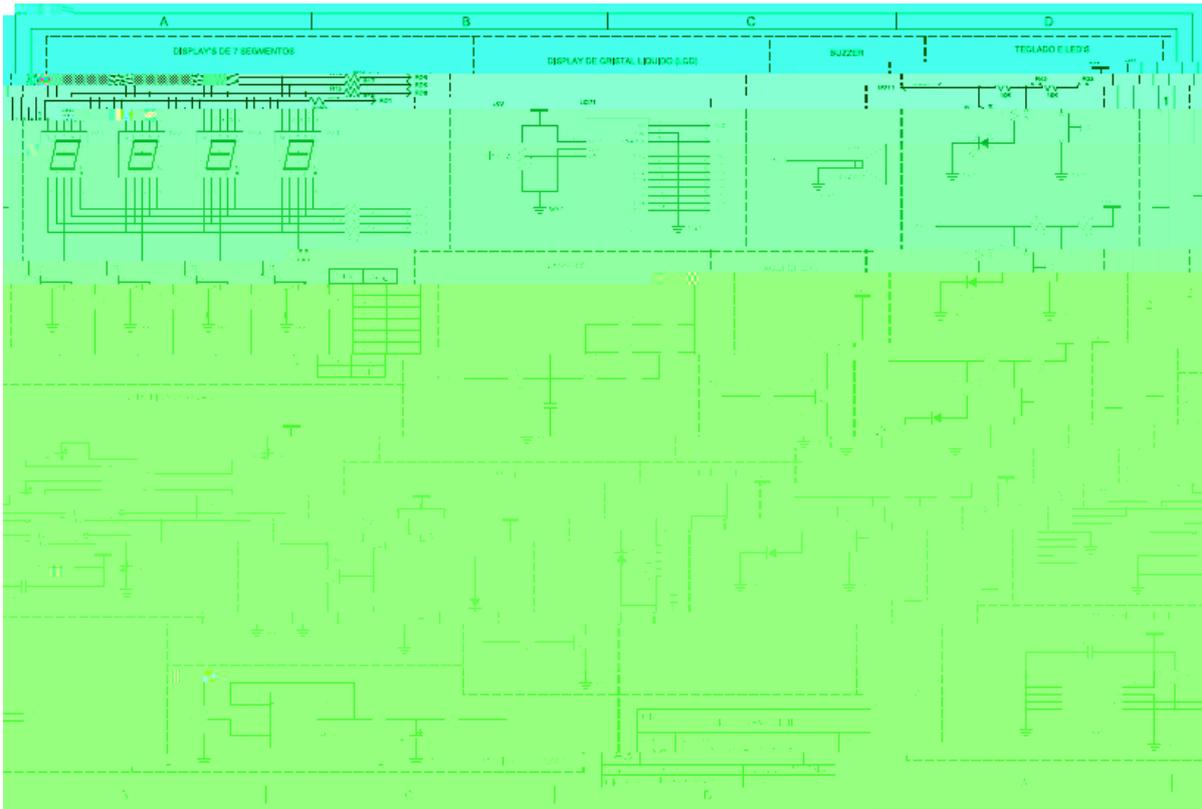


FIG. 2. Diagrama esquemático da placa de circuito do kit de desenvolvimento McLab2 da Mosaico.

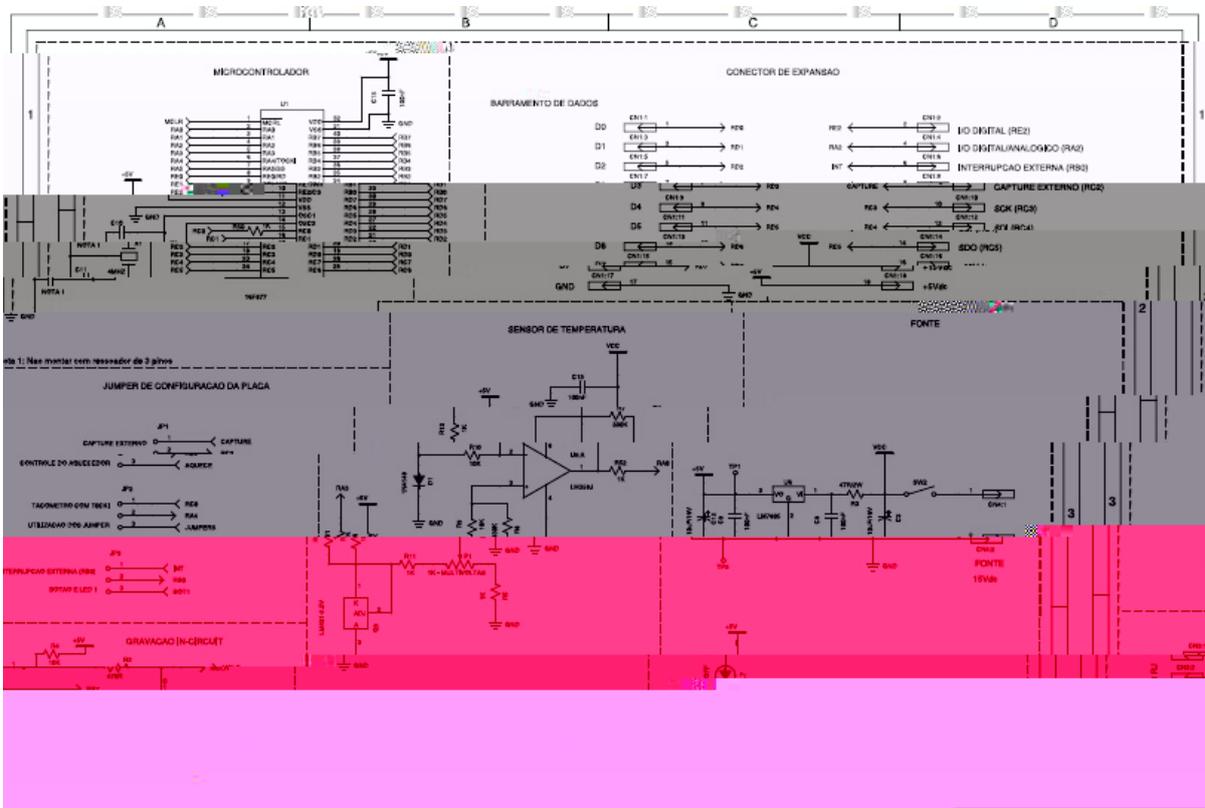


FIG. 3. Diagrama esquemático da Placa de circuito do kit McLab2 da Mosaico.

SENSORES ULTRASÔNICOS

A tecnologia de sensor a ser descrita neste capítulo é o sensor ultrasônico. Os sensores ultrasônicos fornecem os meios para determinar a distância relativa do veículo aos obstáculos no trajeto deste. A tecnologia de ultrassom é o meio mais barato de modelar as mais complexas tecnologias de radares baseados a laser e microondas que estão atualmente em uso no mundo em diversos tipos de produção de veículos.

Os sensores podem ser usados para detectar objetos a frente do veículo, em uma escala de 43 mm a 11 m com definição de 10mm. O sensor ultrasônico é de fato um conjunto (figura 3) projetado para uso em laboratório para experiências que envolvem distância.

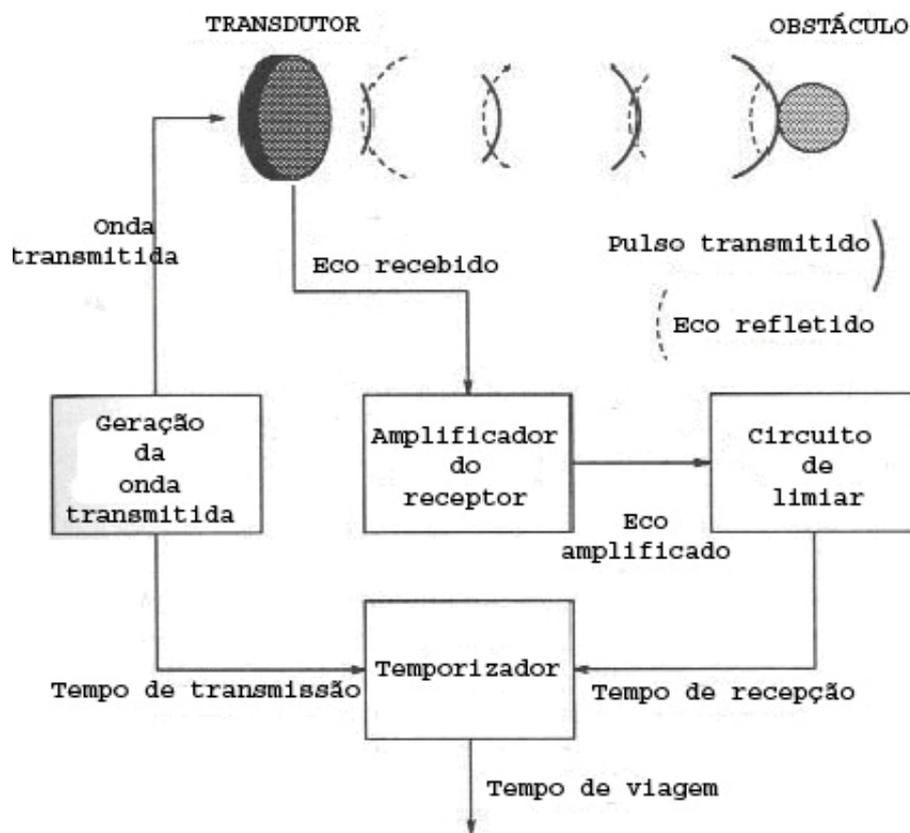


FIG. 4. Diagrama do sensor de proximidade

Operando a 40kHz, o transdutor ultrasônico emite um pulso ultrasônico, o qual é detectado pelo transdutor quando é refletido de obstáculos na frente do veículo, cujo lóbulo principal é visto na figura 4.

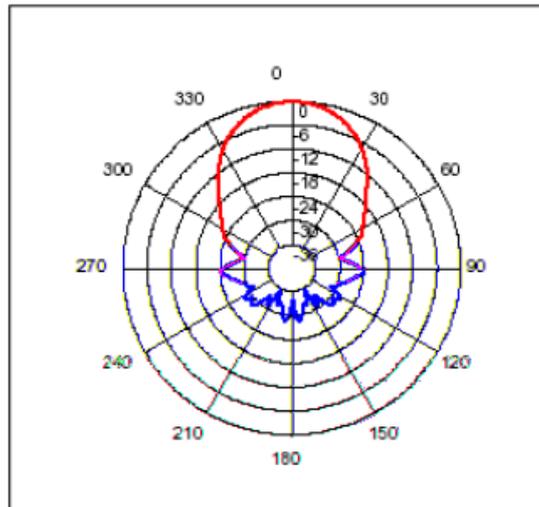


FIG. 5. Lóbulo central dos sensores.

O sensor ultrassônico é conectado à placa de circuito mostrada na figura 5, que controla o sincronismo dos pulsos e calcula a distância percebida do objetos ao sensor.

Utiliza-se os 35 registradores de 8 bits deste microcontrolador a fim de armazenar as distâncias coletadas dos sensores, as quais podem ser em polegadas, centímetros e, até em micro segundos. O alcance mínimo destes sensores é de 4,3 centímetros e máximo de 11 metros.

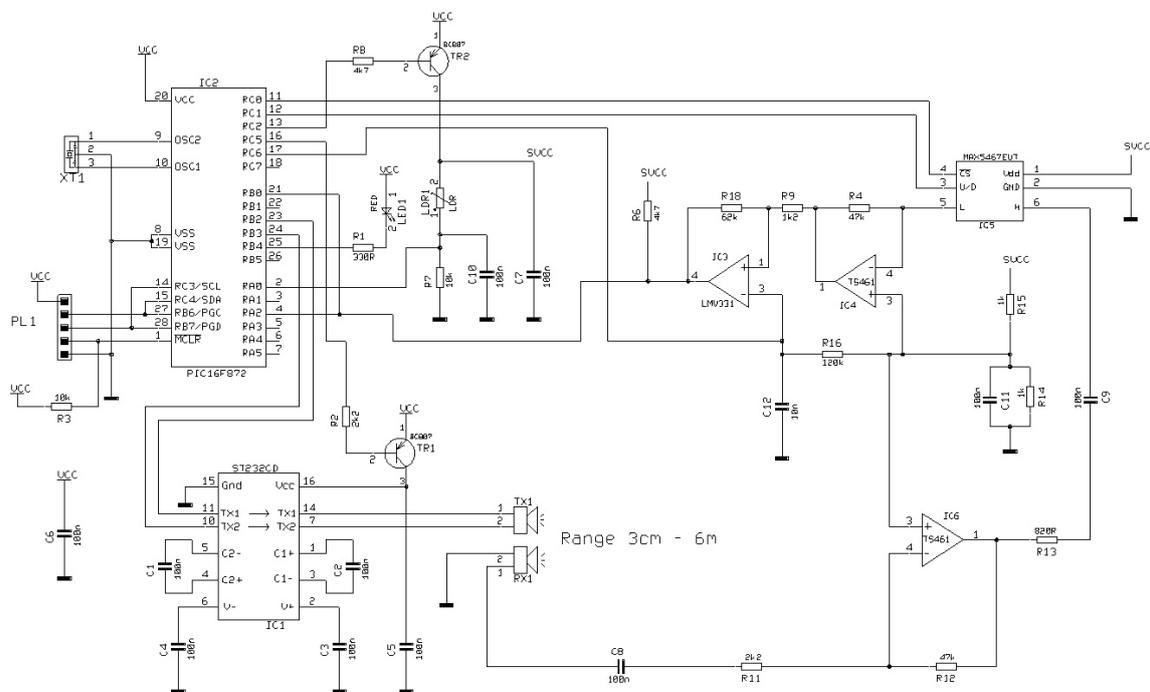


FIG. 6. Esquemático da placa do sonar SRF08.

ESTRUTURA MECÂNICA DOS ROBÔS

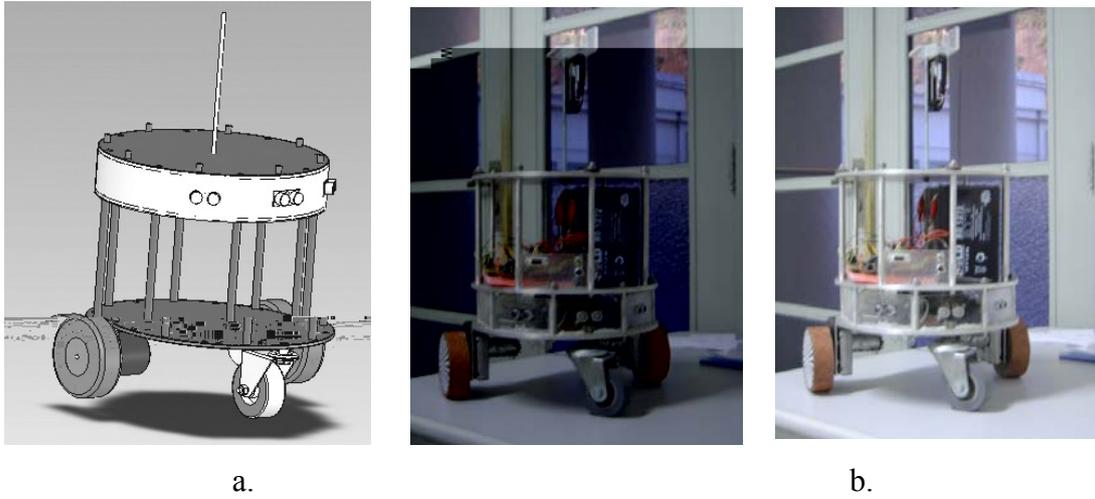


FIG. 7. Estrutura dos robôs autônomos. a. Modelo virtual; b. Modelos reais.

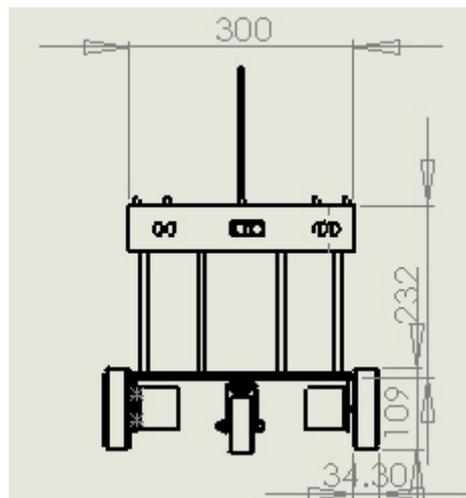


FIG. 8. Medidas do protótipo.

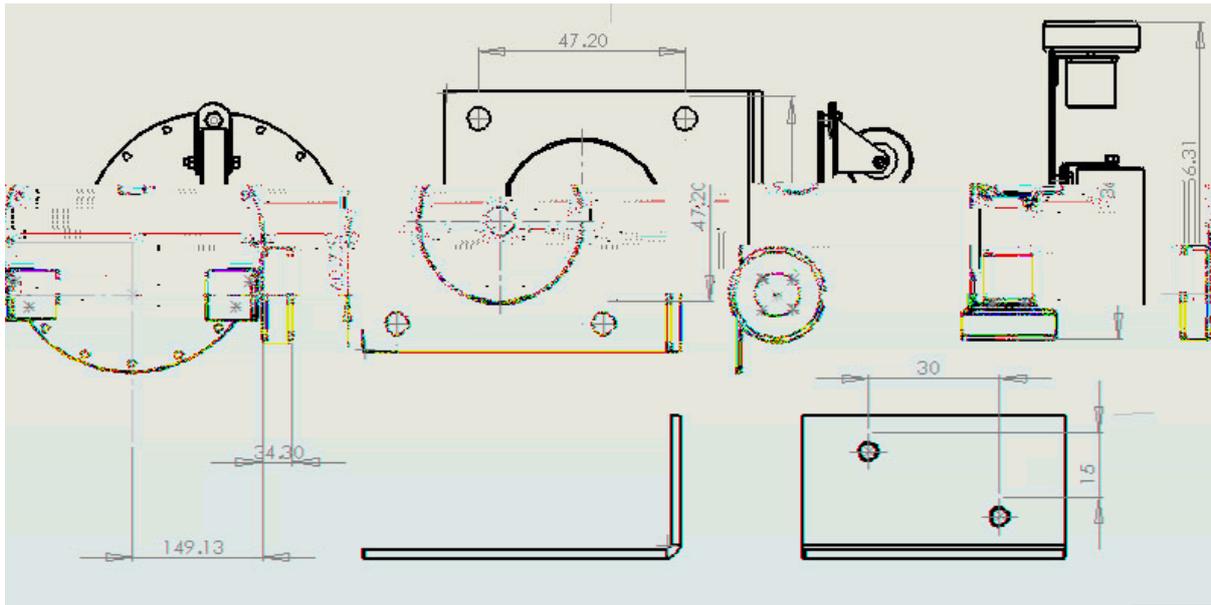


FIG. 9. Desenho das peças que compõe o robô.

10 ANEXO B - REDES NEURAIAS

REGRA DE APRENDIZAGEM DO PERCEPTRON

Os sistemas neurais artificiais podem ser considerados como modelos matemáticos simplificados dos sistemas cerebrais e funcionam como redes computacionais distribuídas em paralelo. Entretanto, ao contrário para calculadores convencionais, os quais são programados para executar uma tarefa específica, a maioria das redes neurais devem ser ensinadas, ou treinadas. Podem aprender novas associações, novas dependências funcionais e novos padrões. Embora os computadores desempenhem ambos os sistemas biológico e neural artificial para tarefas baseadas em operações precisas e de aritmética rápida, sistemas neurais artificiais representam a promessa da nova geração de redes de processamento de informações.

O estudo da computação baseada em atividades cerebrais tem suas raízes há 50 anos atrás no trabalho de McCulloch e Pitts [McCulloch, 1943] e ligeiramente mais tarde no famoso *Organização do comportamento* de Hebb [Hebb, 1949]. O trabalho mais recente em inteligência artificial variou entre aqueles que acreditam que sistemas inteligentes poderiam ser mais bem construídos em computadores modelados após o cérebro, e os que acreditam que inteligência foi fundamentalmente símbolo de processamento de símbolos do tipo facilmente modelado no computador de Von Neumann como Minsky e Papert [Minsky, 1969].

Em 1986, muito dos programas de pesquisa de redes neurais foram iniciados. A lista de aplicações que podem ser resolvidas por redes neurais expandiu de pequenos exemplos de teste a grandes tarefas práticas. Redes neurais de grande escala integradas a chips foram fabricados.

A longo prazo, poder-se-ia esperar que os sistemas neurais artificiais serão usados em aplicações envolvendo a visão, a fala, a tomada de decisão, e o raciocínio, mas também como processadores de sinal tais como filtros, detetores, e sistemas de controle de qualidade.

Definição 2.1.1 [Zurada, 1992] Sistemas neurais artificiais ou redes neurais são sistemas celulares físicos os quais adquirem, armazenam e utilizam conhecimento experimental.

O conhecimento está na forma de estados estáveis ou mapeamentos incorporados as redes que podem ser recordadas em resposta a apresentação de sugestões.

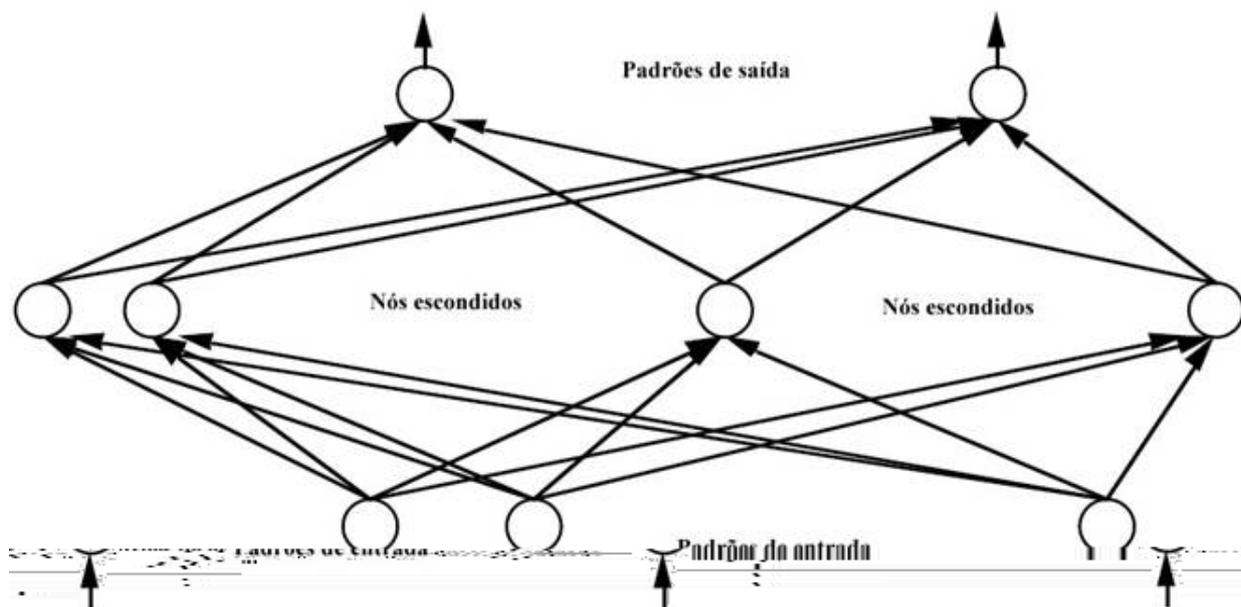


FIG. 10. Camadas escondidas ou internas das redes neurais.

Os elementos de processamento básico das redes neurais são chamados *neurônios artificiais*, ou simplesmente *neurônios* ou *nós*.

Cada unidade de processamento é caracterizado por um nível de atividade (representando o estado de polarização de um neurônio), um valor de saída (representando a taxa de disparo do neurônio), um conjunto de conexões de entrada (representando sinapses na célula e seus dendrito), um valor de viés (representando um nível de descanso interno do neurônio), e um conjunto de conexões de saída (representando projeções axonais do neurônio). Cada um destes aspectos da unidade são representados matematicamente por números reais. Portanto, cada conexão tem um peso associado (força sináptica) o qual determina o efeito da entrada no nível de ativação da unidade. Os pesos podem ser positivo (excitatório) ou negativo (inibitório).

O fluxo de sinal das entradas do neurônio, x_j , é considerado ser unidirecional indicados por setas, como é um fluxo de sinal de saída do neurônio. O sinal de saída do neurônio é dado pela seguinte relação

$$s = f(\langle w, x \rangle) = f(w^T x) = f\left(\sum_{j=1}^n w_j x_j\right)$$

onde w é o vetor de pesos. A função $f(w^T x)$ é frequentemente referida como uma função de ativação (ou transferência). Seu domínio é o conjunto dos valores de ativação, rede do modelo

do neurônio, portanto usa-se frequentemente esta função como $f(\text{rede})$. A variável *rede* é definida como um produto escalar do peso e dos vetores de entrada

$$\text{rede} = \langle w, x \rangle = w^T x = w_1 x_1 + K + w_n x_n$$

e no caso mais simples, o valor de saída s é calculado como

$$s = f(\text{rede}) = \begin{cases} 1 & \text{se } w^T x \geq \theta \\ 0 & \text{se para outros casos} \end{cases}$$

onde θ é chamado nível limiar e este tipo de nó é chamado uma *unidade limiar linear*.

A remoção do limiar da rede é muito fácil pelo aumento da dimensão dos padrões de entrada. Realmente, a identidade

$$w_1 x_1 + K + w_n x_n > \theta \Leftrightarrow w_1 x_1 + K + w_n x_n - 1 \times \theta > 0$$

significa que pelo acréscimo de um neurônio extra à camada de entrada com valor fixo de -1 e peso θ o valor do limiar torna-se zero. É porque daqui para frente suporemos que o limiar são sempre iguais a zero.

Definiremos agora o produto escalar de vetores n -dimensionais, os quais desempenham um papel muito importante na teoria de redes neurais.

Definição 2.1.2 Sejam $w = (w_1, \dots, w_n)^T$ e $x = (x_1, \dots, x_n)^T$ dois vetores de \mathbb{R}^n . O produto escalar (ou interno) de w e x , indicado por $\langle w, x \rangle$ ou $w^T x$, é definido por

$$\langle w, x \rangle = w_1 x_1 + K + w_n x_n = \sum_{j=1}^n w_j x_j$$

O problema de aprendizagem em redes neurais é simplesmente o problema de encontrar um conjunto de resistências de conexão (pesos) os quais permitam que a rede realize o cálculo desejado.

A rede é alimentada com um conjunto de pares entrada/saída de exemplo (um conjunto de treinamento) e é modificada suas conexões a fim de aproximar a função a qual o par entrada/saída tenham sido extraídos. As redes são então testadas para capacitá-la a generalização.

O procedimento de aprendizagem para correção de erro é, no conceito, simples o suficiente. O procedimento é o seguinte: Durante o treinamento uma entrada é colocada na rede e flui pela rede gerando um conjunto de valores nas unidades de saída. Então, a saída real é comparada com o objetivo desejado, e uma proporção é calculada. Se a saída e o objetivo igualem-se, nenhuma mudança é feita na rede. Entretanto, se a saída diferir do objetivo uma mudança deve ser feita em algumas das conexões.

A regra de aprendizagem do perceptron, introduzida por Rosenblatt [Rosenblatt, 1958], é um algoritmo de aprendizado de correção de erro típica de redes de camada simples e alimentação direta com função de ativação limiar linear.

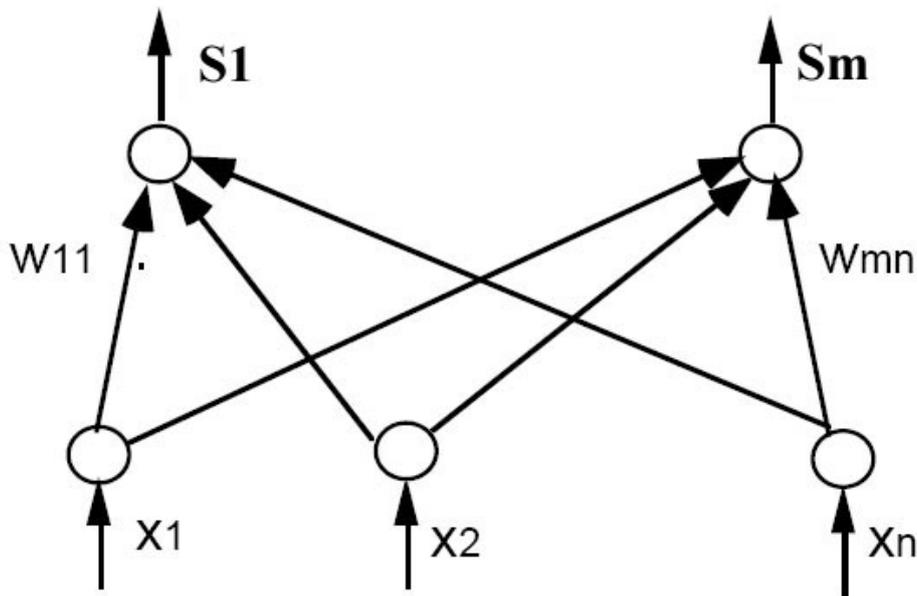


FIG. 11. Rede de camada simples e alimentação direta.

Geralmente, w_{ij} indica o peso da j -ésima unidade de entrada para a i -ésima unidade de saída e w_i indica o vetor de pesos do i -ésimo nó de saída. São fornecidos um conjunto de treinamento de pares entrada/saída

N°	valores de entrada	valores de saída desejada
1	$x^1 = (x_1^1, K, x_n^1)$	$y^1 = (y_1^1, K, y_m^1)$
M	M	M
K		$y^K = (y_1^K, K, y_m^K)$

O problema é encontrar os vetores de peso w_i tal que

$$s_i(x^K) = \text{sign}(\langle w_i, x^K \rangle) = y_i^K, \quad i = 1, K, m$$

para todos os padrões de treinamento k .

A função de ativação dos nós de saída é a função limiar linear da forma

$$s_i(x) = \text{sign}(\langle w_i, x \rangle) = \begin{cases} +1 & \text{se } \langle w_i, x \rangle \geq 0 \\ -1 & \text{se } \langle w_i, x \rangle < 0 \end{cases}$$

e os ajustes dos pesos no método de aprendizagem do perceptron é desempenhado por

$$w_i = w_i + \eta (y_i^k - \text{sign}(\langle w_i, x^k \rangle)) x^k, \quad i = 1, K, m$$

$$w_{ij} = w_{ij} + \eta (y_i^k - \text{sign}(\langle w_i, x^k \rangle)) x_j^k, \quad j = 1, K, n$$

onde $\eta > 0$ é a taxa de aprendizagem.

Desta equação segue que se a saída desejada for igual à saída calculada, $y_i^k = \text{sign}(\langle w_i, x^k \rangle)$, então o vetor de pesos do i -ésimo nó de saída permanece inalterado, isto é, w_i é ajustado se e somente se a saída calculada, $s_i(x^k)$, está incorreta. A aprendizagem pára quando todos os vetores de peso permanecerem inalterados durante um ciclo completo do treinamento.

Considere agora uma rede de camada simples com um nó de saída. Então os componentes de entrada dos padrões de treinamento podem ser classificados em duas classes disjuntas

$$C_1 = \{x^k \mid y^k = 1\}, \quad C_2 = \{x^k \mid y^k = -1\}$$

ou seja, x pertence à classe C_1 se existir um par entrada/saída $(x, 1)$, e x pertence à classe C_2 se existir um par entrada/saída $(x, -1)$.

Levando em consideração a definição da função de ativação é fácil ver que se busca um vetor de pesos w tal que

$$\langle w, x \rangle \geq 0 \text{ para cada } x \in C_1, \text{ e } \langle w, x \rangle < 0 \text{ para cada } x \in C_2.$$

Se tal vetor existir então o problema é dito *linearmente separável*.

Sumário 2.1.1 Algoritmo de aprendizagem do Perceptron.

São fornecidos K pares de treinamento organizados em um conjunto de treinamento

$$\text{onde } x^k = (x_1^k, \dots, x_n^k), y^k = (y_1^k, \dots, y_m^k), \quad k = 1, K, K.$$

- **Etapa 1.** $\eta > 0$ é escolhido
- **Etapa 2.** Os pesos w_i são iniciados com valores aleatórios pequenos, o erro de execução E é ajustado para 0, $k := 1$

• **Etapa 3.** O treinamento inicia aqui. x^k é apresentado, $x = x^k$, $y = y^k$ e saída s é calculada

$$s_i(x) = \text{sign}(\langle w_i, x \rangle), \quad i = 1, K, m$$

• **Etapa 4.** Os pesos são atualizados

$$w_i = w_i + \eta(y_i - \text{sign}(\langle w_i, x \rangle))x, \quad i = 1, K, m$$

• **Etapa 5.** O erro do ciclo cumulativo é calculado adicionando o erro atual a E

$$E = E + \frac{1}{2} \|y - s\|^2$$

• **Etapa 6.** Se $k < K$ então $k = k + 1$ e continua-se o treinamento retornando a **Etapa 3**, caso contrário vai-se para a **Etapa 7**

• **Etapa 7.** O ciclo de treinamento é terminado. Para $E = 0$ termina a sessão de treinamento. Se $E > 0$ então E é ajustado para 0, $k = 1$ e inicia-se um novo ciclo de treinamento indo para a **Etapa 3**.

O seguinte teorema mostra que se o problema tem soluções então o algoritmo de aprendizagem do perceptron encontrará um deles.

Teorema 2.1.1 (Teorema da convergência) se o problema for linearmente separável então o programa irá para a **Etapa 3** somente finitamente muitas vezes.

A REGRA DE APRENDIZADO DELTA

A idéia básica da regra de aprendizado delta é definir uma medida de desempenho total do sistema e então encontrar uma maneira de otimizar este desempenho. Na rede, define-se o desempenho do sistema como

$$E = \sum_{k=1}^K E_k = \frac{1}{2} \sum_{k=1}^K \|y^k - s^k\|^2$$

Isto é

$$E = \frac{1}{2} \sum_{k=1}^K \sum_{i=1}^m (y_i^k - s_i^k)^2 = \frac{1}{2} \sum_{k=1}^K \sum_{i=1}^m (y_i^k - \langle w_i, x^k \rangle)^2$$

onde i é o índice das unidades de saída; k é o índice dos pares entrada/saída a ser aprendidos; y_i^k indica o objetivo para uma unidade de saída particular em um padrão particular; $s_i^k = \langle w_i, x^k \rangle$ indica a saída real para a unidade i no padrão k ; e E é o erro total do sistema. O objetivo, então, é minimizar esta função. Verifica-se, se as funções de saída forem

diferenciáveis, que este problema tem uma solução simples: isto é, pode-se associar um valor a uma unidade em particular em proporção ao grau a qual muda a atividade da unidade conduzindo às mudanças no erro. Isto é, pode-se alterar os pesos do sistema em proporção a derivada do erro em relação aos pesos.

A regra para alterar pesos seguindo a representação do par entrada/saída (x^k, y^k) é dado pelo método do gradiente descendente, ou seja, minimiza-se a função de erro quadrático pelo uso do seguinte processo de iteração

$$w_{ij} = w_{ij} - \eta \frac{\partial E_k}{\partial w_{ij}}$$

onde $\eta > 0$ é a taxa de aprendizagem.

Calculando-se agora a derivada parcial da função de erro E_k em relação a w_{ij}

$$\frac{\partial E_k}{\partial w_{ij}} = \frac{\partial E_k}{\partial rede_i^k} \frac{\partial rede_i^k}{\partial w_{ij}} = -(y_i^k - s_i^k) x_j^k$$

onde $rede_i^k = w_{i1}x_1^k + \dots + w_{in}x_n^k$.

Isto é,

$$w_{ij} = w_{ij} + \eta (y_i^k - s_i^k) x_j^k \quad \text{for } j = 1, K, n.$$

Definição 2.2.3 O termo erro de sinal, indicado por δ_i^k é denominado delta, produzido pelo i -ésimo neurônio de saída é definido como

$$\delta_i^k = -\frac{\partial E_k}{\partial rede_i^k} = (y_i^k - s_i^k)$$

Para unidades de saída lineares δ_i^k é nada mais que a diferença entre os valores de saída desejado e calculado do i -ésimo neurônio.

Assim a regra de aprendizado delta pode ser escrita como

$$w_{ij} = w_{ij} + \eta \delta_i^k x_j^k, \quad \text{for } i = 1, K, m \text{ and } j = 1, K, n.$$

Uma vantagem chave dos sistemas de rede neural é que esta é simples, contudo procedimentos de aprendizagem poderosos podem ser definidos, permitindo que os sistemas adaptem-se a seus ambientes.

O caráter essencial de tais redes é que mapeiam padrões de entrada similares a padrões de saída similares.

Esta característica é o que permite que estas redes façam generalizações razoáveis e desempenhem razoavelmente em padrões que nunca antes tinham sido apresentados. A similaridade de padrões em um sistema conectado é determinado pelo sua sobreposição. A sobreposição em tais redes é determinada fora do próprio sistema de aprendizagem por mais que produza os padrões. A regra delta padrão essencialmente implementa o gradiente descendente no erro de somas-quadráticas para funções de ativação linear.

Esta regra assemelha-se a regra de treinamento do perceptron discreta. Pode ser também chamada de regra contínua de treinamento do perceptron.

Sumário 2.2.1 A regra de aprendizado delta com funções de ativação linear. São fornecidos K pares de treinamento organizados no conjunto de treinamento

$$\{(x^1, y^1), \dots, (x^K, y^K)\}$$

onde $x^k = (x_1^k, \dots, x_n^k)$ e $y^k = (y_1^k, \dots, y_m^k)$, $k = 1, \dots, K$.

- **Etapa 1.** $\eta > 0$, $E_{\max} > 0$ são escolhidos
- **Etapa 2.** Os pesos w_{ij} são iniciados em valores aleatórios pequenos, $k = 1$, e o erro corrente E é configurado para 0
- **Etapa 3.** O treinamento inicia aqui. A entrada x^k é apresentada, $x = x^k$, $y = y^k$, e a saída $s = (s_1, \dots, s_m)^T$ é calculada
|
para $i = 1, \dots, m$.
- **Etapa 4.** Os pesos são atualizados
- **Etapa 5.** O erro do ciclo cumulativo é calculado pelo acréscimo do erro atual a E

$$E = E + \frac{1}{2} \|y - s\|^2$$

• **Etapa 7.** O ciclo de treinamento é terminado. Para $E < E_{max}$ termina a sessão de treinamento. Se $E > E_{max}$ então E é configurado para 0 e inicia-se um novo ciclo de treinamento retornando a **Etapa 3**

Regra de aprendizado delta com função de ativação semi-linear

Em muitos casos práticos em vez das funções de ativação lineares utiliza-se as semi-lineares.

A tabela abaixo mostra que os tipos de funções de ativação mais frequentemente usados.

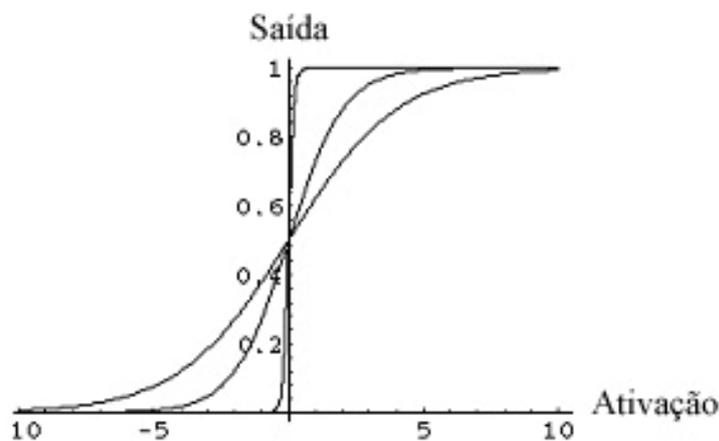


FIG. 12. Função de ativação sigmóide.

TAB. 1. Funções de ativação.

Linear	$f(\langle w, x \rangle) = w^T x$
Linear por partes	$f(\langle w, x \rangle) = \begin{cases} 1 & \text{se } \langle w, x \rangle > 1 \\ \langle w, x \rangle & \text{se } \langle w, x \rangle \leq 1 \\ -1 & \text{se } \langle w, x \rangle < -1 \end{cases}$
Degrau	$f(\langle w, x \rangle) = \text{sign}(w^T x)$
Sigmoidal unipolar	$f(\langle w, x \rangle) = 1 / (1 + \exp(-w^T x))$
Sigmoidal bipolar (1)	$f(\langle w, x \rangle) = \tanh(w^T x)$
Sigmoidal bipolar (2)	$f(\langle w, x \rangle) = 2 / (1 + \exp(w^T x)) - 1$

As derivadas das funções de ativação sigmoidal são usadas extensivamente nos algoritmos de aprendizagem.

Ø Se f é uma função de ativação sigmoidal bipolar da forma

$$f(t) = \frac{2}{1 + \exp(-t)} - 1$$

Então a seguinte igualdade mantém-se

$$f'(t) = \frac{2 \exp(-t)}{(1 + \exp(-t))^2} = \frac{1}{2}(1 - f^2(t))$$

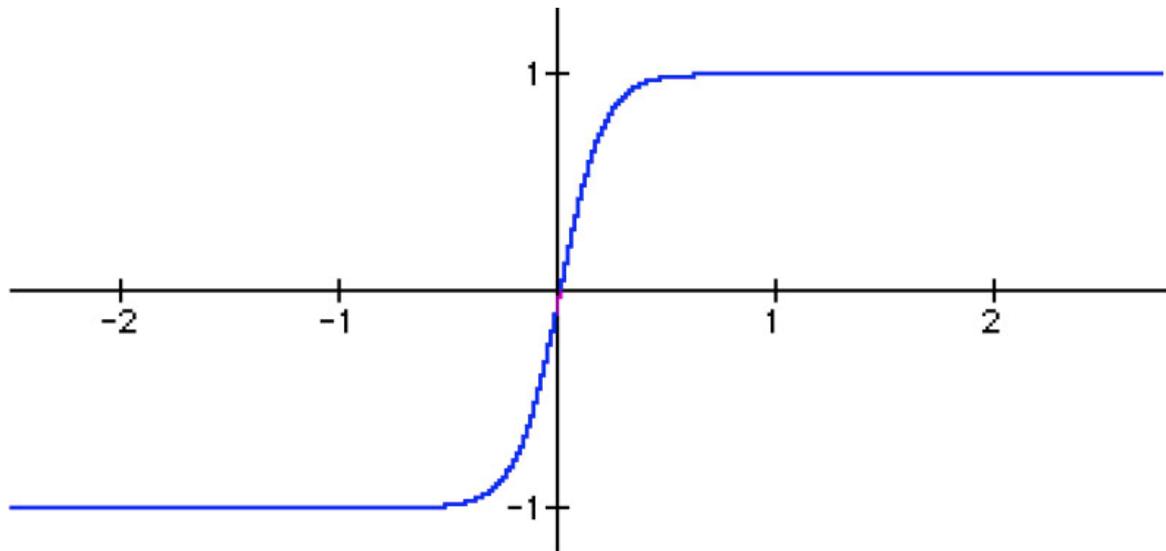


FIG. 13. Função de ativação bipolar.

Ø Se f é uma função de ativação sigmoidal unipolar da forma

$$f(t) = \frac{1}{1 + \exp(-t)}$$

Então f' satisfaz a seguinte igualdade

$$f'(t) = f(t)(1 - f(t))$$

REGRA DE APRENDIZAGEM DELTA GENERALIZADA

Agora trataremos em generalizar a regra de aprendizado delta para redes neurais em camada de alimentação direta. A arquitetura da rede de duas camadas considerada abaixo é mostrada na figura 13. Estritamente falando, as duas camadas de neurônios de processamento. Se, entretanto, as camadas dos nós são contadas, então a rede pode também ser rotulada como uma rede de três camadas. Não há acordo na literatura a respeito de qual abordagem deve ser usada para descrever as arquiteturas de rede. Neste texto utilizar-se-á o termo *camada* em referência ao número real de camadas de neurônios existentes e que processem. As camadas com neurônios cujas saídas não são diretamente acessíveis são chamadas internas ou escondidas. Assim a rede da figura 13 é uma rede de duas camadas, que pode ser chamada uma rede de única camada escondida.

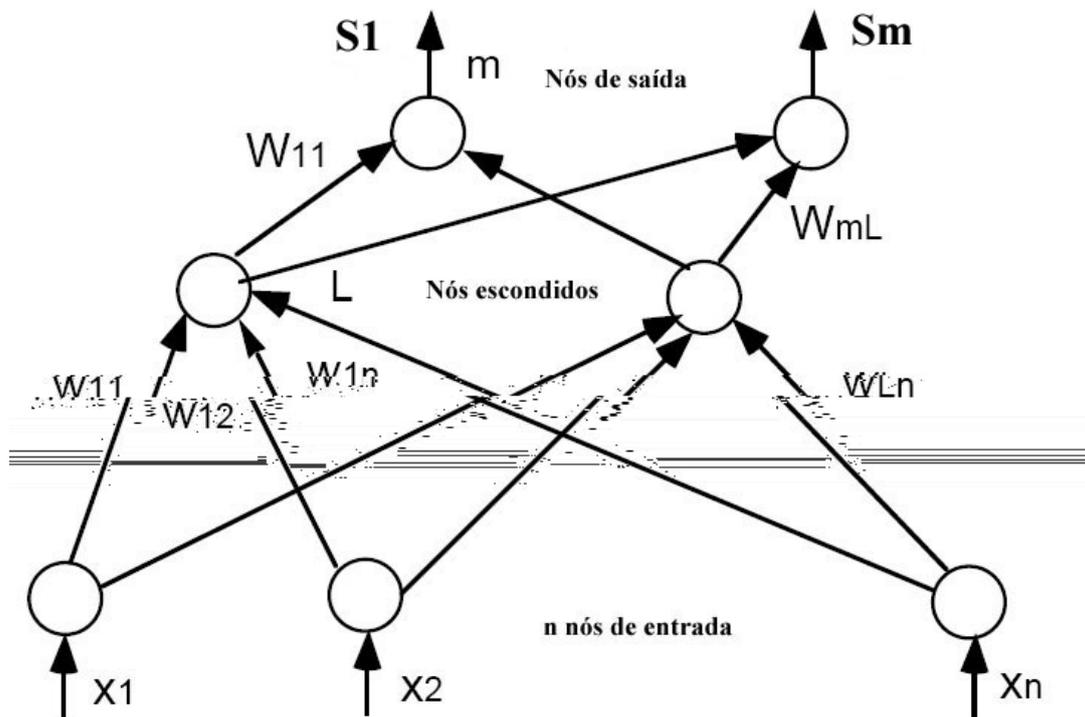


FIG. 14. Rede neural em camadas com duas camadas contínuas de perceptron.

A regra delta generalizada é o algoritmo de aprendizado supervisionado mais frequentemente usado em redes neurais multi-camada de alimentação direta. Por simplicidade considera-se somente uma rede neural com uma camada escondida e um nó de saída.

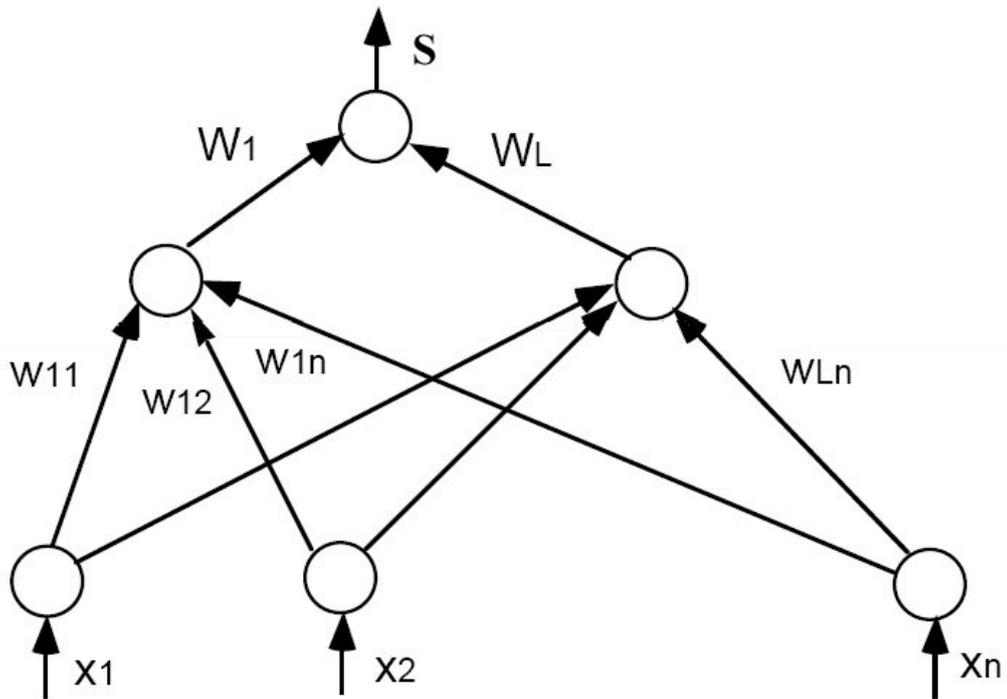


FIG. 15. Rede neural de duas camadas com um nó de saída

A medida de erro em um padrão de treinamento entrada/saída (x^k, y^k) é definido por

$$E_k(W, w) = \frac{1}{2}(y^k - S^k)^2$$

onde S^k é a saída calculada e a medida total do erro é

$$E(W, w) = \sum_{k=1}^K E_k(W, w)$$

Se um vetor de entrada x^k é apresentado a rede então gera a seguinte saída

$$S^k = \frac{1}{1 + \exp(-W^T s^k)}$$

onde s^k é o vetor de saída da camada escondida

$$s_l^k = \frac{1}{1 + \exp(-w_l^T x^k)}$$

e w_l indica o vetor de peso do l -ésimo neurônio escondido, $l = 1, \dots, L$.

A regra para mudança de pesos segue apresentação do par entrada/saída k que é calculado pelo método do gradiente descendente, ou seja, minimiza-se a função quadrática de erro usando o seguinte processo de iteração

$$W = W - \eta \frac{\partial E_k(W, w)}{\partial W},$$

$$w_l = w_l - \eta \frac{\partial E_k(W, w)}{\partial w_l},$$

,

para $l = 1, \dots, L$, e $\eta > 0$ que é a taxa de aprendizagem.

Usando-se a regra da cadeia para derivadas de funções compostas obter-se-á

$$\frac{\partial E_k(W, w)}{\partial W} = \frac{1}{2} \frac{\partial}{\partial W} \left[y^k - \frac{1}{1 + \exp(-W^T s^k)} \right]^2 = -(y^k - S^k) S^k (1 - S^k) s^k$$

isto é, a regra para mudança dos pesos da unidade de saída é

$$W = W + \eta (y^k - S^k) S^k (1 - S^k) s^k = W + \eta \delta_k s^k$$

ou seja

$$W_l = W_l + \eta \delta_k s_l^k$$

para $l = 1, \dots, L$, e utiliza-se a notação $\delta_k = (y^k - S^k) S^k (1 - S^k)$.

Calcula-se a derivada parcial de E_k em relação a w_l

$$\frac{\partial E_k(W, w)}{\partial w_l} = -S^k (1 - S^k) W_l s_l^k (1 - s_l^k) x^k$$

a regra para mudança dos pesos das unidades escondidas é

$$w_l = w_l + \eta \delta_k W_l s_l^k (1 - s_l^k) x^k, l = 1, \dots, L.$$

isto é

$$w_{lj} = w_{lj} + \eta \delta_k W_l s_l^k (1 - s_l^k) x_j^k, j = 1, K, n.$$

Sumário 2.3.1 A regra de aprendizagem delta generalizada (aprendizado de retropropagação do erro)

É fornecido o conjunto de treinamento

$$\{(x^1, y^1), \dots, (x^K, y^K)\}$$

onde $x^k = (x_1^k, \dots, x_n^k)$ e $y^k \in R, k = 1, \dots, K$.

- **Etapa 1.** $\eta > 0, E_{\max} > 0$ são escolhidos
- **Etapa 2.** Os pesos w são iniciados com valores aleatórios pequenos, $k = 1$, e o erro corrente E é configurado para 0
- **Etapa 3.** O treinamento inicia-se aqui. A entrada x^k é apresentada, $x = x^k, y = y^k$, e a saída S é calculada

$$S = \frac{1}{1 + \exp(-W^T s)}$$

onde s_l é o vetor de saída da camada escondida

$$s_l = \frac{1}{1 + \exp(-w_l^T x)}$$

- **Etapa 4.** Os pesos da unidade de saída são atualizados

$$W = W + \eta \delta s$$

onde $\delta = (y - S)S(1 - S)$.

- **Etapa 5.** Os pesos das unidades escondidas são atualizados

$$w_l = w_l + \eta \delta W_l s_l (1 - s_l) x, l = 1, K, L$$

- **Etapa 6.** O erro cumulativo do ciclo é calculado adicionando o erro atual a E

$$E = E + \frac{1}{2}(y - S)^2$$

- **Etapa 7.** Se $k < K$ então $k = k + 1$ e continua-se o treinamento retornando para a **Etapa 2**, caso contrário vai-se a **Etapa 8**.
- **Etapa 8.** O ciclo de treinamento é terminado. Para $E < E_{máx}$ termina-se a sessão de treinamento. Se $E > E_{máx}$ então $E = 0$, $k = 1$ e inicia-se um novo ciclo de treinamento retornando a **Etapa 3**.

APRENDIZAGEM: O VENCEDOR LEVA TUDO

O aprendizado não supervisionado de classificação é baseado em agrupar os dados de entrada. Nenhum conhecimento *a priori* é assumido estar disponível considerando um membro da entrada em uma classe particular. Além disso, as características gradualmente detectadas e uma história de treinamento serão usadas para auxiliar a rede na definição das classes e possíveis limites entre eles. *Agrupar é entendido ser reunir objetos similares e separar os não similares.*

Discutir-se-á a rede de Kohonen [Blanco, 1995], que classifica os vetores de entrada em um número específico de m categorias, de acordo com os grupos detectados no conjunto de treinamento

$$\{\mathbf{x}^1, \dots, \mathbf{x}^K\}.$$

O algoritmo de aprendizagem trata o conjunto de m vetores de peso como vetores variáveis que precisam ser aprendidos. Antes da aprendizagem, a normalização de todos (aleatoriamente escolhido) os vetores de peso é exigida.

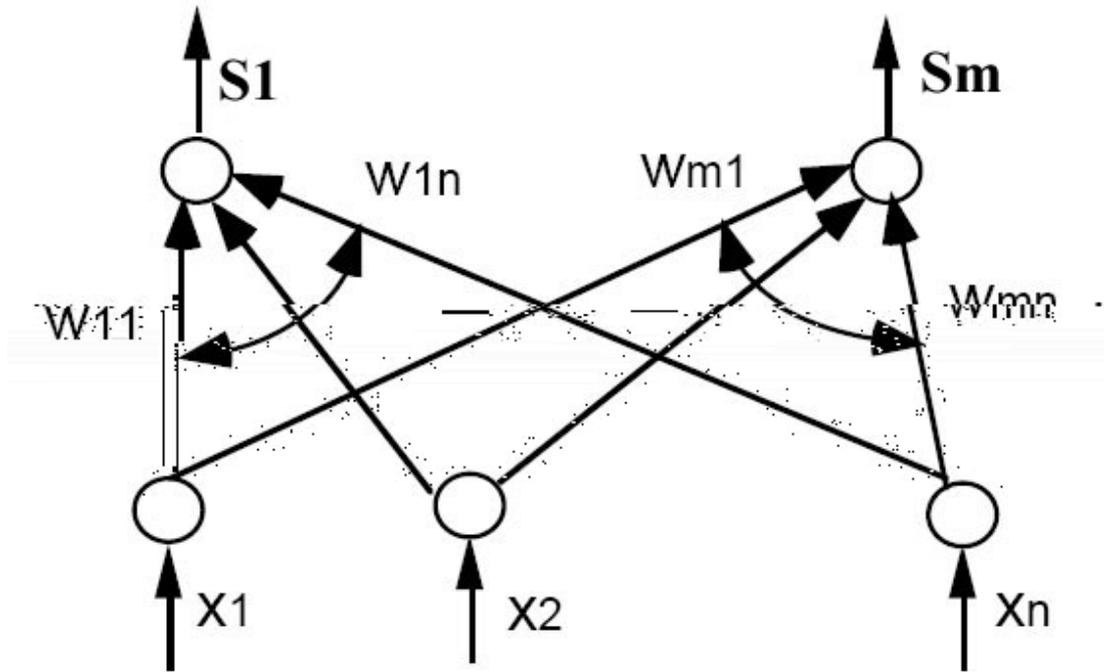


FIG. 16. Rede de aprendizagem o vencedor leva tudo

O critério de ajuste dos pesos para esta modalidade de treinamento é a seleção de w_r tal que

$$\|x - w_r\| = \min_{i=1, \dots, m} \|x - w_i\|$$

O índice r indica o número do neurônio *vencedor* que corresponde ao vetor w_r , que é a aproximação mais próxima da entrada atual x . Utilizando a igualdade

$$\|x - w_i\|^2 = \langle x - w_i, x - w_i \rangle = \langle x, x \rangle - 2\langle w_i, x \rangle + \langle w_i, w_i \rangle = \|x\|^2 - 2\langle w_i, x \rangle + \|w_i\|^2 = \|x\|^2 - 2\langle w_i, x \rangle + 1$$

pode-se inferir que a busca pelo mínimo de m distâncias corresponde a encontrar o máximo entre os m produtos escalares

$$\langle w_r, x \rangle = \max_{i=1, \dots, m} \langle w_i, x \rangle$$

Levando-se em consideração que, o produto escalar $\langle w_i, x \rangle$ é nada mais que a projeção de x na direção de w_i . Está claro que o vetor mais próximo w_i de x maior a projeção de x em w_i . Está claro que o vetor mais próximo w_i de x maior a projeção de x em w_i .

Observe que $\langle w_r, x \rangle$ é o valor de ativação do neurônio *ganhador* o qual tem a rede de maior valor, $i = 1, \dots, m$.

Após o neurônio vencedor ter sido identificado e declarado um vencedor, seu peso deve ser ajustado de modo que a distância $\|x - w_r\|$ é reduzida na etapa atual de treinamento. Portanto, $\|x - w_r\|$ deve ser reduzido, preferencialmente ao longo da direção do gradiente no espaço dos peso w_{r1}, \dots, w_{rn}

$$\begin{aligned} \frac{d\|x-w\|^2}{dw} &= \frac{d}{dw}(\langle x-w, x-w \rangle) = \frac{d}{dw}(\langle x, x \rangle - 2\langle w, x \rangle + \langle w, w \rangle) = \frac{d}{dw}(\langle x, x \rangle) - \frac{d}{dw}(2\langle w, x \rangle) + \frac{d}{dw}(\langle w, w \rangle) = \\ &= -2 \times \frac{d}{dw}(w_1 x_1 + K + w_n x_n) + \frac{d}{dw}(w_1^2 + K + w_n^2) = -2 \times \left[\frac{d}{dw_1}(w_1 x_1 + K + w_n x_n), K, \frac{d}{dw_n}(w_1 x_1 + K + w_n x_n) \right]^T + \\ &+ \left[\frac{d}{dw_1}(w_1^2 + K + w_n^2), K, \frac{d}{dw_n}(w_1^2 + K + w_n^2) \right]^T = -2(x_1, K, x_n)^T + 2(w_1, K, w_n)^T = -2(x - w) \end{aligned}$$

Parece razoável recompensar os pesos do neurônio vencedor com um incremento do peso no sentido negativo do gradiente, portanto na direção $(x - w_r)$. Tem-se assim

$$w_r = w_r + \eta(x - w_r)$$

onde η é uma pequena constante de aprendizado selecionada heurísticamente, geralmente entre 0,1 e 0,7. Os vetores de peso restantes são deixados inalterados.

Sumário 2.4.1 *O algoritmo de aprendizagem de Kohonen pode ser resumido nas três etapas seguintes*

- **Etapa 1.** $w_r = w_r + \eta(x - w_r), s_r = 1$, (r é o neurônio vencedor)
- **Etapa 2.** $w_r = w_r / \|w_r\|$ (normalização)
- **Etapa 3.** $w_i = w_i, s_i = 0, i \neq r$ (perdedores não são afetados)

Deve-se notar que da identidade

$$w_r = w_r + \eta(x - w_r) = (1 - \eta)w_r + \eta x$$

segue que o vetor de peso atualizado é uma combinação linear convexa dos pesos antigos e os vetores padrão.

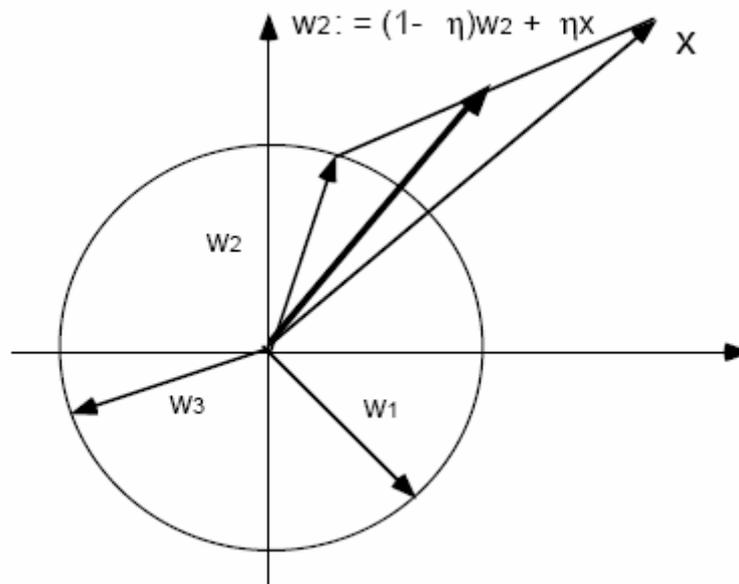


FIG. 17. Atualização dos pesos do neurônio vencedor

Ao final do processo de treinamento o vetor de pesos final aponta para o centro de gravidade das classes. *A rede será somente treinável se as classes/agrupamentos dos padrões seja linearmente separável de outras classes por hiperplanos passando pela origem.*

Para assegurar a separabilidade dos conjuntos com um conjunto de números desconhecidos *a priori* de grupos de treinamento, treinamento não supervisionado pode ser desempenhado por um número excessivo de neurônios, o qual fornece uma margem segura da correta separabilidade.

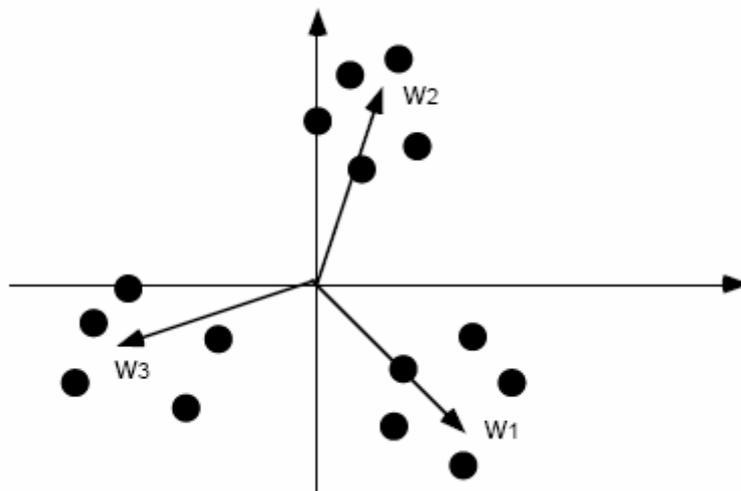


FIG. 18. Vetores de peso final aponta para o centro de gravidade das classes

Durante o treinamento, alguns neurônios provavelmente não desenvolveram seus pesos, e se seus pesos mudassem caoticamente, eles não serão considerados como indicativo de grupos.

Conseqüentemente tais pesos podem ser omitidos durante a fase de retorno, desde que sua saída não forneça qualquer informação de agrupamento essencial. Os pesos dos neurônios restantes devem estabelecer-se em valores que são indicativos de agrupamentos.

Outra extensão de aprendizagem é possível para esta rede quando a classe apropriada para alguns padrões é conhecida *a priori* [Buckley, 1994]. Embora isto signifique que a codificação dos dados em pesos está tornando-se então supervisionada, esta informação acelera o processo de aprendizagem significativamente. Os ajustes do peso são calculados no modo supervisionado como em (2.1), ou seja,

$$\Delta w_{ij} = \eta(x - w_r);$$

e somente para classificações corretas. Para respostas de agrupamento impróprias da rede, o ajuste de peso carrega o sinal oposto comparado a (2.2). Isto é, $\eta > 0$ para respostas apropriadas do nó, e $\eta < 0$ caso contrário, no modo de aprendizagem supervisionado para a camada de Kohonen.

APLICAÇÕES DE REDES NEURAIIS ARTIFICIAIS

Há grandes classes de problemas que parecem ser mais amenos à solução por redes neurais do que por outras técnicas disponíveis. Estas tarefas frequentemente envolvem ambigüidade, tal como que é inerente ao reconhecimento de caracteres manuscritos. Problemas deste tipo são difíceis de rastrear com métodos convencionais tais como filtragem por comparação ou classificação por vizinhança mais próxima, em parte por causa da métrica usada pelo cérebro para comparar padrões pode não ser muito relacionada por proximidade àqueles escolhidos por um engenheiro projetando um sistema de reconhecimento. Do mesmo modo, por falta de regras confiáveis de reconhecimento de padrões, projetos em lógica fuzzy e redes neurais enfrentam também a dificuldade e, às vezes, a impossibilidade de encontrar descrições aceitáveis de relações complexas que governam a inclusão de classes. Em sistemas com rede neural treinável, estas relações são abstraídas diretamente dos dados de treinamento. Além disso, devido as redes neurais poderem ser construídas com números de entradas e saídas que variam em milhares, podem ser usados para resolver problemas que exigem a

consideração de mais variáveis de entrada do que poderia ser utilizada praticamente por muitas outras abordagens, tal como lógica fuzzy. Deve-se observar, entretanto, que as redes neurais não trabalharão bem na solução de problemas para os quais conjuntos gerais e suficientemente grandes de dados de treinamento não são obtidos.

11 ANEXO C – DEMONSTRAÇÃO DAS EQUAÇÕES 25 E 26

Sabendo que

$$z = T(x) = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} h(q) \\ L_f h(q) \\ \tilde{h}(q) \end{bmatrix} = \begin{bmatrix} h(q) \\ \Phi(q)v \\ \tilde{h}(q) \end{bmatrix}, \quad \dot{q} = S(q)v(t) \text{ e } J_h = \frac{\partial h}{\partial q} \text{ calcula-se as}$$

derivadas de z :

$$\dot{z}_1 = \frac{\partial h}{\partial q} \dot{q} = J_h S(q)v, \text{ dando } \dot{z}_1 = \Phi(q)v = z_2, \text{ sabendo que } \Phi(q) = J_h(q)S(q)$$

$$\dot{z}_2 = \dot{\Phi}(q)v + \Phi(q)\dot{v}, \text{ e tomando } \dot{v} = u \text{ em (9) resulta } \dot{z}_2 = \dot{\Phi}(q)v + \Phi(q)u = u$$

$$\dot{z}_3 = \frac{\partial \tilde{h}}{\partial q} \dot{q} \rightarrow \dot{z}_3 = J_{\tilde{h}} S(q)v(t). \text{ Para eliminar } v \text{ na equação de } \dot{z}_3 \text{ e sabendo que}$$

$z_2 = \Phi(q)v \Rightarrow v = z_2 \Phi^{-1}(q)$, e considerando que \dot{z}_3 resulta de termos Jacobianos, é necessário eliminar $\Phi(q)$ como segue:

$$\dot{z}_3 = \frac{\partial \tilde{h}}{\partial q} \dot{q} \rightarrow \dot{z}_3 = J_{\tilde{h}} S \Phi^{-1}(q) z_2 \left| \begin{array}{l} \Rightarrow \dot{z}_3 = J_{\tilde{h}} S (J_h S)^{-1} z_2 \\ \Phi(q) = J_h S \end{array} \right.$$

Observa-se neste caso que z_3 e \dot{z}_3 representam a dinâmica interna do sistema, que é a parte não observável.

12 ANEXO D – DEMONSTRAÇÃO DO CÁLCULO DE S_3

Sabendo que o cálculo dos “Lie brackets” é obtido através de $s_3 = [s_1, s_2] = \frac{\partial s_2}{\partial q} s_1 - \frac{\partial s_1}{\partial q} s_2$,

então

$$\frac{\partial s_2}{\partial q} s_1 = \begin{bmatrix} 0 & 0 & c(-bsen\phi + d \cos \phi) & 0 & 0 \\ 0 & 0 & c(b \cos \phi + dsen\phi) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} c(b \cos \phi - dsen\phi) \\ c(bsen\phi + d \cos \phi) \\ c \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} c^2(-bsen\phi + d \cos \phi) \\ c^2(b \cos \phi + dsen\phi) \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\frac{\partial s_1}{\partial q} s_2 = \begin{bmatrix} 0 & 0 & c(-bsen\phi - d \cos \phi) & 0 & 0 \\ 0 & 0 & c(b \cos \phi - dsen\phi) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} c(b \cos \phi + dsen\phi) \\ c(bsen\phi - d \cos \phi) \\ -c \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} c^2(bsen\phi + d \cos \phi) \\ -c^2(b \cos \phi - dsen\phi) \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Observando o fato que $s_3 = \frac{\partial s_2}{\partial q} s_1 - \frac{\partial s_1}{\partial q} s_2$ então

$$s_3 = \begin{bmatrix} c^2(-bsen\phi + d \cos \phi) - c^2(bsen\phi + d \cos \phi) \\ c^2(b \cos \phi + dsen\phi) + c^2(b \cos \phi - dsen\phi) \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -c^2(2bsen\phi) \\ c^2(2b \cos \phi) \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

e como $c = \frac{r}{2b}$, obtém-se finalmente

$$s_3 = \begin{bmatrix} -\frac{r}{2b}c(2b \operatorname{sen} \phi) \\ \frac{r}{2b}c(2b \cos \phi) \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -rc \operatorname{sen} \phi \\ rc \cos \phi \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

A demonstração para s_4, s_5, s_6, s_7 e s_8 são similares.

13 ANEXO E - DEFINIÇÕES DE ESTABILIDADE

Definição de estabilidade: O estado de equilíbrio $x = 0$ do sistema dinâmico é dito ser estável se, para qualquer $R > 0$, existir $r > 0$, tal que se $\|x(0)\| < r$

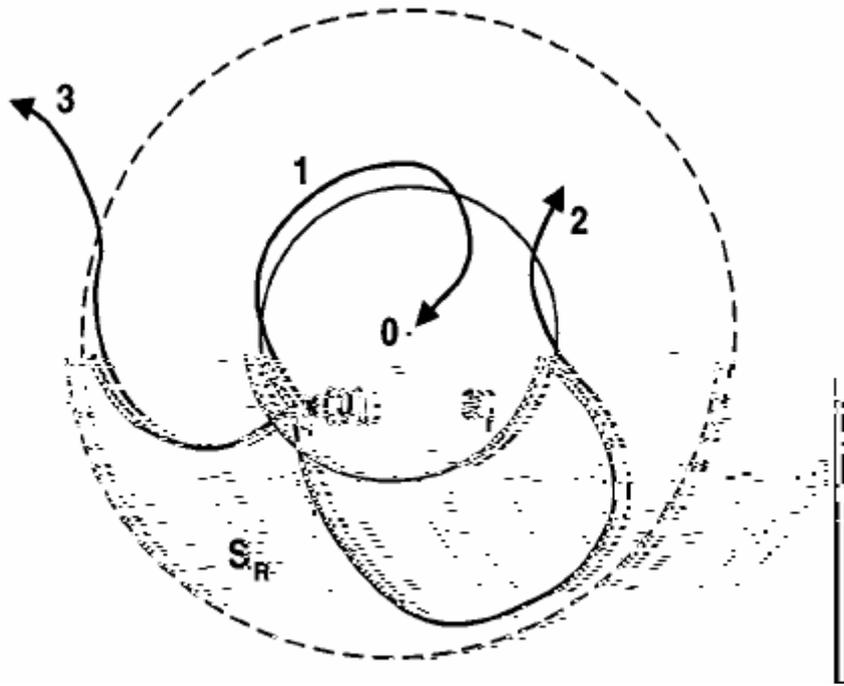


FIG. 19. Gráfico de estabilidade.

14 ANEXO F - TEOREMAS RELACIONADOS A DIAGRAMA DE VORONOI

Vimos que cada célula do diagrama é a intersecção de um número de semi-planos, então o Diagrama de Voronoi é uma subdivisão do plano cujas bordas são retas. Segue abaixo teoremas derivados do Diagrama de Voronoi.

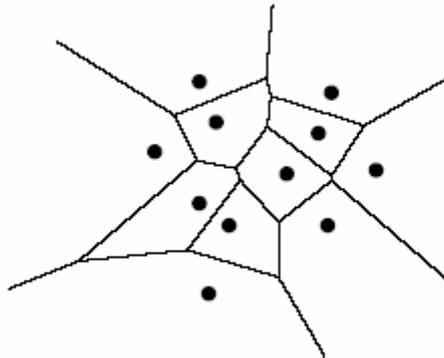


FIG. 20: Representação do Diagrama de Voronoi.

Teorema 1: Seja P o conjunto de n “sites” no plano. Se todos os “sites” são colineares então $\text{Vor}(P)$ consiste em $n - 1$ linhas paralelas. Caso contrário, $\text{Vor}(P)$ é conectado e suas bordas são segmentos com semi-linhas.

Demonstração:

A primeira parte do teorema é fácil provar, então assumindo que nem todos os “sites” em P são colineares. Primeiro mostramos que as bordas de $\text{Vor}(P)$ são tanto segmentos quanto semi-linhas. Sabe-se que as bordas de $\text{Vor}(P)$ são partes de linhas retas, sabidamente partes dos bissetores entre pares de “sites”. Agora suponha por contradição que há uma borda e de $\text{Vor}(P)$ que é uma reta. Seja e a fronteira das células de Voronoi $C(p_i)$ e $C(p_j)$. Seja $p_k \in P$ um ponto que não é colinear com p_i e p_j . O bissetor de p_j e p_k não é paralelo a e e, conseqüentemente, intercepta e . Mas então a parte de e que se encontra no interior de $h(p_k, p_j)$ não pode ser fronteira de $C(p_j)$, porque é mais próximo de p_k do que p_j , uma contradição.

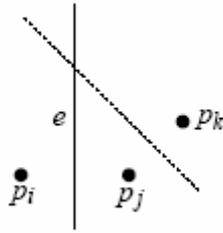


FIG. 21: Construção do Diagrama de Voronoi através dos bissetores.

Permanece a provar que $\text{Vor}(P)$ é conectado. Se isto não for o caso então haveria uma célula de Voronoi $C(p_i)$ dividindo o plano em dois. Como as células de Voronoi são convexas, $C(p_i)$ seria uma faixa limitada por duas retas paralelas. Contudo provou-se que bordas do Diagrama de Voronoi não podem ser retas, uma contradição.

Agora que entendemos a estrutura do Diagrama de Voronoi investigaremos sua complexidade, isto é, o número total de seus vértices e bordas. Desde que haja n “sites” e cada célula de Voronoi tem no máximo $n - 1$ vértices e bordas, a complexidade de $\text{Vor}(P)$ é no máximo quadrática. O teorema a seguir mostra que se pode construir simples células de Voronoi com complexidade linear, contudo não se pode afirmar que o Diagrama de Voronoi terá complexidade linear, mas sim que o número médio de vértices das células de Voronoi é ao menos seis.

Teorema 2: Para $n \geq 3$, o número de vértices no Diagrama de Voronoi de um conjunto de n “sites” no plano é no máximo $2n - 5$ e o número de bordas é no máximo $3n - 6$.

Demonstração:

Se os “sites” forem todos colineares então o teorema imediatamente segue do teorema 1, então assumindo que este não é caso, prova-se o teorema usando a fórmula de Euler, a qual diz que para qualquer grafo planar conectado com m_v nós, m_e arcos e m_f faces existe a seguinte relação:

$$m_v - m_e + m_f = 2$$

Não se pode aplicar diretamente a fórmula de Euler em $\text{Vor}(P)$, pois $\text{Vor}(P)$ bordas semi-infinitas e, portanto, não um grafo próprio. Para remediar a situação adicionar-se-á um vértice extra v_∞ “no infinito” ao conjunto de vértices e conectar-se-á todas as bordas semi-infinitas de $\text{Vor}(P)$ a este vértice. Agora tem-se um grafo planar conectado a qual aplica-se a fórmula de Euler. Obtém-se a seguinte relação entre n_v , o número de vértices de $\text{Vor}(P)$, n_e , o número de bordas de $\text{Vor}(P)$ e n , o número de “sites”: $(n_v + 1) - n_e + n = 2$. Além do mais, toda borda

em um grafo aumentado tem exatamente dois vértices, então se somarmos os graus de todos os vértices obter-se-á duas vezes o número de bordas, pois todo vértice, incluindo v_∞ , tem grau ao menos igual a três, logo obtém-se $2n_e \geq 3(n_v + 1)$.

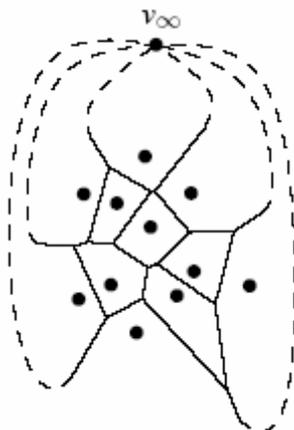


FIG. 22: Grafo planar conectado com vértice no infinito.

Para finalizar esta seção, discutiremos uma caracterização das bordas e vértices do Diagrama de Voronoi. Sabe-se que as bordas são partes dos bissetores dos pares de “sites” q e que os vértices são os pontos de interseção entre estes bissetores. Há um número quadrático de bissetores, visto que a complexidade de $\text{Vor}(P)$ é somente linear. Consequentemente, nem todos os bissetores definem bordas de $\text{Vor}(P)$ e nem todas as interseções são vértices de $\text{Vor}(P)$. Para caracterizar quais bissetores e interseções definem características do Diagrama de Voronoi cria-se a seguinte definição. Para um ponto q define-se o maior círculo de q em relação a P , representado por $F_P(q)$, como o maior círculo com centro em q que não contém qualquer “site” de P em seu interior. O seguinte teorema caracteriza os vértices e bordas do Diagrama de Voronoi.

Teorema 3: Para o Diagrama de Voronoi $\text{Vor}(P)$ de um conjunto de pontos P segue o seguinte:

- (i) Um ponto q é um vértice de $\text{Vor}(P)$ se e somente se seu maior círculo $F_P(q)$ contém três ou mais “sites” em sua circunferência;
- (ii) O bissetor entre os “sites” p_i e p_j definem uma borda de $\text{Vor}(P)$ se e somente se existir um ponto q sobre o bissetor tal que $F_P(q)$ contenha ambos p_i e p_j em sua circunferência, mas não outro “site”.

Demonstração:

(i) Suponha que haja um ponto q tal que $F_P(q)$ contenha três ou mais “sites” em sua circunferência. Seja p_i , p_j e p_k três destes “sites”. Desde que o interior de $F_P(q)$ esteja vazio, q deve estar na fronteira de cada $C(p_i)$, $C(p_j)$ e $C(p_k)$, e q deve ser um vértice de $\text{Vor}(P)$. Por outro lado, todo vértice q de $\text{Vor}(P)$ é incidente ao menos em três bordas e, conseqüentemente, em ao menos em três células de Voronoi $C(p_i)$, $C(p_j)$ e $C(p_k)$. O vértice q deve ser equidistante de p_i , p_j e p_k e não pode haver outro “site” próximo mais próximo a q , senão $C(p_i)$, $C(p_j)$ e $C(p_k)$ não se encontrariam em q . Conseqüentemente, o interior do círculo com p_i , p_j e p_k em sua circunferência não conteria qualquer “site”;

(ii) Suponha que exista um ponto q com a propriedade enunciada no teorema. Desde que $F_P(q)$ não contém qualquer “site” em seu interior e p_i e p_j estão em sua circunferência, tem-se $\text{dist}(q,p_i) = \text{dist}(q,p_j) \leq \text{dist}(q,p_k)$ para todo $1 \leq k \leq n$, segue-se que q pertence a uma borda ou vértice de $\text{Vor}(P)$. A primeira parte do teorema implica que q não pode ser um vértice de $\text{Vor}(P)$, logo q pertence a uma borda de $\text{Vor}(P)$, a qual é definida por um bissetor de p_i e p_j .

Inversamente, seja o bissetor de p_i e p_j definindo uma borda do Diagrama de Voronoi. O maior círculo vazio de qualquer ponto q no interior desta borda deve conter p_i e p_j em sua circunferência e não outros “sites”.

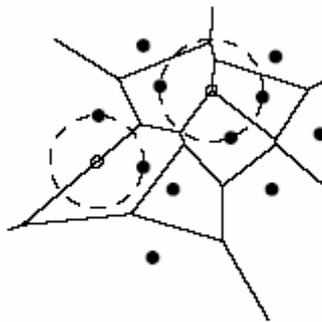


FIG. 23: Construção dos vértices e bordas do Diagrama de Voronoi, maiores círculos vazios.

15 ANEXO G - ANÁLISE DOS ERROS DE ODOMETRIA

Erros de odometria sistemáticos e não sistemáticos

As configurações do triciclo (veja figura 24), que emprega uma única roda dianteira de guia e duas rodas passivas, são razoavelmente comuns em aplicações de Veículos de Guiamento Automático (VGA) por causa de sua simplicidade inerente. Para a instrumentação de odometria na forma de um codificador do ângulo de direção, a solução de perda de contagem (falha na medida de deslocamento) é equivalente àquela de um veículo dirigido por Ackerman, onde a roda guiada substitui a roda central imaginária, ou seja, a direção Ackerman é projetada para assegurar que a roda frontal interna é rotacionada de uma leve variação angular em relação a roda mais externa quando realizando uma curva, portanto eliminando geometricamente o deslizamento induzido. Alternativamente, se a odometria diferencial do eixo traseiro for usada para determinar o sentido, a solução é utilizar equações geométricas simples para calcular a posição do veículo em relação a uma posição inicial conhecida. Um problema associado com a configuração do triciclo é que o centro de gravidade do veículo tende a mover-se afastando-se da roda dianteira ao atravessar uma inclinação, causando uma perda de tração. Como no caso de projetos dirigidos por Ackerman, alguns danos a superfície e erros de guiamento induzido são possíveis quando agindo na direção enquanto a plataforma não está movendo.

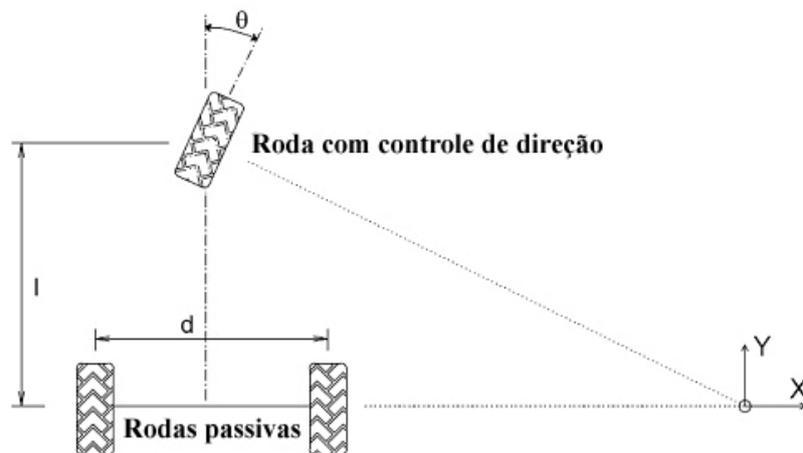


FIG. 24. A configuração do triciclo empregando uma roda guia e duas rodas passivas podem originar informações importantes diretamente de um codificador angular de direção ou indiretamente da odometria diferencial.

A odometria é o método de navegação mais extensamente usado para posicionamento de robôs móveis. Sabe-se bem que a odometria fornece uma boa exatidão a curto prazo, é barata, e permite taxas muito elevadas de amostragem. Entretanto, a idéia fundamental da odometria é a integração da informação do movimento incremental sobre o tempo, o qual conduz inevitavelmente à acumulação dos erros. Particularmente, acumulação dos erros de orientação causarão grandes erros de posição, os quais aumentam proporcionalmente com a distância percorrida pelo robô. Apesar destas limitações, muitos pesquisadores concordam que a odometria é uma parte importante de um sistema de navegação e que as tarefas de navegação serão simplificadas se a exatidão da odometria puder ser melhorada. A odometria é usada em quase todos os robôs móveis, por várias razões:

- Ø Os dados da odometria podem ser fundidos com medidas de posição absoluta para fornecer a melhor e mais confiável estimativa de posição [Cox, 1991; Byrne, 1992; Chenavier, 1992; Evans, 1994, Hollingum, 1991].
- Ø A odometria pode ser usada em entre as atualizações de posição absoluta com marcos. Dada uma exatidão de posicionamento requerida, o aumento da exatidão em odometria permite a menor frequência de atualização da posição absoluta. Em consequência, poucos marcos são necessários para uma dada distância de percurso.
- Ø Muitos algoritmos combinando mapeamento e demarcação (por exemplo: [Gonzalez, 1992; Chenavier e Crowley, já citados anteriormente]) assumem que o robô pode manter sua posição o suficiente para permitir que o robô veja os marcos em uma área limitada e combine características naquela área limitada para alcançar um tempo de processamento curto e melhorar a exatidão combinada [Cox, já citado anteriormente].
- Ø Em alguns casos, a odometria é a única informação de navegação disponível; por exemplo: quando nenhuma referência externa está disponível, quando circunstâncias impedem a colocação ou a seleção de marcos no ambiente, ou quando outro subsistema de sensores não fornece dados úteis.

Medida dos erros de odometria

Uma importante, mas raramente tratada dificuldade em robótica móvel é a *medida quantitativa* dos erros de odometria. A falta de procedimentos de medida bem definidos para a quantificação dos erros de odometria resulta na calibração pobre das plataformas móveis e relatos incomparáveis da exatidão odométrica em comunicações científicas. Para superar este problema Borenstein e Feng desenvolveram métodos para quantitativamente medir erros de

odometria sistemáticos, para um grau limitado, erros de odometria não-sistemáticos. Estes métodos confiam em um modelo simplificado de erro, em que dois dos erros sistemáticos são considerados como sendo dominante, a saber:

Ø o erro devido aos diâmetros desiguais da roda, definidos como

$$E_d = D_d / D_e \quad (1)$$

onde D_d e D_e são os diâmetros *reais* da roda direita e esquerda, respectivamente.

O erro devido à incerteza sobre a efetiva base da roda, definido como

$$E_b = b_{real} / b_{nominal} \quad (2)$$

onde b é a base da roda do veículo.

Medida de erros sistemáticos de odometria

O método descrito abaixo devido a Borenstein e Feng é intuitivo e largamente utilizado (e.g., [Komoriya, já citado anteriormente, Borenstein, 1987]), mas é um teste de desempenho fundamentalmente inadequado para robôs móveis com mudança de direção diferencial.

Teste do trajeto quadrado unidirecional – uma péssima medida para a exatidão odométrica

A figura 23 mostra um trajeto quadrado unidirecional de 4×4 m. O robô inicia em uma posição x_0, y_0, θ_0 o qual é rotulado INÍCIO. A área inicial deveria ser localizada próxima ao canto de duas paredes perpendiculares. As paredes servem como uma referência fixa antes e depois do funcionamento: medição de distância entre três pontos específicos no robô e as paredes permitem a exata determinação da posição e orientação absolutas do robô.

Para conduzir o teste, o robô deve ser programado para atravessar os quatro trechos do trajeto quadrado. O trajeto retornará o veículo à área inicial mas, por causa da odometria e erros do controlador, não precisamente na posição inicial. Desde que este teste visa determinar erros odométricos e não erros do controlador, o veículo não precisa ser programado para retornar precisamente a sua posição inicial -- retornando aproximadamente à área inicial é o suficiente. Ao concluir o trajeto quadrado, mede-se outra vez a posição

absoluta do veículo, usando as paredes fixas como uma referência. Estas medidas absolutas são então comparadas à posição e à orientação do veículo como calculada pelos dados da odometria. O resultado é um conjunto de *erros de posição de retorno* causado pela odometria e indicado por $\varepsilon x, \varepsilon y$ e $\varepsilon \theta$.

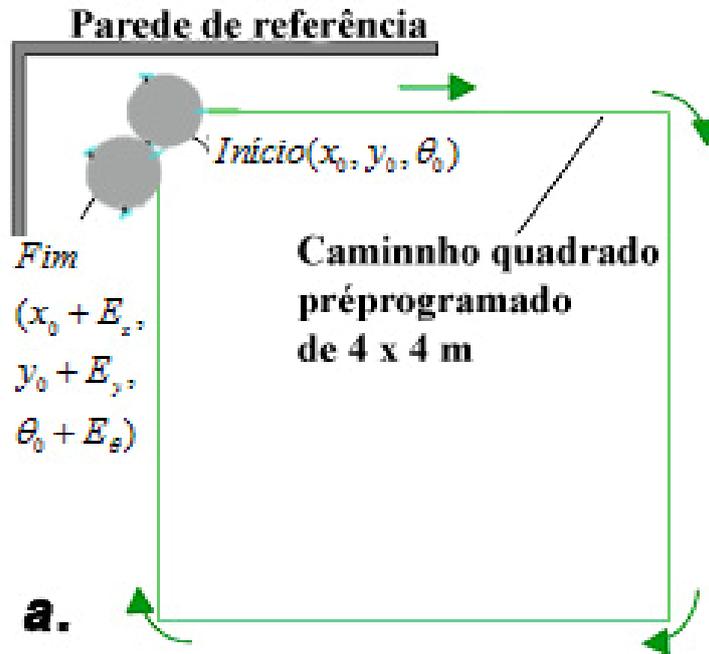


FIG. 25. Teste do trajeto quadrado

$$\begin{aligned} \varepsilon x &= x_{abs} - x_{calc} \\ \varepsilon y &= y_{abs} - y_{calc} \\ \varepsilon \theta &= \theta_{abs} - \theta_{calc} \end{aligned} \quad (3)$$

onde

$\varepsilon x, \varepsilon y, \varepsilon \theta$ = erros de posição e orientação devido a odometria;

posição e orientação absoluta do robô;

posição e orientação do robô calculada da odometria.

O trajeto mostrado na figura 23 compreende quatro segmentos em linha reta e as quatro rotações puras sobre o centro do robô, nos cantos do quadrado. A posição final do robô mostrada na figura 24 visualiza o erro de odometria.

Ao analisar os resultados deste experimento, pode-se extrair duas conclusões diferentes: O erro de odometria é o resultado do diâmetro desigual das rodas, E_d , como mostrado pelo trajetória ligeiramente curvada na figura 24 (linha pontilhada). Ou, o erro de odometria é o

resultado de incertezas sobre a base da roda, E_b . No exemplo da figura 24, E_b causou um giro do robô de 87° ao invés dos 90° desejados (trajetória tracejada na Figura 24).

Como se pode ver na figura 24, qualquer um destes dois casos *poderiam* alcançar aproximadamente o mesmo erro de posição. O fato que dois erros de mecanismos diferentes puderam resultar no mesmo erro de posição total podendo conduzir a um sério erro: corrigindo somente uma das duas fontes de erro em software. Este erro é tão sério porque alcançará aparentemente "excelentes" resultados, como mostrado no exemplo da figura 25. Neste exemplo, inicia-se "melhorando" o desempenho pelo ajuste da base da roda b no software de controle. De acordo com as equações "dead-reckoning" para veículos com mudança de direção diferencial, precisa-se somente aumentar o valor de b para fazer o robô girar mais em cada giro nominal de 90° . Em fazendo isso, logo ajustar-se-á b para um valor "ideal" conveniente que causará um giro do robô de 93° , portanto efetivamente compensando o erro de orientação de 3° introduzido por cada trecho levemente curvado (mas nominalmente reto) do trajeto quadrado.

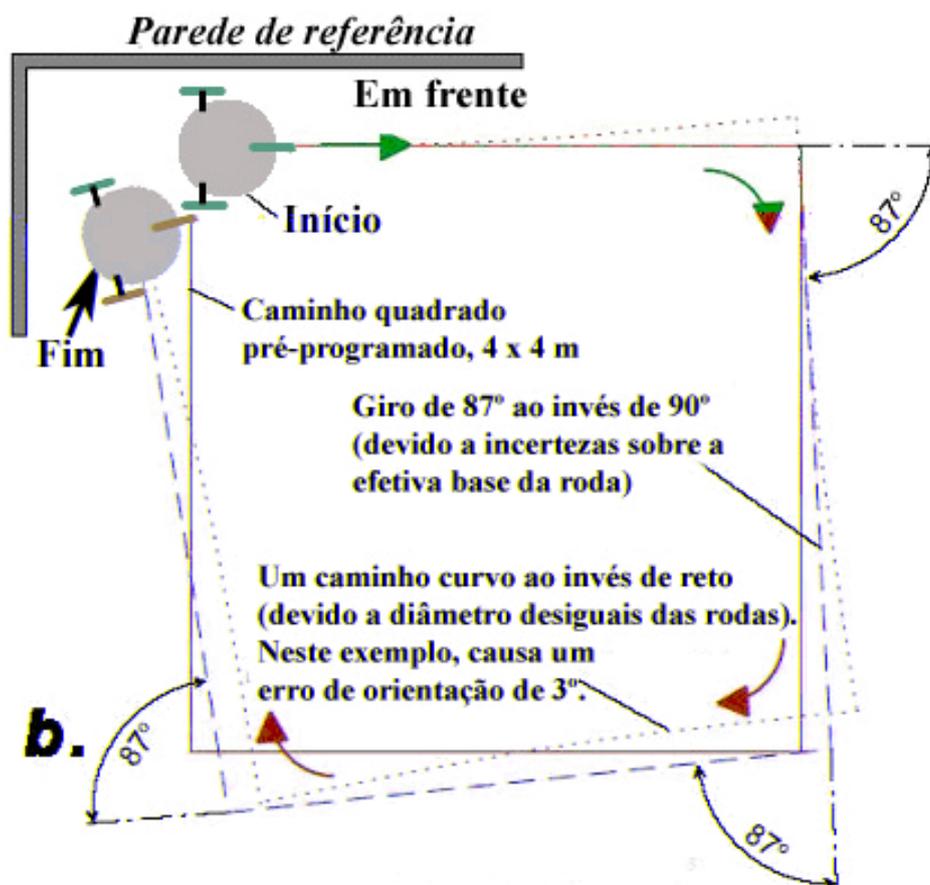


FIG. 26. Erros de odometria.

O experimento de caminho quadrado unidirecional.

- a. Caminho nominal
- b. Cada um dos dois erros significativos E_b ou E_d podem causar o mesmo erro de posição final.

O experimento do trajeto quadrado bidirecional

O exemplo detalhado da seção precedente ilustra que o experimento do trajeto quadrado unidirecional é inadequado para testar o desempenho da odometria em plataformas de guiamento diferencial, porque pode facilmente esconder dois erros de odometria que mutuamente compensam-se. Para super5 teee

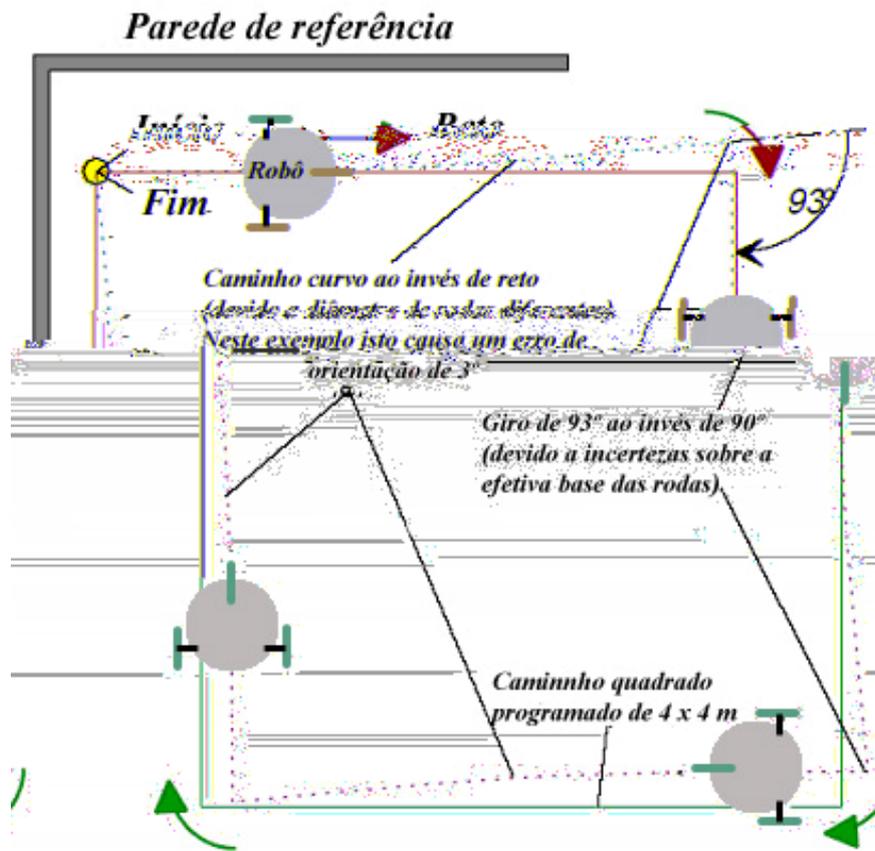


FIG. 27: O efeito de dois erros sistemáticos dominantes E_b e E_d . Observa-se que ambos os erros podem cancelar-se mutuamente quando o teste é desempenhado em uma única direção

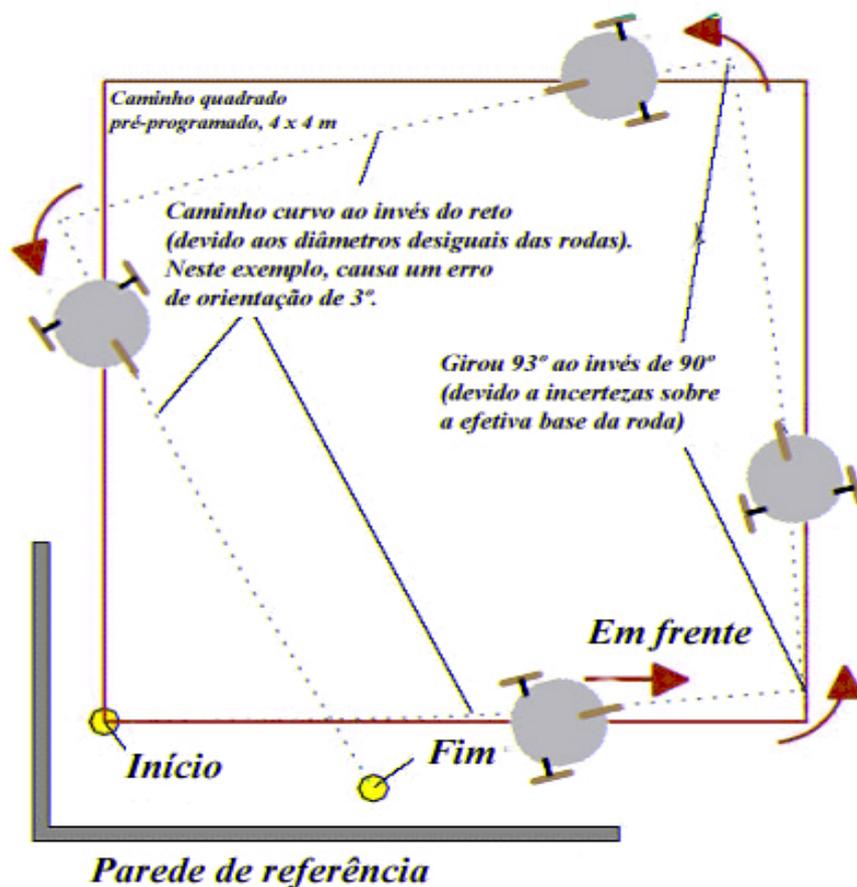


FIG. 28: O efeito de dois erros odométricos sistemáticos E_b e E_d : quando o trajeto quadrado é desempenhado na direção oposta pode-se observar que os erros crescem.

Ø As posições de parada depois de percorrido os sentidos horário e anti-horário são agrupados em duas áreas distintas.

Ø A distribuição nos agrupamentos horário e anti-horário são o resultado de erros não-sistemáticos, tais como mencionados na seção de erros sistemáticos e não sistemáticos. Entretanto, a figura 27 mostra que em um veículo não calibrado, percorrendo um assoalho de concreto razoavelmente liso, a contribuição dos erros *sistemáticos* ao erro total de odometria pode ser notavelmente maior do que a contribuição dos erros não-sistemáticos.

Após de conduzir o experimento UMBmark, pode-se desejar obter um único valor numérico que expressa a exatidão odométrica (em relação a erros sistemáticos) do veículo testado. A fim de minimizar os efeitos dos erros não-sistemáticos, sugeriu-se [Komoriya; Borenstein e Feng, já citados anteriormente] considerar o centro de gravidade de cada agrupamento como o representante para os erros de odometria sistemáticos nos sentidos horário e anti-horário.

As coordenadas dos dois centros de gravidade são calculadas do resultado da equação (3) como

$$\begin{aligned} x_{c.g.,h/ah} &= \frac{1}{n} \sum_{i=1}^n \varepsilon x_{i,h/ah} \\ y_{c.g.,h/ah} &= \frac{1}{n} \sum_{i=1}^n \varepsilon y_{i,h/ah} \end{aligned} \quad (4)$$

onde $n = 5$ é o número de funcionamentos em cada um sentido. Os desvios absolutos dos dois centros de gravidade da origem são nomeados $r_{c.g.,h}$ e $r_{c.g.,ah}$ (veja a figura 27) e são dados por

$$r_{c.g.,h} = \sqrt{(x_{c.g.,h})^2 + (y_{c.g.,h})^2} \quad (5a)$$

e

$$r_{c.g.,ah} = \sqrt{(x_{c.g.,ah})^2 + (y_{c.g.,ah})^2} \quad (5b)$$

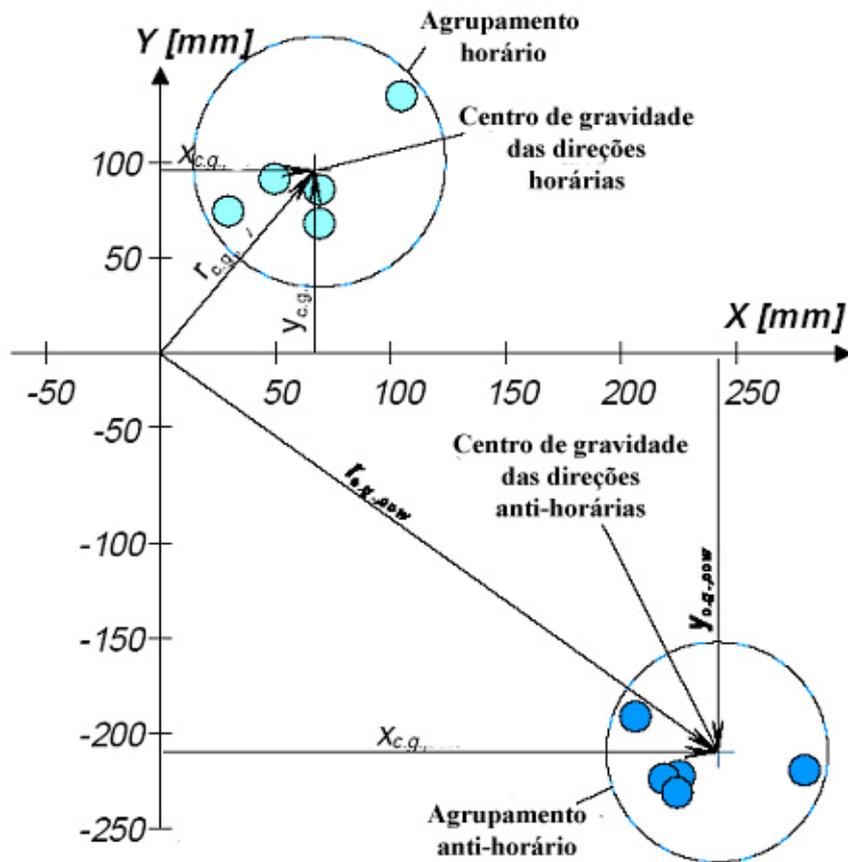


FIG. 29: Resultados típicos da execução do experimento UMBmark (um trajeto quadrado percorrido em ambos os sentidos horário e anti-horário) com um veículo descalibrado.

Finalmente, o maior valor entre $r_{c.g.,h}$ e $r_{c.g.,ah}$ é definido como "a medida da exatidão da odometria para erros sistemáticos":

$$E_{\max,sist} = \max(r_{c.g.,h}; r_{c.g.,ah}) \quad (6)$$

A razão para não se usar *média* dos dois centros de gravidade $r_{c.g.,h}$ e $r_{c.g.,ah}$ é que para aplicações práticas necessita-se preocupar-se com o maior erro odométrico possível. Deve-se também observar que o erro final de orientação $\varepsilon\theta$ não é considerado explicitamente na expressão para $E_{\max,sist}$. Isto é porque todos os erros sistemáticos de orientação são implicados pelos erros de posição final. Em outras palavras o trajeto quadrado tem os lados de comprimento fixo, erros sistemáticos de orientação traduzem diretamente em erros de posição.

Medida de erros não-sistemáticos

Alguma informação limitada sobre a suscetibilidade de um veículo aos erros não-sistemáticos podem ser derivados da propagação do retorno dos erros posição que foi mostrado em figura 29. Ao executar o procedimento UMBmark nos assoalhos lisos (e.g., um assoalho de concreto sem elevações visíveis ou rachaduras), uma indicação do valor dos erros não-sistemáticos podem ser obtido do cálculo do desvio padrão estimado). Entretanto, Borenstein e Feng cuidaram que houvesse somente valores limitados σ conhecidos, desde que σ reflita somente na interação entre o veículo e um determinado assoalho. Além disso, pode-se mostrar que comparando σ de dois robôs diferentes (até mesmo se viajarem no mesmo assoalho), um pode não necessariamente concluir que os robôs com grande σ mostrem maior suscetibilidade a erros não-sistemáticos.

Em aplicações reais é imperativo que *o maior distúrbio possível* seja determinado e usado em testes. Por exemplo, o desvio padrão estimado do teste na figura 29 não dá indicação alguma de tudo a respeito de qual erro deve esperar se uma roda do robô atravessar inadvertidamente uma grande elevação ou rachadura no assoalho. Pelas razões acima é difícil (talvez impossível) projetar geralmente um procedimento de teste quantitativo aplicável para erros não-sistemáticos. Entretanto, Borenstein propôs um teste facilmente reproduzível que permitisse comparar suscetibilidade aos erros não-sistemáticos de veículos diferentes. Este teste, chamado *UMBmark prolongado*, usa o mesmo trajeto quadrado bidirecional que o UMBmark, mas introduz elevações artificiais. As elevações artificiais são introduzidas por meio de um cabo elétrico (tal como esses que usam seis linhas de 15 A). Tal cabo tem um diâmetro de aproximadamente 9 a 10 milímetros. Sua forma arredondada e revestimento

plástico permitem até mesmo o menor robô atravessá-lo sem impacto físico demasiado. No teste UMBmark prolongado proposto o cabo é colocado 10 vezes sob uma das rodas do robô, durante o movimento. A fim fornecer a melhor repetibilidade para este teste e para evitar erros de compensação mutuamente, Borenstein e Feng sugerem que estas 10 elevações sejam introduzidas tão uniformemente quanto possível. As elevações devem também ser introduzidos durante o primeiro segmento reto do trajeto quadrado, e sempre sob a roda que está no interior do quadrado. Pode-se mostrar que o efeito mais visível de cada elevação é um erro fixo da orientação no sentido da roda que encontrou a elevação. No TRC *LabMate*, por exemplo, o erro de orientação resultante de uma elevação de altura $h = 10$ milímetros é aproximadamente $\Delta\theta = 0.44^\circ$.

Borenstein e Feng prosseguem a discussão que o parâmetro mensurável seja a maioria útil para expressar a suscetibilidade do veículo aos erros não-sistemáticos. Considere, por exemplo, o trajeto A e trajeto B na figura 30. Se as 10 elevações exigidas pelo teste *UMBmark prolongado* foram concentradas no começo da primeira reta (como mostrado no trajeto A), então o erro do retorno a posição seria muito pequeno. Inversamente, se as 10 elevações forem concentradas no fim da primeira reta (trajeto B na figura 30), então o erro de retorno a posição será maior.

Por causa desta sensibilidade aos erros de retorno a posição para a localização exata das elevações não é uma boa idéia usar o erro de retorno a posição como um indicador para a suscetibilidade de um robô aos erros não-sistemáticos. Ao invés disso, o erro de retorno a orientação $\varepsilon\theta$ deve ser usado. Embora seja mais difícil medir ângulos pequenos, a medida de $\varepsilon\theta$ é um indicador quantitativo mais consistente para comparar o desempenho de robôs diferentes. Assim, um pode medir e expressar a suscetibilidade de um veículo aos erros não-sistemáticos em termos do seu *erro de orientação absoluto médio* definido como

$$\varepsilon\theta_{med}^{n\acute{a}osis} = \frac{1}{n} \sum_{i=1}^n \left| \varepsilon\theta_{i,h}^{n\acute{a}osis} - \varepsilon\theta_{med,h}^{sis} \right| + \frac{1}{n} \sum_{i=1}^n \left| \varepsilon\theta_{i,ah}^{n\acute{a}osis} - \varepsilon\theta_{med,ah}^{sis} \right| \quad (7)$$

onde $n = 5$ é o número das experiências no sentido horário ou no sentido anti-horário, sobrescritos "*sistema*" e "*não sistema*" indicando um resultado obtido de qualquer teste UMBmark regular (para erros sistemáticos) ou do teste UMBmark prolongado (para erros não-sistemáticos). Observe que a equação (7) melhora a exatidão na identificação dos erros não-sistemáticos removendo o viés sistemático do veículo, dado por

$$\varepsilon\theta_{med,h}^{sis} = \frac{1}{n} \sum_{i=1}^n \varepsilon\theta_{i,h}^{sis} \quad (8a)$$

e

$$\varepsilon\theta_{med,ah}^{sis} = \frac{1}{n} \sum_{i=1}^n \varepsilon\theta_{i,ah}^{sis} \quad (8b)$$

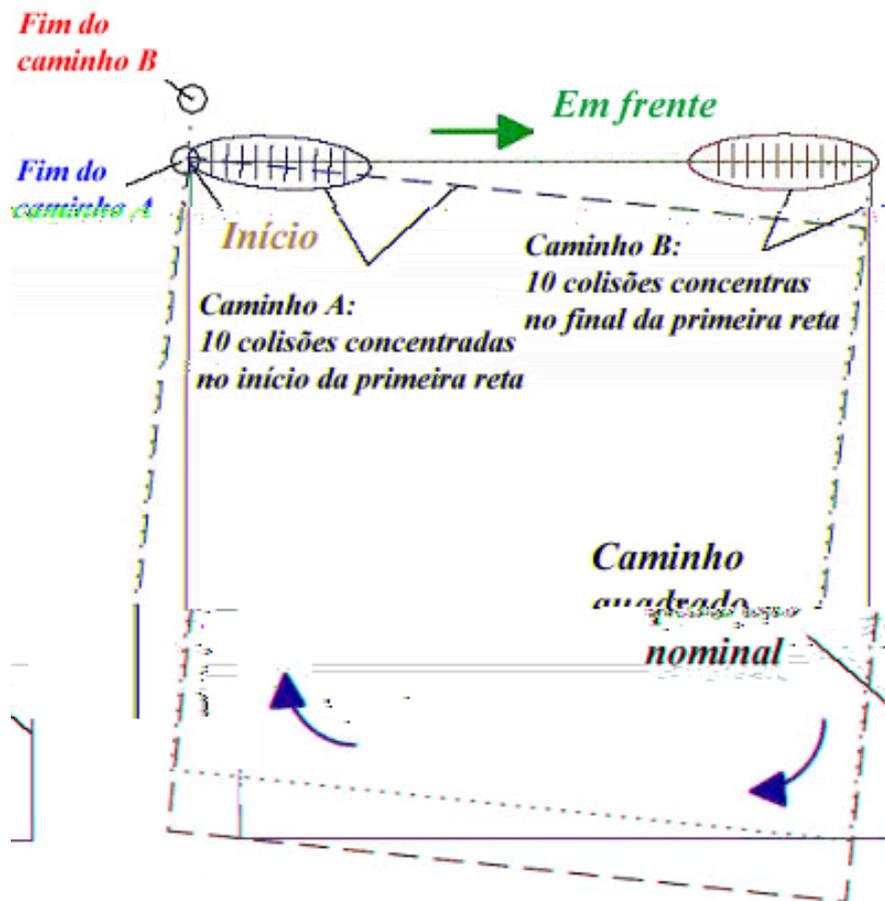


FIG. 30. Suscetibilidade aos erros não sistemáticos.

A posição de retorno do teste UMBmark prolongado é sensível a localização exata onde as 10 elevações ocorreram. A orientação de retorno não é.

Observe também que os argumentos nos sigmas na equação (7) são valores absolutos dos erros de orientação de retorno livre de viés. Isto é porque queria-se evitar o caso em que dois erros de orientação de retorno de sinais opostos cancelarem-se. Por exemplo, se um fosse $\varepsilon\theta = 1^\circ$ e na etapa seguinte $\varepsilon\theta = -1^\circ$, então não se deve concluir $\varepsilon\theta_{médio}^{nãosistema} = 0$. Usando o erro de retorno absoluto médio como calculado na equação (7) poderia calcular corretamente. Pelo contrário, na equação (8) a média aritmética real é calculada para identificar um viés fixo.

Redução dos erros de Odometria

A exatidão da odometria em plataformas móveis comerciais depende de algum grau em seu projeto cinemático e em determinadas dimensões críticas. Aqui são algumas das considerações específicas de projeto que afetam a exatidão da perda de contagem:

Os veículos com uma base das rodas pequena são mais sujeitos a erros de orientação do que veículos com uma base maior. Por exemplo, o robô de direção diferencial *LabMate* da TRC tem uma base relativamente pequena de 340 milímetros (13,4 polegadas). Em consequência, Gourley e Trivedi [Russel, 1995], sugerem que a odometria com o *LabMate* seja limitado a aproximadamente 10 metros (33 pés), antes um novo "reset" torna-se necessário.

- Ø Veículos com rodas do rodízio que carregam uma significativa porção do peso total são provavelmente induzidos ao deslizamento quando no sentido inverso (o “efeito do carro de supermercado”). Inversamente, se as rodas de rodízio carregarem somente uma pequena parcela do peso total, então o deslizamento não ocorrerá quando no sentido inverso [Borenstein e Koren, já citados anteriormente].
- Ø Sabe-se extensamente que, idealmente, as rodas usado para odometria devem ser em face estreita e não compressível. A roda ideal seria feita de alumínio com uma fina camada de borracha para melhor tração. Na prática, este projeto não é plausível para tudo mas, os mais leves veículos, porque a odometria das rodas são geralmente também rodas para carga, que exigem uma superfície de contato um tanto maior.
- Ø Tipicamente o projeto com direção síncrona fornece melhor exatidão odométrica do que veículos com direção diferencial. Isto é especialmente verdadeiro ao viajar sobre assoalho irregular: as irregularidades arbitrárias afetarão somente uma roda de cada vez. Assim, desde que outras duas rodas permaneçam em contato com o solo, elas fornecem mais tração e força a roda afetada pelo deslizamento. Conseqüentemente, a distância total viajada será refletida corretamente pela quantidade indicada pela odometria.

Outras as tentativas em melhorar a exatidão odométrica são baseadas em modelos mais detalhados. Por exemplo, Larsson et al. usou segmentos circulares para substituir os segmentos lineares em cada período de amostragem. Os benefícios desta abordagem são relativamente pequenas. Boyden e Velinsky compararam (nas simulações) técnicas odométricas convencionais, baseadas somente na cinemática, às soluções baseadas na dinâmica do veículo. Apresentaram resultados da simulação para mostrar que para ambos os robôs móveis usando guiamento diferencial, o modelo cinemático era exato somente em

velocidades mais lentas, até 0.3 m/s quando realizando uma curva fechada. Este resultado concorda com as observações experimentais, as quais sugerem que erros devido ao deslizamento da roda podem ser reduzidos em algum grau pela limitação da velocidade do veículo durante o giro, e pela limitação das acelerações.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)