

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

Proposta de Simulador Virtual para Sistema de
Navegação de Robôs Móveis Utilizando
Conceitos de Prototipagem Rápida

Autor: Leonimer Flávio de Melo
Orientador: Prof. Dr. João Maurício Rosário

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
DEPARTAMENTO DE PROJETO MECÂNICO

Proposta de Simulador Virtual para Sistema de Navegação de Robôs Móveis Utilizando Conceitos de Prototipagem Rápida

Autor: **Leonimer Flávio de Melo**

Orientador: Prof. Dr. João Maurício Rosário

Curso: Engenharia Mecânica

Área de Concentração: Projeto Mecânico

Tese de doutorado apresentada à comissão de Pós Graduação da Faculdade de Engenharia Mecânica, como requisito para a obtenção do título de Doutor em Engenharia Mecânica

Campinas, 2007

S.P. – Brasil

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

M49p Melo, Leonimer Flávio de
 Proposta de simulador virtual para sistema de navegação de
 robôs móveis utilizando conceitos de prototipagem rápida /
 Leonimer Flávio de Melo.–Campinas, SP: [s.n.], 2007.

 Orientador: João Maurício Rosário
 Tese (Doutorado) - Universidade Estadual de Campinas,
 Faculdade de Engenharia Mecânica.

 1. Robótica. 2. Simulação (Computadores). 3. Robôs
 móveis. 4. Robôs - Sistema de controle. I. Rosário, João
 Maurício. II. Universidade Estadual de Campinas. Faculdade
 de Engenharia Mecânica. III. Título.

Título em Inglês: Virtual Simulator Propose for Mobile Robots Navigation Systems
 Using Rapid Prototyping Concepts

Palavras-chave em Inglês: Mobile robotic systems, Embedded control systems,
 Reconfigurable control architecture, Open architecture
 systems, rapid control prototyping

Área de concentração: Mecânica dos Sólidos e projeto Mecânico

Titulação: Doutor em Engenharia Mecânica

Banca Examinadora: Humberto Ferasoli Filho, Edson Roberto De Pieri, Niederauer
 Mastelari e Helder Aníbal Hermini

Data da defesa: 22/11/2007

Programa de Pós-Graduação: Engenharia Mecânica

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
DEPARTAMENTO DE PROJETO MECÂNICO

TESE DE DOUTORADO

**Proposta de Simulador Virtual para Sistema
de Navegação de Robôs Móveis Utilizando
Conceitos de Prototipagem Rápida**

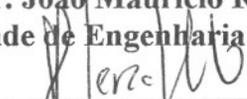
Autor: Leonimer Flávio de Melo

Orientador: Prof. Dr. João Maurício Rosário

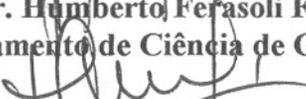
A Banca Examinadora composta pelos membros abaixo aprovou esta Tese:



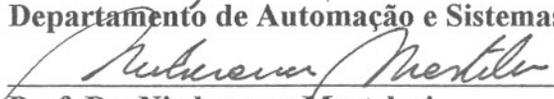
Prof. Dr. João Maurício Rosário, Presidente
Faculdade de Engenharia Mecânica, UNICAMP



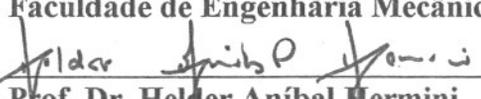
Prof. Dr. Humberto Fefasoli Filho
Departamento de Ciência de Computação, UNESP-Bauru



Prof. Dr. Edson Roberto De Pieri
Departamento de Automação e Sistemas, UFSC



Prof. Dr. Niederauer Mastelari
Faculdade de Engenharia Mecânica, UNICAMP



Prof. Dr. Helder Aníbal Mermini
Faculdade de Engenharia Mecânica, UNICAMP

Campinas, 22 de novembro de 2007

200805381

Dedicatória:

Em memória de meu pai, Antonio de Oliveira Melo, pelo exemplo que foi e pelo nome que sempre honrarei...

“Não sonhe, realize!”

Agradecimentos

Obrigado meu Deus por ter me dado mais esta oportunidade... Obrigado Jesus por estar sempre comigo, me incentivando nas horas de desânimo, me mostrando o Caminho nas tentativas de desvios de rota... Obrigado Maria pelo regaço sagrado e reconfortante onde chagas d'alma foram curadas com prantos...

Agradeço ao meu orientador, Prof. Dr. João Maurício Rosário, pela sua compreensão, apoio e dedicação.

Agradeço ao Prof. Dr. Humberto Ferasoli Filho por ter cedido o ambiente robótico “Aedromo” para validação experimental de parte deste trabalho, ao colega José Guilherme pela colaboração preciosa nas programações e coleta de resultados junto aos robôs deste ambiente e aos colegas e companheiros do LAIR - Laboratório de Automação Integrada e Robótica, da Faculdade de Engenharia Mecânica da UNICAMP, em especial o amigo Miro, pelo apoio, ajuda e lições proporcionadas, junto a um ambiente de camaradagem e cooperação.

Agradeço: a minha querida mãezinha, Chams Antonio Melo, que sempre me auxiliou nas horas difíceis e esteve comigo em todos os momentos de minha vida. A minha esposa amada (e eterna namorada), Xú, por ter me apoiado e incentivado na conclusão deste trabalho, sem sua ajuda nunca teria conseguido... Aos meus filhos adorados, Rodrigo, Nick e Tata, por todas as alegrias e felicidades que já me proporcionaram. Aos manos diletos, Naid, Tinhinha, Neno e Meméia, pela companhia, amor, compreensão e companheirismo.

Agradeço, enfim, a todos parentes, amigos, colegas e companheiros, que de uma maneira ou outra, diretamente ou indiretamente, me ajudaram na conclusão deste trabalho. Sem vocês todos, que o espaço aqui se faz pequeno para relacioná-los, eu nunca teria chegado ao fim desta odisséia!

Resumo

MELO, Leonimer Flávio, *Proposta de Simulador Virtual para Sistema de Navegação de Robôs Móveis Utilizando Conceitos de Prototipagem Rápida*, Campinas,: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2007. 298 p. Tese (Doutorado)

Este trabalho apresenta a proposta de implementação de um ambiente virtual de simulação para projeto e concepção de sistema de supervisão e controle para robôs móveis, que sejam capazes de operar e de se adaptar a diferentes ambientes e condições. Este sistema virtual tem como finalidade facilitar o desenvolvimento de protótipos de sistemas embarcados, enfatizando a implementação de ferramentas que permitam a simulação das condições cinemáticas, dinâmicas e de controle, com monitoração em tempo real de todos pontos importantes do sistema. Para isso, é proposta uma arquitetura aberta de controle, integrando as duas principais técnicas de implementação de controle robótico a nível de hardware: sistemas microprocessadores e dispositivos de hardware reconfiguráveis. O sistema simulador implementado é composto de um módulo gerador de trajetória, de um módulo simulador cinemático e dinâmico e de um módulo de análise de resultados e erros. O módulo gerador de trajetória tem a finalidade de, uma vez conhecendo-se o ambiente em que o robô irá atuar, com seus obstáculos e particularidades, gerar uma trajetória cartesiana ótima, respeitando os limites e características do robô móvel. Todos os resultados cinemáticos e dinâmicos colhidos durante a simulação podem ser avaliados e visualizados em formatos de gráficos e tabelas, no módulo de análise de resultados, permitindo que seja feito um aperfeiçoamento no sistema, no sentido de minimizar os erros com a otimização dos ajustes necessários. Para a implementação do controlador no sistema embarcado utiliza-se a prototipagem rápida, que é a tecnologia que permite, em conjunto com o ambiente virtual de simulação, o desenvolvimento de um projeto de um controlador para robôs móveis. A validação e testes foram realizados com modelos de robôs móveis não holonômicos de transmissão diferencial.

Palavras-chave

Sistemas Robóticos Móveis, Sistemas de Controle Embarcado, Arquitetura de Controle Reconfigurável, Sistemas de Arquitetura Aberta, Prototipagem de Controle Rápida

Abstract

MELO, Leonimer Flávio, *Virtual Simulator Propose for Mobile Robots Navigation Systems Using Rapid Prototyping Concepts*, Campinas,: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2007. 298 p. Tese (Doutorado)

This work presents the proposal of virtual environment implementation for project simulation and conception of supervision and control systems for mobile robots, that are capable to operate and adapting in different environments and conditions. This virtual system has as purpose to facilitate the development of embedded architecture systems, emphasizing the implementation of tools that allow the simulation of the kinematic conditions, dynamic and control, with real time monitoring of all important system points. For this, an open control architecture is proposal, integrating the two main techniques of robotic control implementation in the hardware level: systems microprocessors and reconfigurable hardware devices. The implemented simulator system is composed of a trajectory generating module, a kinematic and dynamic simulator module and of a analysis module of results and errors. The kinematic and dynamic simulator module makes all simulation of the mobile robot following the pre-determined trajectory of the trajectory generator. All the kinematic and dynamic results shown during the simulation can be evaluated and visualized in graphs and tables formats, in the results analysis module, allowing an improvement in the system, minimizing the errors with the necessary adjustments optimization. For controller implementation in the embedded system, it uses the rapid prototyping, that is the technology that allows, in set with the virtual simulation environment, the development of a controller project for mobile robots. The validation and tests had been accomplish with nonholonomics mobile robots models with diferencial transmission.

Key words

Mobile Robotic Systems, Embedded Control Systems, Reconfigurable Control Architecture, Open Architecture Systems, Rapid Control Prototyping

Sumário

Lista de Figuras	xvi
Lista de Tabelas	xxiii
Lista de Símbolos	xxiv
Lista de Acrônimos	xxvii
1 Introdução	1
1.1 Breve histórico da robótica móvel	2
1.2 Motivação	3
1.3 Prototipagem rápida	4
1.4 Sistemas Embarcados	4
1.5 Objetivos	6
1.5.1 Objetivos específicos	6
1.6 Descrição dos capítulos	7
2 Modelagem cinemática de robôs móveis	9
2.1 Modelos cinemáticos utilizando-se referência local e global	10
2.1.1 Representação da posição do robô móvel	11
2.1.2 Modelos de cinemática direta	13

2.2	Modelo cinemático de robôs móveis através do CCI	15
2.3	Sistema de transmissão diferencial	18
2.3.1	Cinemática direta para transmissão diferencial	21
2.3.2	Cinemática dos robôs móveis não holonômicos	25
2.4	Transmissão síncrona em robôs móveis	26
2.4.1	Cinemática direta para transmissão síncrona	28
2.4.2	Cinemática inversa para transmissão síncrona	28
2.5	Modelos cinemáticos de robôs móveis com rodas direcionáveis	29
2.5.1	Bicicletas	29
2.5.2	Triciclos	29
2.5.2.1	Cinemática direta para robôs móveis com rodas direcionáveis	30
2.5.2.2	Cinemática inversa para robôs móveis com rodas direcionáveis	32
2.5.3	Sistema automotivo de direção	32
2.5.3.1	Cinemática direta para sistemas de direção de Ackerman . .	33
2.6	Considerações finais	34
3	Proposta de arquitetura aberta de supervisão e controle embarcada	35
3.1	Sistemas reconfiguráveis	35
3.2	Processamento digital de sinais	38
3.3	Proposta do sistema embarcado	40
3.3.1	Descrição dos níveis de controle	43
3.3.2	Comunicação remota	45
3.3.3	Descrição da estrutura hierárquica	46
3.3.4	Detalhes da implementação do hardware	49
3.4	Considerações finais	54
4	Implementação de um simulador de movimento com ênfase na dinâmica e	

controle	55
4.1 Modelagem matemática	55
4.1.1 Modelo cinemático	57
4.1.2 Modelagem dinâmica do sistema	57
4.1.2.1 Modelo do atuador	57
4.2 Controlador PID	60
4.2.1 Ajustes de parâmetros para controladores PID	60
4.2.1.1 Primeiro método de Ziegler-Nichols para sintonia de controladores PID	62
4.2.1.2 Segundo método de Ziegler-Nichols para sintonia de controladores PID	63
4.2.2 Determinação das constantes do controlador PID	65
4.3 Análise dinâmica do sistema	65
4.3.1 Análise de resposta transitória e de regime estacionário	67
4.3.2 Análise do lugar das raízes	68
4.3.3 Análise de resposta em frequência	69
4.3.3.1 Digramas de Bode	70
4.3.3.2 Diagramas de Nyquist	71
4.3.3.3 Carta de Nichols	72
4.4 Considerações finais	73
5 Implementação do simulador virtual	74
5.1 Visão geral do sistema	75
5.2 Bloco gerador de trajetória	77
5.2.1 Considerações matemáticas da implementação do gerador de trajetória	79
5.2.2 Gráficos do gerador de trajetória	80
5.3 O sistema simulador virtual	81

5.3.1	Módulo geração de movimentos	85
5.3.2	Módulo do controlador e modelo dinâmico	88
5.3.3	Módulo cinemático	89
5.4	Analisador gráfico de resultados	91
5.5	Prototipagem rápida	94
5.6	Prototipagem rápida nos sistemas robóticos móveis	96
5.6.1	Descrição da plataforma	98
5.6.2	Simulação <i>hardware-in-the-loop</i>	99
5.6.3	Prototipagem de controle rápida	101
5.7	Considerações finais	102
6	Validação experimental	104
6.1	Analisador gráfico de resultados da simulação	104
6.2	Primeiro estudo de caso	105
6.2.1	Velocidade linear do robô	105
6.2.2	Deslocamento linear nos eixos cartesianos	107
6.2.3	Trajetória cartesiana do robô	108
6.2.4	Deslocamento angular do robô	108
6.2.5	Velocidade angular do robô	109
6.2.6	Raio de curvatura do robô	109
6.2.7	Velocidade linear nas rodas	110
6.2.8	Estudo dos erros	112
6.2.9	Conclusões	112
6.3	Segundo estudo de caso	113
6.3.1	Velocidade linear do robô	113
6.3.2	Deslocamento angular do robô	116

6.3.3	Velocidade angular do robô	116
6.3.4	Raio de curvatura do robô	118
6.3.5	Velocidade linear nas rodas	118
6.3.6	Velocidade angular nos eixos	119
6.3.7	Análise dos erros	119
6.3.8	Conclusões	120
6.4	Validação experimental cinemática	123
6.4.1	O ambiente “Aedromo”	123
6.4.2	Elementos constituintes do Aedromo	125
6.4.2.1	Sistema de visão	125
6.4.2.2	Robôs móveis	126
6.4.2.3	A arena	127
6.4.2.4	O sistema de comunicação	127
6.4.3	Descrição material dos elementos do Aedromo	128
6.4.4	Implementação do supervisor de trajetórias	130
6.4.5	Conclusões	133
6.5	Considerações finais	133
7	Conclusões e Sugestões para Próximos Trabalhos	135
	Referências	139
	Apêndice A – Sensores nos Sistemas Robóticos Móveis	150
A.1	Conceitos básicos sobre sensores	150
A.2	Modelamento de sensores	153
A.3	Sensores de contato	156
A.4	Sensores inerciais	157

A.4.1	Acelerômetros	157
A.4.2	Giroscópios	158
A.4.3	Inclinômetros e bússolas	160
A.4.3.1	Inclinômetros	160
A.4.3.2	Bússolas magnéticas	161
A.4.3.3	Giro-bússolas	161
A.4.3.4	Bússolas de fluxo de porta	162
A.5	Sensores infravermelho	163
A.6	Sonar	165
A.6.1	Modelamento de transdutores ultra-sônicos	169
A.6.2	Interpretação dos dados do sonar	171
A.7	Radar	172
A.8	Laser	173
A.8.0.1	Telêmetro a laser	174
A.9	Sistema de posicionamento baseado em satélite	176
A.9.1	Descrição técnica	177
A.9.1.1	Trilateração do Satélite	177
A.9.1.2	Alcance do Satélite	178
A.9.1.3	Precisão de tempo	178
A.9.1.4	Posicionamento do Satélite	178
A.9.1.5	Correção de erros	179
A.9.1.6	Precisão dos serviços GPS	179
A.9.2	Aplicação do GPS na robótica móvel	180
	Apêndice B – Sistemas de navegação	187
B.1	Técnicas de medição de distância	188

B.1.1	Medições no deslocamento de fase	188
B.1.2	Medições no ToF de um sinal	192
B.1.3	Medições baseadas em radares e lasers	192
B.2	Planejamento de trajetória	192
B.2.1	Espaço de configuração	192
B.2.1.1	Simplificações do espaço-C	195
B.2.2	Planejamento de trajetória nos robôs móveis não holonômicos	196
B.3	Sistemas de posicionamento dos robôs móveis	198
B.3.1	Cálculo preditivo por inferência	201
B.3.2	Medições com pontos de referência	201
B.3.3	Triangulação	203
B.3.4	Triangulação com incerteza	207
B.4	Fusão de dados sensoriais	210
B.4.1	Modelos de espaço de estado	211
B.4.2	Modelo de planta	212
B.4.3	Modelo de medição	213
B.4.4	Filtro de Kalman	216
B.4.4.1	Combinando-se medições e estimativa	217
B.4.5	Filtro de Kalman estendido	220
B.4.6	Utilizando o Filtro de Kalman na localização de robôs móveis	222
B.5	Mapas	225
B.5.1	Mapas sensoriais	227
B.5.1.1	Mapas baseados em imagem	227
B.5.1.2	Modelos de Markov	228
B.5.2	Mapas geométricos	230

B.5.3	Mapas topológicos	231
B.6	Considerações finais	233
Apêndice C – Sistemas de transmissão digital		234
C.1	Codificação para transmissão digital	234
C.1.1	Codificação NRZ	234
C.1.2	Codificação RZ	234
C.1.3	Codificações bifásicas	235
C.2	Modulação em telecomunicações	237
C.2.1	Modulação em amplitude	238
C.2.1.1	Modulação AM-ASK	238
Apêndice D – Proposta de implementação de um sistema de navegação autônoma		242
D.1	Protocolo de comunicação	242
D.2	O quadro de comunicação	244
D.2.1	O link de RF	244
D.2.2	O módulo de transmissão-recepção	244
Anexo A – Listagem da rotina de geração de trajetória		249
Anexo B – Listagem da rotina de geração de movimentos		257
Anexo C – Listagem da rotina do modelo cinemático do robô		259
Anexo D – Listagem da rotina de inicialização dos atuadores do robô		261
Anexo E – Publicações		264

Lista de Figuras

2.1	O plano de referência global e a referência local do robô.	11
2.2	O robô móvel está alinhado com os eixos do plano de referência global.	13
2.3	Robô móvel com tração diferencial no plano global de referência.	14
2.4	Sistema de locomoção ideal com roda	16
2.5	Centro de curvatura instantânea (CCI)	17
2.6	Ilustração de uma roda castor	18
2.7	Ilustração do protótipo do robô móvel experimental.	19
2.8	A cinemática da transmissão diferencial num robô móvel	20
2.9	Geometria da cinemática direta num robô com transmissão diferencial.	21
2.10	Cinemática direta num robô com transmissão diferencial em $t \rightarrow t + \delta t$	23
2.11	Robô Nomad 200 fabricado pela Nomadic Technologies. Utiliza transmissão síncrona para locomoção.	27
2.12	Cinemática de robôs com transmissão síncrona.	28
2.13	Localização do CCI para uma bicicleta.	30
2.14	Cinemática de um robô triciclo.	31
2.15	Cinemática de um robô como o mecanismo de Ackerman.	33
3.1	Proposta de arquitetura aberta global para sistema robótico móvel.	41
3.2	Diagrama de blocos ilustrando os níveis de controle para o sistema robótico móvel.	44

3.3	Diagrama de blocos mostrando a interligação da camada de controle supervisor e a camada de controle embarcado através de um link de rádio frequência. . .	46
3.4	Ilustração das comunicações do módulo supervisor e do comando manual. . .	47
3.5	Diferentes níveis de controle do sistema proposto.	48
3.6	Representação das estruturas de níveis de controle do robô móvel.	49
3.7	Ilustração da arquitetura de hardware do sistema proposto.	50
3.8	Módulo de RF BiM-433 usado para implementação da comunicação remota.	52
3.9	Implementação física da arquitetura proposta no robô móvel.	53
3.10	Blocos de controle do motor DC do robô móvel.	54
4.1	Estrutura de controle do robô móvel.	56
4.2	Esquema equivalente do motor CC com acionamento da carga.	58
4.3	Modelamento do motor CC.	58
4.4	Exemplo da rotina de inicialização dos parâmetros dos atuadores.	61
4.5	Controlador PID do motor CC.	61
4.6	Controlador PID de uma planta genérica.	61
4.7	Curva de resposta ao degrau.	62
4.8	Oscilação sustentada com período P_{cr}	64
4.9	Análise dinâmica do sistema no Simulink [®]	66
4.10	Resposta ao degrau.	67
4.11	Gráficos de ω_m e θ_L em resposta ao degrau.	68
4.12	Pólos e zeros do sistema.	69
4.13	Gráfico do lugar das raízes do sistema.	70
4.14	Análise de resposta em frequência com o módulo em dB e o ângulo de fase. .	71
4.15	Diagrama de Nyquist.	72
4.16	Carta de Nichols.	73
5.1	Uma visão geral do sistema.	75

5.2	Exemplo de um ambiente com obstáculos para navegação do robô móvel. . .	76
5.3	Implementação do bloco gerador de trajetória no MatLab [®]	78
5.4	Pontos iniciais fornecidos ao gerador de trajetória.	81
5.5	Ensaio do traçado da trajetória.	82
5.6	Traçado final da trajetória do robô móvel.	82
5.7	Diagrama de Blocos representando os módulos de geração de movimentos, controlador, modelo cinemático e modelo dinâmico.	83
5.8	Janela inicial do simulador do sistema robótico móvel.	84
5.9	Exemplo de um sinal utilizado como entrada para velocidade linear do robô móvel.	85
5.10	Simulador do sistema robótico móvel implementado no Simulink [®]	86
5.11	Bloco gerador de movimentos.	87
5.12	Módulo do controlador e modelo dinâmico da roda direita.	89
5.13	Controlador PID dos atuadores do robô móvel.	89
5.14	Implementação do modelo cinemático do robô móvel.	90
5.15	Menu principal do analisador gráfico de resultados do sistema robótico. . . .	91
5.16	Sub-menus do sistema analisador gráfico.	92
5.17	Sub-menus do sistema analisador gráfico.	93
5.18	Sub-menus do sistema analisador gráfico.	93
5.19	Submenu do módulo de estudos de erros do robô com um exemplo de gráfico gerado.	94
5.20	Plataforma do Robô móvel com os principais elementos.	98
5.21	Protótipo do robô móvel experimental. LAIR - Laboratório de Automação Integrada e Robótica - FEM - UNICAMP.	99
5.22	Simulação HIL do sistema robótico móvel.	100
5.23	Sistema de prototipagem de controle rápida (RCP).	102
6.1	Trajétoria de referência e dinâmica do robô móvel. Primeiro estudo de caso.	105

6.2	Gráficos da velocidade linear de referência e velocidade linear dinâmica do robô e do erro.	106
6.3	Gráficos do deslocamento linear de referência e dinâmico nos eixos cartesianos, com seus respectivos erros.	107
6.4	Trajetória de referência e dinâmica do robô móvel com o erro apresentado. .	108
6.5	Gráficos do deslocamento angular de referência e dinâmico do robô com o respectivo erro.	109
6.6	Gráficos da velocidade angular de referência e dinâmica do robô e o gráfico do respectivo erro.	110
6.7	Gráficos do raio de curvatura de referência e dinâmica do robô e do respectivo erro.	111
6.8	Gráfico da velocidade linear na roda esquerda e do erro.	111
6.9	Gráficos da análise dos erros nas velocidades do robô.	112
6.10	Trajetória de referência e dinâmica do robô móvel. Segundo estudo de caso. .	113
6.11	Sinal de referência da velocidade linear do robô.	114
6.12	Gráficos da velocidade linear do robô e do erro respectivo.	115
6.13	Erro de trajetória causada por uma simulação de velocidade linear acima da máxima permitida.	115
6.14	Gráficos da velocidade linear do robô e do erro correspondente.	116
6.15	Gráficos do deslocamento angular de referência e dinâmico do robô com o respectivo erro.	117
6.16	Gráficos da velocidade angular de referência e dinâmica do robô e do respectivo erro.	117
6.17	Gráficos do raio de curvatura de referência e dinâmica do robô e do respectivo erro.	118
6.18	Velocidade linear nas rodas direita e esquerda com os respectivos erros dinâmicos.	119
6.19	Velocidade angular nas rodas direita e esquerda com os respectivos erros dinâmicos.	120

6.20	Submenu do módulo de estudos de erros do robô com um exemplo de gráfico gerado.	121
6.21	Submenu do módulo de estudos de erros dos eixos com um exemplo de gráfico gerado.	122
6.22	O ambiente Aedromo.	124
6.23	Diagrama em blocos da eletrônica embarcada dos robôs.	127
6.24	Diagrama de blocos do sistema de comunicação e transmissão de sinal. . . .	128
6.25	Arena - Plano cartesiano utilizado para navegação do robô.	129
6.26	Sistemas robóticos do Aedromo.	129
6.27	A posição e a direção dos robôs no plano cartesiano.	130
6.28	Detalhes do robô utilizado.	130
6.29	Trajetória a ser executada pelo robô.	131
6.30	Execução da trajetória proposta pelo robô móvel.	132
6.31	Trajetórias de referência e executada pelo robô.	132
6.32	Execução de uma trajetória retangular pelo robô móvel em estudo.	133
A.1	Exemplo de giroscópio mecânico.	158
A.2	Exemplos de inclinômetros comerciais.	160
A.3	Exemplos de giro-bússolas.	161
A.4	Esquema elétrico de uma bússola de fluxo de porta.	162
A.5	Bússola de fluxo de porta em suspensão fluídica.	163
A.6	Exemplo de sensores infravermelho nos robôs móveis.	164
A.7	Ilustração do princípio de funcionamento do <i>sonar</i>	165
A.8	Exemplo de um sistema sonar.	167
A.9	Amplitude em dB do sinal sonar em função da orientação com relação a normal do transdutor.	168
A.10	Interação do sonar com obstáculos.	181

A.11 Pulso do sonar e a função de propagação pontual.	182
A.12 Convolução da resposta de impulso do transdutor e a atenuação do sinal em função do ângulo do transdutor.	183
A.13 Exemplo das medidas do sonar feitas pelo robô móvel.	184
A.14 Exemplo da região de distância constante (RDC) vista pelo sonar.	185
A.15 Erro de alinhamento nas varreduras a laser feitas em B' e em B	185
A.16 Resultado do alinhamento das duas varreduras dos sensores laser.	186
B.1 Relação entre a onda gerada e a refletida, onde x é a medida correspondente à diferença de fase ϕ	189
B.2 Circuito simples para detecção da diferença de fase entre o sinal transmitido (V_1) e o recebido (V_2).	190
B.3 Efeito da dilatação pelo raio r do robô.	196
B.4 Exemplo de triangulação simples. O robô móvel está na posição \mathbf{x}_1	205
B.5 Estimativa da localização do robô móvel baseado na triangulação com marcadores.	206
B.6 Estimativa da localização do robô móvel baseado na triangulação com incerteza.	208
B.7 DGP para dois marcadores posicionados em $(0,0)$ e $(1,0)$	209
B.8 Deslocamento do vetor posicionamento $\mathbf{x}(k)$	216
B.9 O filtro de Kalman utilizado na localização de robôs móveis.	224
B.10 O filtro de Kalman utilizado na estimativa de localização de robôs móveis.	224
B.11 Simulação utilizando o filtro de Kalman na estimativa de localização de robôs móveis.	225
B.12 Representação baseada em gráfico de um ambiente do robô móvel.	232
C.1 Exemplo das codificações NRZ e RZ.	235
C.2 Exemplo das codificações bifásicas.	236
C.3 Exemplo da codificações bifásica de <i>Miller</i>	236
C.4 Exemplo da codificações bipolar AMI.	237

C.5	Modulação em amplitude.	239
C.6	Modulação BASK.	240
C.7	Modulação MASK.	240
C.8	Modulação OOK.	241
D.1	Códigos do protocolo de comunicação.	243
D.2	Quadro do protocolo de comunicação.	244
D.3	Comunicação entre o robô móvel e os marcadores.	245
D.4	Módulo de controle embarcado de transmissão-recepção.	245
D.5	Módulo de controle embarcado para transmissão-recepção do sinal de RF.	246

Lista de Tabelas

4.1	Parâmetros dos motores CC.	59
4.2	Regra de sintonia do controlador PID baseada na resposta ao degrau.	63
4.3	Regra de sintonia do controlador PID baseada no ganho crítico.	64
D.1	Formação lógica dos códigos.	242
D.2	Formação lógica dos códigos.	243

Lista de Símbolos

X_I, Y_I	- Plano de referência global
X_R, Y_R	- Plano de referência local
P	- Ponto de referência no chassi do robô
θ	- Diferença angular entre os planos de referência global e local
ξ_I	- Base do posicionamento do robô no plano de referência global
$R(\theta)$	- Matriz rotacional ortogonal do robô
$\dot{\varphi}_1$	- Velocidade de rotação da roda 1 do robô
v_d	- Velocidade linear da roda direita com referência ao plano local do robô
v_e	- Velocidade linear da roda esquerda com referência ao plano local do robô
v	- Velocidade linear do robô
\mathbf{p}	- Vetor posicionamento do robô no plano
L	- Distância dos centros das duas rodas ao longo do eixo das rodas do robô
R	- Distância do CCI até o ponto central entre as duas rodas do robô
ω	- Velocidade angular do robô com relação ao CCI
α	- Direcionamento da roda dianteira do robô
ϕ	- Deslocamento de fase de um sinal refletido
d	- Distância do robô móvel até o alvo
λ	- Comprimento de onda do sinal modulado
f	- Freqüência de modulação da onda emitida
T	- Intervalo de amostragem
A	- Fator de amplitude do ganho do amplificador integrador
R_a	- Intervalo de ambigüidade
c	- Velocidade da luz no vácuo
\mathcal{C}	- Possibilidades de configurações do robô no espaço
$G(\mathbf{p})$	- Restrições holonômicas do robô
\mathcal{C}_{livre}	- A parte do ambiente que é acessível ao robô
δ_i	- Vetor intervalo de posicionamento do robô
P_n	- Posição do marcador n no plano cartesiano

d_n	- Distância do robô ao marcador n no plano cartesiano
h	- Distância entre dois marcadores
\mathcal{J}	- Jacobiano das equações de medições
x_i	- Medição sensorial i do robô móvel
\hat{x}	- Estimativa combinada das medições sensoriais
$\hat{\sigma}$	- Variância ou, para medições vetoriais, a matriz covariância
\mathbf{z}	- Vetor de medição
$\hat{\mathbf{x}}(k k_1)$	- Estimativa de estado a ser computada
$\tilde{\mathbf{x}}(k)$	- Predição do vetor de estado $\mathbf{x}(k)$ antes do tempo k
$\hat{\mathbf{x}}_k(-)$	- Opção à notação $\tilde{\mathbf{x}}(k)$
$\hat{\mathbf{x}}(k)$	- Estimativa do vetor de estado $\mathbf{x}(k)$ no tempo k , ou $\hat{\mathbf{x}}_k(+)$
$\mathbf{u}(t)$	- Descrição das entradas de controle permitidas
$\mathbf{x}(k)$	- Modelo de planta que descreve o estado do sistema
$\mathbf{u}(k)$	- Entrada de controle do robô
$\mathbf{v}(k)$	- Ruído da medição
$\Psi()$	- Função transição de estado
$N(0, \mathbf{Q}(k))$	- Modelo de ruído gaussiano de média zero com covariância $\mathbf{Q}(k)$
Ψ	- Matriz que expressa a evolução do sistema
Υ	- Matriz que expressa como as entradas do controle modificam o estado do sistema
$\mathbf{z}_i(k)$	- Modelo de medição
$\mathbf{w}_i(k)$	- Função ruído
$N(0, \mathbf{R}_i(k))$	- Modelo de ruído gaussiano de média zero com a matriz covariância $\mathbf{R}_i(k)$
\mathbf{p}_t	- Vetor de estado de alvo
$\mathbf{h}_{st}()$	- Função de medição
$\mathbf{x}(k)$	- Localização do veículo
Λ_E	- Matriz que expressa como as medições são derivadas com uma transformação linear do estado
$\mathbf{Q}(k)$	- Matriz covariância
$E[v_i]$	- Ruído de sistema de média zero
$\sigma_i(k)$	- Covariância de ruído
$\mathbf{P}()$	- Matriz covariância de estado
$\nabla\Psi()$	- Jacobiano do modelo de planta
$\nabla\mathbf{h}_{st}()$	- Jacobiano do modelo de medição
I_i	- Conjunto de medições
$\mathcal{B}el(p)$	- Probabilidade que o robô tenha um certo posicionamento
$P_\alpha(p p')$	- Probabilidade de uma ação α tomada a partir do estado p' até o estado p
$P(s)$	- Probabilidade intrínseca da leitura s
\mathcal{H}	- Entropia de $\mathcal{B}el(p)$
$E_\alpha[\mathcal{H}]$	- Entropia futura esperada

$P(o \mathbf{p}, m)$	- Probabilidade do robô observar o no posicionamento \mathbf{p} e no mapa m
\mathbf{X}_{ref}	- Vetor posicionamento
V_{ref}	- Velocidade linear de referência do robô
R_{curv}	- Raio de trajetória instantâneo do robô
VE_{ref}	- Velocidade linear da roda esquerda
VD_{ref}	- Velocidade linear da roda direita
$e_a(t)$	- Tensão de alimentação da armadura do motor
$T_m(t)$	- Torque do motor
$\theta_m(t)$	- Posição angular do rotor do motor
$i(t)$	- Corrente do motor
L_a	- Indutância da armadura do motor
R_a	- Resistência elétrica da armadura do motor
B_m	- Coeficiente de atrito viscoso
K_b	- Constante de força contra-eletromotriz
K_t	- Constante de torque
J_m	- Inércia do eixo do motor
N	- Razão da engrenagem do redutor
B_{eq}	- Coeficiente de atrito viscoso equivalente do sistema motor-carga
J_{eq}	- Inércia equivalente
K_p	- Constante proporcional do PID
T_i	- Constante integral do PID
T_d	- Constante derivativa do PID
K_{cr}	- Valor crítico de K_p
P_{cr}	- Período crítico
ω_m	- Velocidade angular do motor
θ_L	- Deslocamento angular da carga
$G(j\omega)$	- Diagrama polar de uma função de transferência senoidal
dB	- decibel
\mathbf{TJ}_{ref}	- Matriz trajetória de referência do robô
\mathbf{TJ}_{din}	- Matriz trajetória dinâmica do robô móvel
\mathbf{V}_{din}	- Vetor velocidade dinâmica do robô móvel
\mathbf{R}_c	- Vetor raio de curvatura dinâmico do robô móvel
ω_r	- Velocidade angular das rodas do robô móvel
ω_e	- Velocidade angular do eixo do motor
θ_r	- Ângulo de deslocamento instantâneo das rodas dos robô móvel
\mathbf{x}_{din}	- Vetor de trajetória dinâmica do robô móvel no eixo x
\mathbf{y}_{din}	- Vetor de trajetória dinâmica do robô móvel no eixo y

Lista de Acrônimos

CAD	-	<i>Computer-Aided Design</i>
CCI	-	Centro de Curvatura Instantânea
CRI	-	Centro de Rotação Instantânea
ToF	-	<i>Time-of-Flight</i>
RF	-	Rádio Freqüência
FM	-	Freqüência Modulada
LORAN	-	<i>L</i> ong <i>R</i> ange <i>N</i> avigation
GPS	-	<i>G</i> lobal <i>P</i> ositioning <i>S</i> ystem
DGP	-	Diluição Geométrica da Precisão
GDOP	-	<i>G</i> eometric <i>D</i> ilution of <i>P</i> recision
PDOP	-	<i>P</i> ositional <i>D</i> ilution of <i>P</i> recision
DGHP	-	Diluição Geométrica Horizontal da Precisão
HDOP	-	<i>H</i> orizontal <i>G</i> eometric <i>D</i> ilution of <i>P</i> recision
DGPP	-	Diluição Geométrica Posicional da Precisão
PDOP	-	<i>P</i> ositional <i>G</i> eometric <i>D</i> ilution of <i>P</i> recision
RMTR	-	Robôs Móveis Tracionados por Rodas
WMRs	-	<i>w</i> heeled <i>m</i> obile <i>r</i> obots
FKE	-	Filtro de Kalman Estendido
SLAM	-	<i>S</i> imultaneous <i>L</i> ocalization and <i>M</i> apping
ROM	-	<i>R</i> ead- <i>O</i> nly <i>M</i> emory
FPGA	-	<i>F</i> ield <i>P</i> rogrammable <i>G</i> ate <i>A</i> rray
CCM	-	<i>C</i> ustom <i>C</i> omputing <i>M</i> achines
FCCM	-	<i>F</i> PGA-based <i>C</i> ustom <i>C</i> omputing <i>M</i> achines
ASICs	-	<i>A</i> pplication <i>S</i> pecific <i>I</i> ntegrated <i>C</i> ircuits
IP-CORE	-	<i>I</i> ntellectual <i>P</i> roperty <i>C</i> ore
PCI	-	<i>P</i> eripheral <i>C</i> omponent <i>I</i> nterconnect
FFT	-	<i>F</i> ast <i>F</i> ourier <i>T</i> ransform
PLD	-	<i>P</i> rogrammable <i>L</i> ogic <i>D</i> evice
EPLD	-	<i>E</i> rasable <i>P</i> rogrammable <i>L</i> ogic <i>D</i> evices

CPLD	-	<i>Complex Programmable Logic Devices</i>
DFGA	-	<i>Dinamically Field Programmable Gate Array</i>
DSP	-	<i>Digital Signal Processor</i>
VHDL	-	<i>VHSIC Hardware Description Language</i>
VHSIC	-	<i>Very-High-Speed Integrated Circuit</i>
AHDL	-	<i>Altera Hardware Description Language</i>
MPYA	-	<i>Multiply and Accumulate Previous Product</i>
DTMF	-	<i>Dual Tone MultiFrequential</i>
MFC	-	<i>Multifrequencial Compelida</i>
RTDX	-	<i>Real Time Data Exchange</i>
HIL	-	<i>Hardware In-the-Loop</i>
USB	-	<i>Universal Serial Bus</i>
PCMCIA	-	<i>Personal Computer Memory Card International Association</i>
UHF	-	<i>Ultra High Frequency</i>
PWM	-	<i>Pulse-Width Modulation</i>
PID	-	<i>Proporcional-Integral-Derivativo</i>
ANSI	-	<i>American National Standards Institute</i>
WLAN	-	<i>Wireless Local Area Network</i>
IEEE	-	<i>Institute of Electrical and Electronics Engineers</i>
TCP/IP	-	<i>Transmission Control Protocol / Internet Protocol</i>
GUI	-	<i>Graphical User Interface</i>
RAM	-	<i>Random Access Memory</i>
DC	-	<i>Direct Current</i>
CC	-	<i>Corrente Contínua</i>
CAD	-	<i>Computer-Aided Design</i>
CAM	-	<i>Computer-Aided Manufacturing</i>
CNC	-	<i>Controle Numérico por Computador ou Computer Numeric Control</i>
ERSP	-	<i>Evolution Robotics Software Platform</i>
ERSA	-	<i>Evolution Robotics Software Architecture</i>
ECU	-	<i>Electronic controle unit</i>
RCP	-	<i>Rapid Control Prototyping</i>
FIRA	-	<i>Federation of International Robot-soccer Association</i>
IA	-	<i>Inteligência Artificial</i>
RPM	-	<i> rotações por Minuto</i>
VDC	-	<i>Volts of Direct Current</i>
IR	-	<i>Infra-Red</i>

Capítulo 1

Introdução

A robótica móvel é uma área de pesquisa que trata com o controle de veículos autônomos ou semi-autônomos. O que diferencia a robótica móvel de outras áreas de pesquisa em robótica tais como a robótica de manipuladores, é a sua ênfase nos problemas relacionados com a operação (locomoção) em ambientes complexos de larga escala, que se modificam dinamicamente, compostos tanto de obstáculos estáticos como de obstáculos móveis. Para operar neste tipo de ambiente o robô deve ser capaz de adquirir e utilizar conhecimento sobre o ambiente, estimar uma posição dentro deste ambiente, possuir a habilidade de reconhecer obstáculos, e responder em tempo real as situações que possam ocorrer neste ambiente. Além disso, todas estas funcionalidades devem operar em conjunto. As tarefas de perceber o ambiente, se localizar no ambiente, e se mover pelo ambiente são problemas fundamentais no estudo dos robôs móveis autônomos (THIELSCHER, 2005).

O estudo dos robôs móveis é uma área de pesquisa predominantemente interdisciplinar envolvendo as seguintes tecnologias (DUDEK; JENKIN, 2000):

- Engenharia Mecânica: Projetos de veículos e dos sistemas mecânicos envolvidos, particularmente os mecanismos de locomoção.
- Ciência da Computação: Desenvolvimento de programas, simuladores, algoritmos para modelamento, sensoramento e planejamento.
- Engenharia Elétrica: Integração dos sistemas, sensores e comunicação.
- Psicologia Cognitiva, Percepção e Neurociência: Discernimento de como sistemas biológicos resolvem problemas similares.

Sistemas robóticos móveis são cometimentos da engenharia. Os atuadores, processadores,

interfaces com o usuário, sensores e mecanismos de comunicação que permitem que o robô móvel opere devem estar integrados de tal maneira a permitir que o sistema inteiro funcione em sincronismo como um todo.

1.1 Breve histórico da robótica móvel

A robótica móvel vem gradativamente se desenvolvendo há muitos anos. Desde a década de 50 pesquisadores já se interessavam no desenvolvimento de robôs móveis. Walter (1950) construiu diversos robôs móveis em 1950 que eram capazes de executar tarefas tais como desviar de obstáculos e seguir fontes luminosas, utilizando capacitores para controlar o robô.

Em Stanford, Nilsson (1969) desenvolveu o robô móvel SHAKEY em 1969. Este robô utilizava dois motores de passo em uma configuração diferencial (cinemática diferencial) para se locomover e era equipado com sensores de distância, câmeras de vídeo e sensores táteis. Ele era conectado a dois computadores por links de rádio e de vídeo. O robô SHAKEY utilizava programas para percepção, modelagem, e atuação no ambiente. As tarefas desempenhadas pelo robô incluíam desviar de obstáculos e a movimentação de blocos coloridos. O robô móvel tinha grandes dificuldades em processar e interpretar as informações sensoriais obtidas do ambiente e nunca foi capaz de completar uma seqüência completa de ações em um ambiente real.

Novamente em Stanford, Moravec (1990) desenvolveu o robô móvel CART no final da década de 70. A tarefa do robô era desviar de obstáculos utilizando uma câmera de vídeo. O robô móvel CART conseguia desviar dos obstáculos com sucesso, mas era muito lento e tinha problemas em se localizar no ambiente necessitando de uma iluminação propícia para perceber os obstáculos adequadamente. Também final da década de 70 foi desenvolvido o robô móvel HILARE em Toulouse, França (BRIOT, 1979). Este foi um dos primeiros projetos de robô móvel desenvolvido na Europa. HILARE utilizava câmeras de vídeo, sensores de distância a laser e ultra-som para navegar no ambiente. O planejamento de trajetória era executado utilizando-se uma representação poligonal do ambiente. Os sensores ultra-sônicos eram utilizados para evitar os obstáculos próximos. O sistema de visão era utilizado para detectar obstáculos distantes e tinha a limitação de ser muito lento.

Todos estes exemplos reforçam a idéia de que controlar robôs móveis é uma tarefa complexa que desafia os pesquisadores das diversas áreas tecnológicas, tais como Inteligência Artificial, Computação, Sistemas de Controle, etc., envolvendo um trabalho mecatrônico com as engenharias mecânica, elétrica e da computação.

1.2 Motivação

O desenvolvimento de sistemas de controle para robôs móveis autônomos tem se mostrado um grande desafio para a os pesquisadores até os dias atuais. Diferentes abordagens para o projeto de sistema de controle para robôs móveis autônomos vem sendo utilizadas em diversas áreas de pesquisa. Por muitos anos os pesquisadores têm construído sistemas de controle que apresentam um comportamento inteligente em ambientes controlados, com situações ideais, mas que normalmente não mantém o mesmo desempenho no mundo real. Existem inúmeros sistemas de controle para serem utilizados no mundo real, mas geralmente estes sistemas são limitados e não apresentam um comportamento autônomo ou inteligente.

Existem diversas aplicações possíveis para os robôs móveis. No transporte, vigilância, inspeção, limpeza de casas, exploração espacial, auxílio a deficientes físicos, entre outros. No entanto, os robôs móveis autônomos ainda não causaram muito impacto em aplicações domésticas ou industriais, principalmente devido a falta de um sistema de controle robusto, confiável e flexível que permitiria que estes robôs operassem em ambientes dinâmicos, pouco estruturados, e habitados por seres humanos. O desenvolvimento de um modelo de um sistema robótico móvel com arquitetura de controle aberta e flexível, possuindo um sistema de controle robusto, que incorpore o que existe de mais moderno em termos de tecnologia de *hardware* embarcada e que possibilite a operação de um robô móvel em um ambiente do mundo real é uma das motivações desta tese de doutorado.

O planejamento da locomoção, sob vários tipos de restrições, é um campo muito vasto de pesquisas na área da robótica móvel (GRAF; WANDOSELL; SCHAEFFER, 2001). O planejamento básico de trajetória para os robôs móveis refere-se a determinação de um caminho no espaço-C (*espaço de configuração*) entre uma configuração inicial do robô e uma configuração final, de tal forma que o robô não colida com nenhum obstáculo no ambiente, e que a movimentação planejada seja consistente com as restrições cinemáticas do veículo. Neste contexto, um dos pontos abordados neste trabalho foi o desenvolvimento de um planejador de trajetória para robôs móveis.

Uma das principais motivações deste trabalho é propiciar um ambiente virtual que facilite o desenvolvimento de protótipos de sistemas embarcados, enfatizando a implementação de ferramentas que permitam a simulação das condições cinemáticas, dinâmicas e de controle, com monitoração em tempo real de todos pontos importantes do sistema. Desta monta, a proposta de um simulador virtual de sistemas robóticos móveis é apresentada juntamente com técnicas de prototipagem rápida.

1.3 Prototipagem rápida

O termo prototipagem rápida, na sua versão tradicional da engenharia mecânica, designa um conjunto de tecnologias usadas para se fabricar objetos físicos diretamente a partir de fontes de dados gerados por sistemas de projeto auxiliado por computador (CAD). Tais métodos são bastante peculiares, uma vez que eles agregam e ligam materiais, camada a camada, de forma a constituir o objeto desejado. Eles oferecem diversas vantagens em muitas aplicações quando comparados aos processos de fabricação clássicos baseados em remoção de material, tais como fresamento ou torneamento (WOHLERS, 2001).

Tais métodos permitem aos projetistas criar rapidamente protótipos concretos a partir de seus projetos, ao invés de figuras bidimensionais. Esses modelos apresentam diversos usos. Eles constituem um auxílio visual excelente durante a discussão prévia do projeto com colaboradores ou clientes. Além disso, o protótipo pode permitir testes prévios como, por exemplo, ensaios em túnel de vento para componentes aeronáuticos ou análise fotoelástica para se verificar pontos de concentração de tensões na peça. A verdade é que os projetistas sempre construíram protótipos; os processos de prototipagem rápida permitem que eles sejam feitos mais depressa e de forma mais barata. De fato, estima-se que a economia de tempo e de custos proporcionada pela aplicação das técnicas de prototipagem rápida na construção de modelos sejam da ordem de 70 a 90% (GORNI, 2001).

No contexto deste trabalho, a prototipagem rápida é a tecnologia que permite, em conjunto com um ambiente virtual de simulação, o desenvolvimento de um projeto de um controlador para robôs móveis. Depois de testado e validado no simulador, o sistema de controle é programado na memória da placa de controle do robô móvel. Desta maneira, uma economia de tempo e material é conseguida, validando primeiramente todo o modelo virtualmente para depois se operar a implementação física do sistema.

1.4 Sistemas Embarcados

O desenvolvimento de protótipos de sistemas embarcados tem sido alvo de inúmeros trabalhos na área de automação e controle. Impressoras de computadores, automóveis, robôs industriais, robôs móveis, aparelhos de telefonia, cadeiras de rodas, próteses antropomórficas, e inúmeros outros dispositivos que congregam sensores e atuadores para execução de alguma tarefa e que exijam maior ou menor volume de controle, são sistemas embarcados ou que utilizam sistemas embarcados. Usualmente são usados micro-controladores com algoritmos

desenvolvidos em várias linguagens de software para realizar o controle destes dispositivos. alguns destes algoritmos foram também desenvolvidos em trabalhos associados com computação reconfigurável, com vantagens em termos de desempenho (ITO; CARRO, 2000).

Entre todos os campos associados com projeto de sistemas embarcados, as tecnologias de software e hardware têm sofrido rápida evolução. É muito grande a quantidade de novos microprocessadores, interfaces de comunicação, interfaces de potência, sensores, compiladores, sistemas operacionais e sistemas de desenvolvimento fornecidos aos engenheiros a cada ano. Em função desta acelerada evolução tecnológica, a idéia de utilizar-se estruturas abertas e reconfiguráveis, que possam adaptar-se a novas demandas, torna-se muito atraente, sendo mesmo um pré-requisito na consideração de um projeto de sistema embarcado.

O conceito de sistemas abertos tem sido estudado nos últimos anos por várias instituições no campo de máquinas-ferramenta e engenharia de produção, focalizando os aspectos de modularidade, os efeitos da arquitetura de controle e da rede de comunicação na performance do sistema. O objetivo deste conceito aplicado à arquitetura reconfigurável é permitir uma fácil e rápida adaptação de dispositivos embarcados a novas evoluções tecnológicas, para uma melhor portabilidade e capacidade de intercâmbio para o sistema final. A divisão da estrutura em pequenos blocos funcionais, com interfaces bem especificadas, permite uma melhor especificação das tarefas de uma equipe de projeto multidisciplinar, bem com a rápida adaptação de um determinado bloco a uma nova evolução tecnológica. Este tipo de solução não pode ser alcançada em muitos dos sistemas comerciais disponíveis, pois estes não fornecem informações ou não apresentam recursos para permitir este tipo de manipulação.

Dentre os sistemas embarcados, os robôs móveis têm-se apresentado como plataformas para consolidação de conhecimento em diversas áreas de pesquisa, como modelagem, controle, automação, sistemas de potência, sensores, transmissão de dados, eletrônica embarcada e engenharia de software, sendo cada vez mais usadas em instituições de ensino e pesquisa.

Através da proposta de uma arquitetura hierárquica e aberta, distribuindo as diversas ações de controle em níveis crescentes de complexidade e da utilização de recursos de computação reconfigurável, propõe-se a validação deste ambiente num simulador virtual para robôs móveis.

1.5 Objetivos

O principal objetivo deste trabalho é apresentar o desenvolvimento de um ambiente virtual de simulação para o projeto de um sistema de controle para robôs móveis, utilizando-se conceitos de prototipagem rápida, e que seja capaz de operar e de se adaptar a diferentes ambientes e condições, e para isso é proposta uma arquitetura aberta de supervisão e controle.

1.5.1 Objetivos específicos

Estudar o modelamento cinemático, dinâmico e de controle dos sistemas robóticos móveis e propor um modelo a ser utilizado.

Estudar os sistemas de navegação utilizados em robôs móveis, avaliando suas capacidades e limitações.

Propor uma arquitetura aberta de supervisão e controle que integre as melhores características das técnicas que existem atualmente.

Implementar um simulador virtual que será utilizado para validar o sistema proposto, preparando o ambiente para a prototipagem rápida, sendo composto dos seguintes módulos:

- Módulo gerador de trajetória. Tem a finalidade de, uma vez conhecendo-se o ambiente em que o robô irá atuar, com seus obstáculos e particularidades, gerar uma trajetória cartesiana ótima, respeitando os limites e características do robô móvel.
- Módulo simulador cinemático e dinâmico. Faz toda simulação cinemática e dinâmica do robô móvel em tempo real, rastreando a trajetória pré-determinada no gerador de trajetória.
- Módulo de análise de resultados e erros. Todos os resultados cinemáticos e dinâmicos colhidos durante a simulação são visualizados e avaliados em formatos de gráficos e tabelas, permitindo que sejam verificados pormenores a respeito de cada item em avaliação e sejam feitas otimizações no sistema no sentido de minimizar os erros e otimizar os processos.

1.6 Descrição dos capítulos

Devido a complexidade do assunto e a diversidade de temas abordados neste trabalho, optou-se por apresentar revisões bibliográficas inseridas em cada capítulo pertinente ao assunto tratado. Por conseguinte, deixamos de apresentar um capítulo único com toda revisão bibliográfica, introduzindo-a paulatinamente dentro dos capítulos, tornando-se de maneira mais abrangente nos capítulos 2 e 3.

Uma revisão bibliográfica sobre os modelos cinemáticos de sistemas robóticos móveis é apresentada no capítulo 2, com os conceitos de modelos cinemáticos genéricos e enfatizando-se os aspectos matemáticos de alguns sistemas robóticos mais relevantes para este trabalho. No capítulo 3 é feita uma revisão bibliográfica a respeito dos sistemas reconfiguráveis e do processamento digital de sinais e, em seguida, apresentado a proposta de arquitetura aberta de supervisão e controle, onde são mostrados os objetivos e propriedades do sistema, enfatizando-se aspectos de implementação de *hardware*. No capítulo 4 é descrita a implementação dos modelos cinemático, dinâmico e do controlador. É feita um análise dinâmica do sistema descrevendo o tipo do controlador adotado e o modelamento dos atuadores. No Capítulo 5 é apresentado o ambiente de simulação virtual com a descrição do bloco gerador de trajetória, dos módulos de geração de movimentos, módulo do controlador e modelo dinâmico, e módulo cinemático. É apresentado também neste capítulo o analisador gráfico de resultados. Na seção 5.5 deste mesmo capítulo é feita uma revisão bibliográfica a respeito de prototipagem rápida e são apresentadas as tecnologias de prototipagem rápida a serem utilizadas no sistema proposto. O capítulo 6 apresenta validação experimental do trabalho, apresentando resultados experimentais com exemplos de simulações, envolvendo diferentes situações de navegação do sistema robótico em estudo. O capítulo 7 apresenta as conclusões finais deste trabalho com sugestões para trabalhos futuros. No apêndice A são apresentados detalhes sobre sensores nos sistemas robóticos móveis. No apêndice B é apresentado uma revisão bibliográfica a respeito dos sistemas de navegação, enfatizando-se as técnicas utilizadas em robôs móveis. São abordados neste capítulo algumas das principais técnicas de medição de distância, planejamento de trajetória, sistemas de posicionamento e mapas. São vistos também aspectos da fusão de dados sensoriais e um ensaio da utilização do filtro de Kalman nos sistemas de navegação autônoma dos robôs móveis. No apêndice C são apresentadas algumas das técnicas de codificação de dados utilizadas nas transmissões digitais. O apêndice D apresenta uma proposta de implementação de um sistema de navegação autônoma por marcadores ativos¹. No anexo A pode ser visto a listagem da rotina de geração de trajetória

¹Do ingles: *beacons*

que é descrita na seção 5.2. No anexo C encontra-se a listagem da rotina de geração do modelo cinemático do robô móvel que é descrito na seção 5.3.3. A rotina de inicialização dos atuadores, com todos os parâmetros e características de cada atuador utilizado no robô móvel pode ser vista no anexo D com sua utilização feita na seção 4.1.2. No anexo E é apresentada a relação de artigos submetidos em congressos, simpósios e periódicos nacionais e internacionais.

Capítulo 2

Modelagem cinemática de robôs móveis

A cinemática é o estudo mais básico do comportamento dos sistemas mecânicos. Na robótica móvel, deve-se entender o comportamento mecânico do robô para se projetar o robô apropriado para determinada tarefa e para se criar o *software* de controle para o *hardware* do robô móvel.

Os robôs móveis não são os primeiros mecanismos complexos a requererem estas análises. Os manipuladores robóticos têm sido o assunto de estudos intensivos por mais de trinta anos. Em alguns casos, os manipuladores robóticos são muito mais complexos do que os sistemas robóticos móveis: um robô de solda padrão pode ter cinco ou mais juntas, enquanto que alguns robôs móveis são máquinas mais simples de tração diferencial. Nos anos recentes, a comunidade robótica tem alcançado um entendimento razoavelmente completo da cinemática e mesmo da dinâmica (i.e., com relação às forças e massas) dos manipuladores robóticos (CRAIG, 1989) (SCIAVICCO; SICILIANO, 1996).

Os sistemas robóticos móveis têm muitas das mesmas questões cinemáticas dos manipuladores. O espaço de trabalho de um manipulador robótico é crucial porque define a variedade de possíveis posições que pode ser alcançada pelo elemento terminal com relação a base que está fixa no ambiente. O espaço de trabalho do robô móvel é igualmente importante porque define a extensão de possibilidades de posicionamento que pode ser atingido no ambiente. Nos manipuladores, a controlabilidade dos braços robóticos define a maneira com o qual o ajuste ativo dos motores pode ser utilizado para se mover de posição para posição no espaço de trabalho. Similarmente, a controlabilidade dos robôs móveis define possíveis caminhos e trajetórias no seu espaço de trabalho. A dinâmica dos robôs insere restrições adicionais no espaço de trabalho e na trajetória devido as considerações da massa e das forças. O robô móvel também é limitado pela dinâmica; por exemplo, um centro de gravidade alto limita o

raio de curvatura prático de um robô móvel rápido, devido ao perigo da capotagem.

A diferença fundamental entre os sistemas robóticos móveis e os robôs manipuladores é que o primeiro introduz um desafio significativo para estimação de seu posicionamento (SIEGWART; NOURBAKHS, 2004). Um manipulador tem uma das extremidades fixa e a medição da posição do elemento terminal é simplesmente uma questão de se entender a cinemática do robô e medir a posição de todas as juntas intermediárias. A posição do manipulador pode ser sempre calculada através dos dados dos sensores, mas um robô móvel é um sistema autônomo que pode se locomover livremente com respeito ao ambiente. A medição da posição instantânea do robô móvel é muito mais complexa do que a medição da posição do elemento terminal nos manipuladores. Para tal, deve-se integrar o movimento do robô através do tempo. Adiciona-se a isto as imprecisões da estimação do movimento devido à derrapagens e fica claro que medições do posicionamento preciso dos robôs móveis é uma tarefa difícil e desafiadora.

O processo para se entender os movimentos do robô móvel começa com o processo de se descrever a contribuição que cada roda proporciona para o movimento. Cada roda tem seu papel na movimentação de todo o robô móvel, assim como também impõe restrições nos movimentos (e.g., movimentos laterais para robôs móveis com tração diferencial).

Numa primeira abordagem, é apresentado um modelo cinemático para os robôs móveis com tração diferencial, conforme pode ser visto na seção 2.1, onde são apresentadas as relações entre os planos de referência local, do robô móvel, e o plano de referência global, do ambiente. Numa segunda abordagem, é feita referência somente ao plano de referência global do ambiente e ao ponto de centro de curvatura instantânea (CCI), conforme é visto na seção 2.2. Nas demais seções são abordadas outras técnicas de construção de robôs móveis com seus respectivos modelos cinemáticos.

2.1 Modelos cinemáticos utilizando-se referência local e global

Obter um modelo para todas as possibilidades de movimentação do robô é um processo complexo. Cada roda contribui individualmente para o movimento do robô e, ao mesmo tempo, impõe restrições. As rodas são unidas entre si baseadas na geometria do chassi do robô, e suas restrições particulares se combinam para formarem restrições no movimento global do robô. As forças e restrições de cada roda deve ser expressa com respeito a um

plano de referência claro e consistente. Uma mapa claro entre os planos global e local de referência se faz necessário, o que é visto a seguir.

2.1.1 Representação da posição do robô móvel

Através desta análise, o robô móvel é representado como um corpo rígido sobre rodas, operando num plano horizontal. As variáveis de dimensionamento são três: duas para as posições lineares e uma para a orientação angular através do eixo vertical, que é ortogonal ao plano. Existem graus de liberdade adicionais e flexibilidade devido aos eixos das rodas, às juntas de direção e às juntas da roda livre (roda castor).

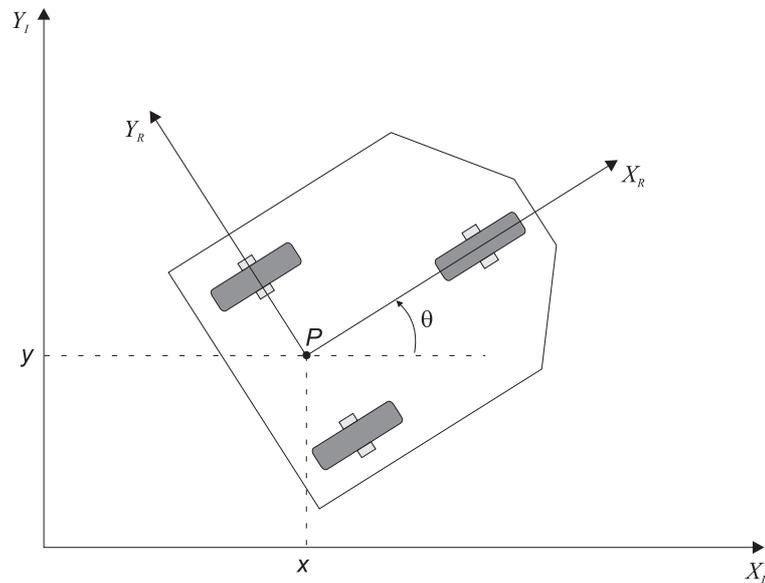


Figura 2.1: O plano de referência global e a referência local do robô.

Para se especificar a posição do robô móvel no plano, estabelece-se uma relação entre o plano de referência global do plano e o plano de referência local do robô, como ilustrado na Figura 2.1. Os eixos X_I e Y_I definem uma base inercial arbitrária no plano. Para se especificar a posição do robô, escolhe-se um ponto P no chassi do robô sendo o ponto de referência. A base X_R, Y_R define dois eixos relativos ao ponto P no chassi do robô e é o plano de referência local do robô. A posição P no plano de referência global é especificada pelas coordenadas x e y , e a diferença angular entre os planos de referência global e local é dada por θ . Pode-se descrever o posicionamento do robô como um vetor composto por estes três elementos. Utiliza-se a notação I para indicar a base deste posicionamento no plano de referência global:

$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (2.1)$$

Para se descrever o movimento do robô em termos dos movimentos dos componentes, é necessário um mapeamento da movimentação ao longo dos eixos do plano global de referência para movimentos ao longo dos eixos do plano de referência local do robô. Um mapeamento é uma função do posicionamento corrente do robô e é conseguido usando a matriz rotacional ortogonal:

$$R(\theta) = \begin{bmatrix} \cos(\theta) & \text{sen}(\theta) & 0 \\ -\text{sen}(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

A matriz $R(\theta)$ pode ser utilizada para o mapeamento da movimentação no plano de referência global X_I, Y_I para a movimentação em termos do plano de referência local X_R, Y_R . Esta operação é denotada por $R(\theta)\xi_I$ porque o cálculo desta operação depende do valor de θ :

$$\xi_R = R(\theta)\xi_I \quad (2.3)$$

Por exemplo, considerando-se o robô da Figura 2.2, como $\theta = \frac{\pi}{2}$, pode-se facilmente calcular-se a matriz de rotação instantânea R :

$$R\left(\frac{\pi}{2}\right) = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

Se o robô estiver se movimentando com certa velocidade $(\dot{x}, \dot{y}, \dot{\theta})$ no plano de referência global pode-se calcular os componentes do movimento com referência aos eixos do robô X_R e Y_R :

$$\dot{\xi}_R = R\left(\frac{\pi}{2}\right)\dot{\xi}_I = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{y} \\ -\dot{x} \\ \dot{\theta} \end{bmatrix} \quad (2.5)$$

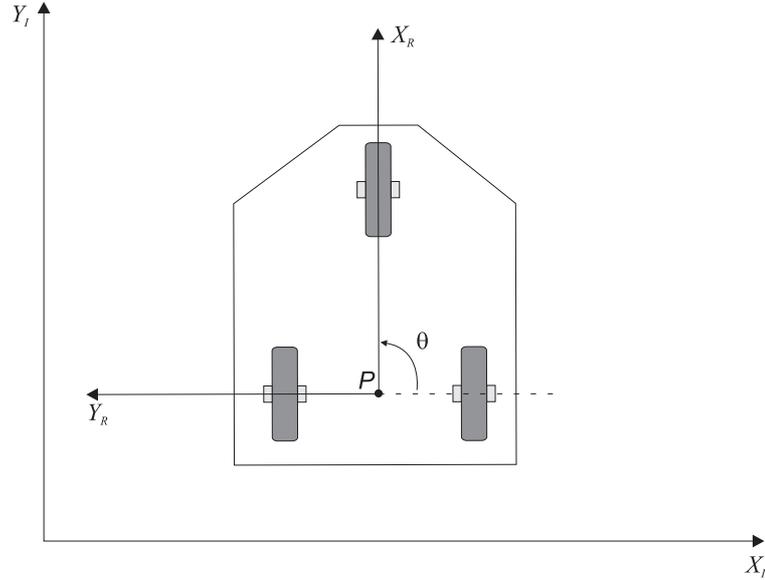


Figura 2.2: O robô móvel está alinhado com os eixos do plano de referência global.

2.1.2 Modelos de cinemática direta

Nos casos mais simples, o mapa descrito na equação 2.3 é suficiente para se gerar a fórmula que captura a cinemática direta do robô móvel.

No modelo ilustrado na Figura 2.3, tem-se um robô móvel com tração diferencial com duas rodas traseiras, cada uma com diâmetro d , com distância entre estas rodas igual a L . Dados d , L , θ , e a velocidade de rotação de cada roda, $\dot{\varphi}_1$ e $\dot{\varphi}_2$, um modelo de cinemática direta que prevê a velocidade do robô no plano geral de referência pode ser dado por:

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(L, d, \theta, \dot{\varphi}_1, \dot{\varphi}_2). \quad (2.6)$$

Da equação 2.3 pode-se computar a variação dos movimentos do robô no plano de referência global a partir da variação dos movimentos no seu plano local de referência:

$$\dot{\xi}_I = R(\theta)^{-1} \dot{\xi}_R. \quad (2.7)$$

Supondo-se que o plano de referência local do robô móvel esteja alinhado de tal forma que o robô se locomova ao longo do eixo $+X_R$, como ilustrado na Figura 2.1. Considerando-se

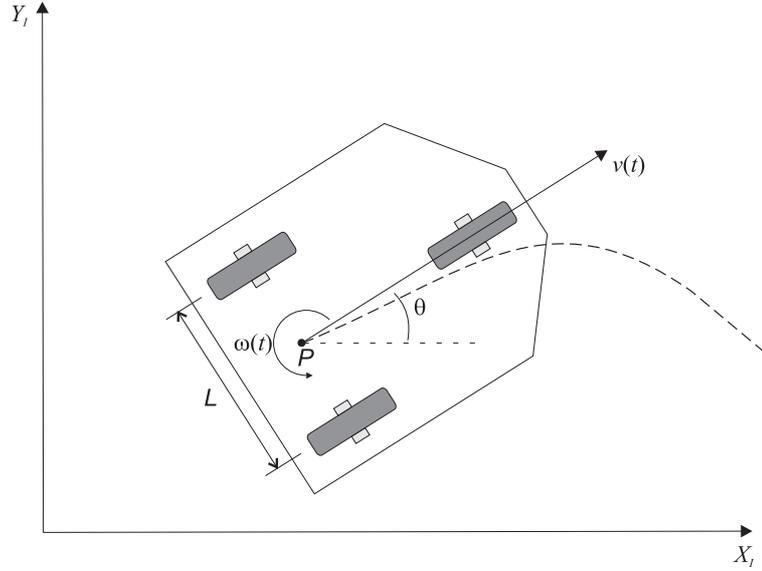


Figura 2.3: Robô móvel com tração diferencial no plano global de referência.

inicialmente a contribuição da velocidade de rotação de cada roda na velocidade de translação do robô no ponto P na direção de $+X_R$. Tem-se que a velocidade da roda direita com referência ao plano local do robô é dada por:

$$v_d = \dot{x}_{R1} = \frac{d\dot{\varphi}_1}{2}, \quad (2.8)$$

e, de forma análoga, a velocidade da roda esquerda é

$$v_e = \dot{x}_{R2} = \frac{d\dot{\varphi}_2}{2}. \quad (2.9)$$

Se, por exemplo, uma das rodas estiver em movimento enquanto a outra estiver parada, uma vez que o ponto P está localizado no meio do eixo das duas rodas, este se moverá com metade da velocidade linear referente a roda em movimento. Num robô com tração diferencial as duas contribuições, v_d e v_e , podem simplesmente ser somadas para se calcular o componente \dot{x}_R de ξ_R . Considerando-se que não haja escorregamentos laterais, \dot{y}_R é sempre igual a zero.

Para se calcular o componente rotacional $\dot{\theta}_R$ de ξ_R , as contribuições de cada roda podem ser computadas independentemente depois somadas. Considerando-se a roda direita (roda 1) com uma rotação no sentido horário o que resulta numa rotação do robô no sentido anti-horário no ponto P e a roda esquerda (roda 2) parada, fazendo o robô pivotar em torno da

mesma. A roda direita está se movendo instantaneamente ao longo do arco de raio L . A velocidade de rotação ω_1 no ponto P pode ser calculada por:

$$\omega_1 = \frac{d\dot{\varphi}_1}{L}. \quad (2.10)$$

De maneira análoga, pode-se aplicar o mesmo raciocínio para a roda esquerda, com a exceção de que um movimento frontal desta roda resulta numa rotação no sentido horário no robô no ponto P :

$$\omega_2 = \frac{-d\dot{\varphi}_2}{L}. \quad (2.11)$$

Da equação 2.7 com a combinação destas fórmulas individuais, tem-se um modelo da cinemática para os robôs com tração diferencial:

$$\dot{\xi}_I = R(\theta)^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = R(\theta)^{-1} \begin{bmatrix} \frac{d\dot{\varphi}_1}{2} + \frac{d\dot{\varphi}_2}{2} \\ 0 \\ \frac{d\dot{\varphi}_1}{L} + \frac{-d\dot{\varphi}_2}{L} \end{bmatrix} = R(\theta)^{-1} \begin{bmatrix} \frac{d}{2}(\dot{\varphi}_1 + \dot{\varphi}_2) \\ 0 \\ \frac{d}{L}(\dot{\varphi}_1 - \dot{\varphi}_2) \end{bmatrix}, \quad (2.12)$$

onde, da equação 2.2 tem-se que

$$R(\theta)^{-1} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.13)$$

Esta abordagem de modelamento cinemático pode fornecer informações sobre o movimento do robô, dadas as velocidades individuais das rodas, nos casos de deslocamentos frontais.

2.2 Modelo cinemático de robôs móveis através do CCI

Dispositivos mecânicos que utilizem rodas ou outros tipos de membros de locomoção aproveitam do atrito com o solo para se locomoverem. Considerando-se que um sistema de locomoção utilize rodas, conforme ilustrado na Figura 2.4, se a roda está livre para girar em torno do eixo (eixo x), o robô desenvolve um movimento preferencial na direção do eixo y e pode realizar um deslocamento na direção do eixo z (deslizamento lateral). Para um sistema

ideal o deslocamento se dará somente na direção do eixo x .

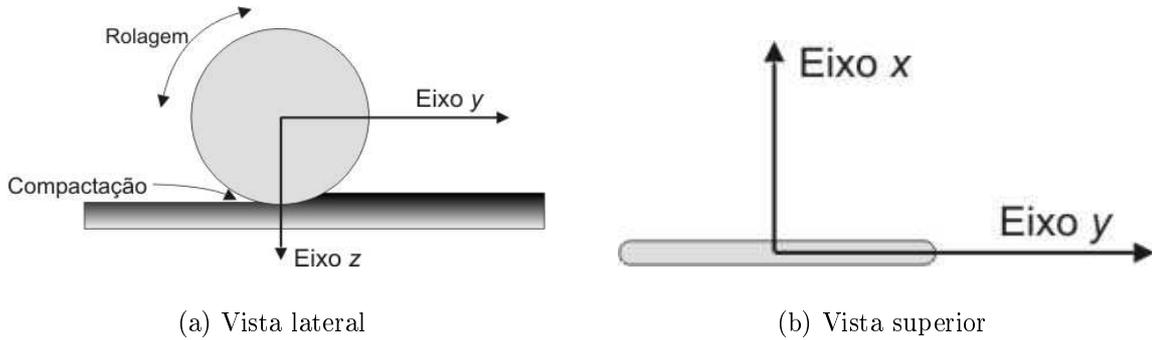


Figura 2.4: Sistema de locomoção ideal com roda

Uma das técnicas mais utilizadas para a estimação do deslocamento linear nos sistemas robóticos móveis é a odometria. Isto significa que a estimativa da distância percorrida é feita pela medida dos giros da roda. No caso ideal, a distância de $2\pi r$ é percorrida em cada rotação da roda com raio r . Na prática, existem outros fatores que influenciam no deslocamento linear do robô. Fora o deslocamento lateral (deslizamento lateral), a insuficiência ou o excesso de tração pode levar a um deslizamento no sentido do eixo y , o que faz a estimativa da distância percorrida ser imprecisa. Fatores adicionais surgem devido a compactação do terreno, conforme ilustrado na Figura 2.4(a), e a coesão entre a superfície e a roda. Devido a estes fatores dependerem das características variáveis do terreno, é muito difícil de se estimar a distância percorrida com precisão se tomada diretamente da rotação da roda. Como os desvios do modelo ideal ocorrem com bastante frequência quando são aplicadas forças de tração ou freagem na roda, utiliza-se normalmente um roda livre (sem tração ou carga) para a medição da estimativa da distância percorrida.

Considerando-se agora um veículo com várias rodas em contato com a superfície do terreno a ser percorrido. Para todas as rodas em contato com a superfície o deslocamento deve ser ao longo do eixo y . Para um sistema robótico móvel deve existir um ponto em que cada roda do veículo segue um curso circular. Este ponto é conhecido como *centro de curvatura instantânea* (CCI) ou como *centro de rotação instantânea* (CRI). Na Figura 2.5(b) as três rodas são arranjadas de tal maneira que linhas através do eixo x tem como intersecção um ponto em comum (o CCI). No exemplo da Figura 2.5(a) isto não ocorre. Um robô móvel com a configuração mostrada na Figura 2.5(b) pode executar movimentos giratórios com facilidade enquanto que sistemas robóticos móveis com eixos das rodas desalinhados, conforme a Figura 2.5(a) não podem.

Para um veículo modificar seu CCI, algumas propriedades das rodas, como as orientações individuais com respeito aos seus eixos verticais, devem ser alteradas. Na prática pode-se identificar o CCI diretamente porque é o ponto de intersecção dos eixos de rotação (eixo x) de cada roda. Existem vários mecanismos de mudança no CCI e alguns deles serão objetos de estudos nas seções seguintes.

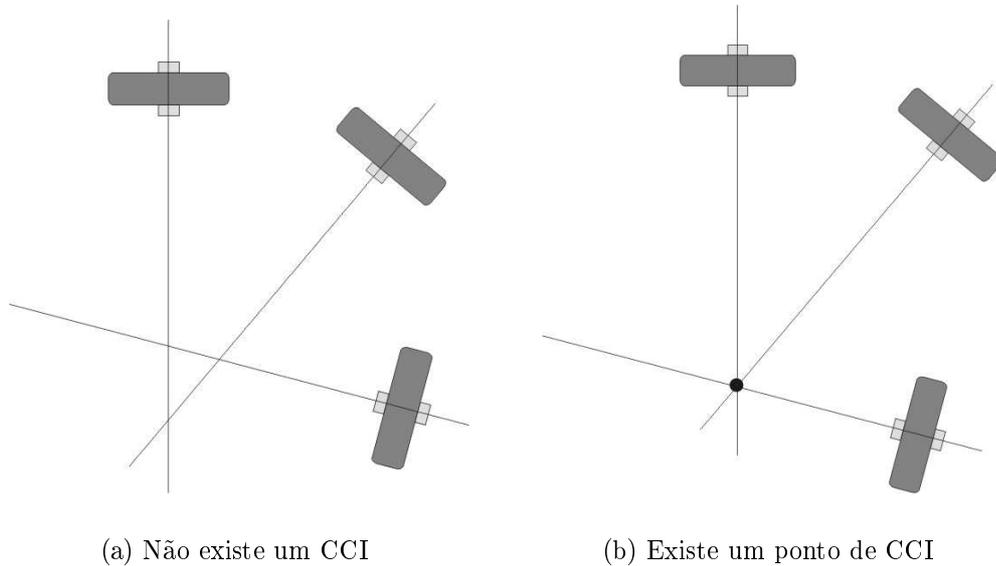


Figura 2.5: Centro de curvatura instantânea (CCI)

Se todas as rodas do veículo estão em contato com o solo, então não só o CCI deve existir como também cada velocidade independente de cada roda deve ser consistente com a rotação de todo veículo com relação ao CCI. Um robô móvel localizado no plano possui três graus de liberdade: a posição (x, y) e a orientação da direção ou orientação θ . O vetor $\mathbf{p} = [x, y, \theta]$ refere-se ao posicionamento¹ do robô no plano.

Os sistemas robóticos móveis normalmente não tem um controle totalmente independente sobre todos os três parâmetros de posicionamento (x, y, θ) e, não raramente, precisam submeter-se a manobras complexas para atingir determinado posicionamento. Por exemplo, se considerarmos um automóvel sendo estacionado em uma vaga, o operador tem que executar uma série complexa de manobras, e as manobras exigidas dependem criticamente da natureza do ambiente e da configuração do veículo. Num automóvel, não é possível mudar o posicionamento $\mathbf{p} = [x, y, \theta]$ arbitrariamente e nem executar mudanças na posição independentemente da orientação do veículo. Este é um exemplo de *restrição não holonômica* (LATOMBE; BAR-

¹Do inglês: *pose*. Posicionamento = posição (x, y) + orientação θ .

RAQUAND, 1993). A quantidade de graus de liberdade que são controláveis, na maioria dos robôs móveis, é de apenas duas, menor do que a quantidade total de graus de liberdade do sistema (três). Este sistema é chamado de *sistema robótico móvel não holonômico* (FIERRO; LEWIS, 1998). Como regra geral, quanto maior for a diferença entre os graus de liberdade controláveis e o total de graus de liberdade, mais difícil será controlar o robô móvel. Se a quantidade total de graus de liberdade e a quantidade de graus de liberdade diretamente controláveis do sistema são iguais, diz-se que é um *sistema robótico móvel holonômico* (GONZALEZ-BANOS; LATOMBE, 2001). Na seção B.2.1 é abordado este tema com relação ao planejamento da trajetória dos robôs móveis e na seção 2.3.2 com relação à cinemática inversa.

Alguns robôs móveis leves são equipados com rodas livres adicionais que provêm suporte físico ao sistema mas não contribuem com a propulsão ou direção. Conhecidas como rodas *castor*, como ilustrado na Figura 2.6, estas rodas são tipicamente ignoradas na computação da cinemática do veículo.

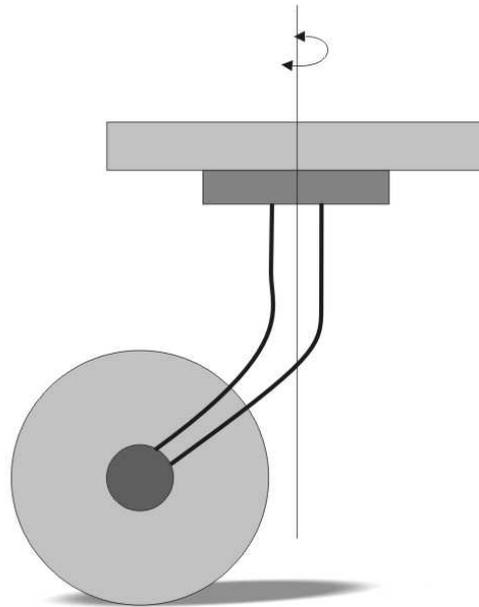


Figura 2.6: Ilustração de uma roda castor

2.3 Sistema de transmissão diferencial

A transmissão diferencial é talvez a opção mais simples de mecanismo de transmissão para sistemas robóticos móveis terrestres. Muito utilizada nos robôs de baixo custo projetados

para uso em ambientes internos ou externos (CHUNG; PARK; HARASHIMA, 2001). A Figura 2.7 apresenta um desenho tridimensional do protótipo do robô móvel experimental com sistema de transmissão diferencial que foi utilizado para validação de parte deste trabalho.

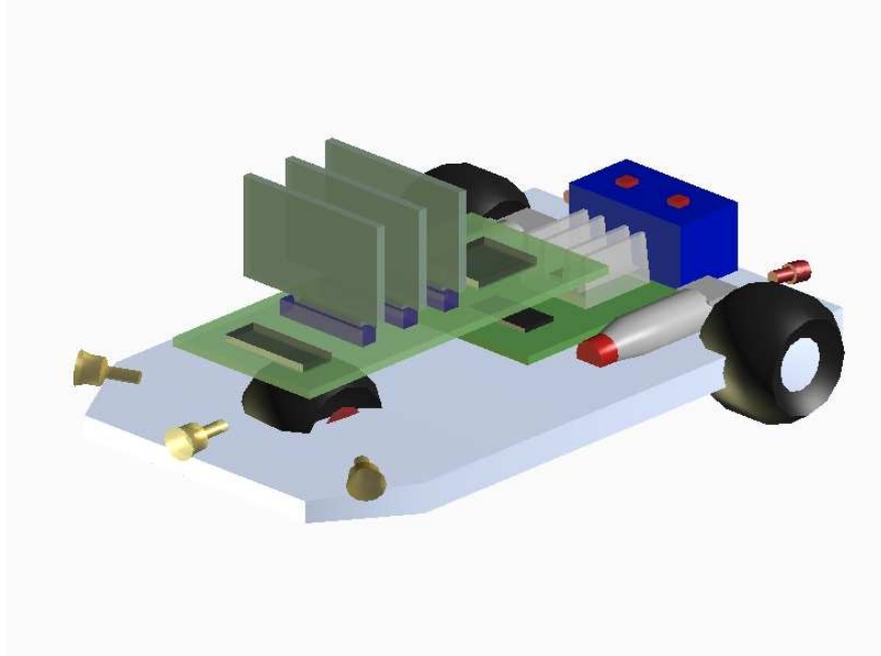


Figura 2.7: Ilustração do protótipo do robô móvel experimental.

Como mostrado na Figura 2.8, a transmissão diferencial destes robôs consiste de duas rodas montada sobre o mesmo eixo de tração controladas por motores individuais.

A cinemática trata da relação entre os parâmetros de controle e o comportamento do sistema no espaço. Considerando que o controle da velocidade das rodas determina o movimento do robô, no caso de transmissão diferencial, para que o robô execute uma curva, o mesmo deve rodar em torno de um ponto que se localiza no eixo comum das duas rodas. Variando a velocidade relativa das duas rodas, o ponto cujo qual o robô executa a rotação pode variar, escolhendo-se diferentes trajetórias. A cada instante de tempo, o ponto de rotação do robô tem que ter a propriedade que faz com que a roda esquerda e a direita sigam um caminho ao redor do CCI com a mesma taxa de velocidade angular ω , então:

$$v_d = \omega(R + L/2) \quad (2.14)$$

e

$$v_e = \omega(R - L/2), \quad (2.15)$$

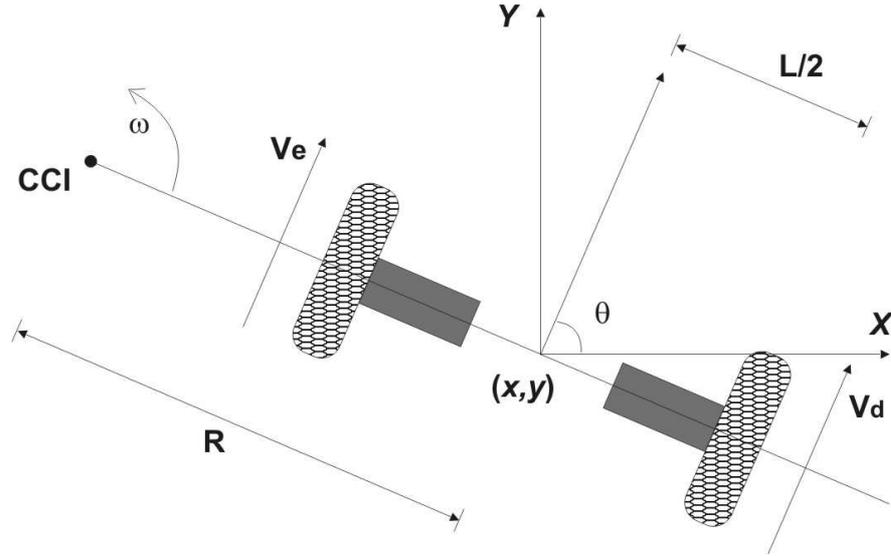


Figura 2.8: A cinemática da transmissão diferencial num robô móvel

onde L é a distância dos centros das duas rodas ao longo do eixo das rodas, a velocidade linear da roda esquerda em relação ao solo é v_e e da direita v_d , e R é a distância do CCI até o ponto central entre as duas rodas. Nota-se que v_e , v_r , ω e R são funções do tempo. A cada instante de tempo, resolvendo-se as equações 2.14 e 2.15 para R tem-se que

$$\frac{v_d}{R + L/2} = \frac{v_e}{R - L/2} \quad (2.16)$$

e rearranjando-se a equação 2.16, resulta-se em

$$R = \frac{L(v_d + v_e)}{2(v_d - v_e)}. \quad (2.17)$$

Resolvendo-se as equações 2.14 e 2.15 para ω , tem-se que

$$\frac{v_d}{\omega} - \frac{L}{2} = \frac{v_e}{\omega} + \frac{L}{2} \quad (2.18)$$

e, rearranjando-se a equação 2.18, resulta-se que

$$\omega = \frac{v_d - v_e}{L}. \quad (2.19)$$

Um número especial de casos são de interesse relevante. Se $v_e = v_d$, então o raio R é infinito e o robô se move numa linha reta. Se $v_e = -v_d$, então o raio é zero e o robô roda sob um ponto central entre as duas rodas (i.e., ele roda sem se deslocar linearmente). Estas

propriedade tornam a transmissão diferencial atrativa para robôs que necessitam navegar em ambientes estreitos ou limitados. Para outros valores de v_e e v_d , o robô não se moverá em linha reta, mas seguirá uma trajetória curva ao redor de um ponto de distância R do centro do robô, mudando ambos: a posição e a orientação do robô.

A estrutura cinemática do veículo proíbe certos movimentos. Por exemplo, não existe combinação possível para v_e e v_d de tal forma que o robô possa se mover diretamente ao longo do eixo das rodas.

Um veículo com transmissão diferencial é muito sensível a velocidade relativa das duas rodas. Pequenos erros na velocidade transmitida a cada roda resulta em diferentes trajetórias. Veículos com transmissão diferencial tipicamente utilizam rodas castor. Eles são sensíveis a pequenas variações no plano do solo. Isto limita sua aplicabilidade em ambientes não laboratoriais.

2.3.1 Cinemática direta para transmissão diferencial

Supondo-se que o robô esteja em um dado ponto (x, y) e voltado para uma posição ao longo de uma linha perfazendo um ângulo θ com o eixo x , conforme ilustrado na Figura 2.8. Através da manipulação dos parâmetros de controle v_e e v_d , o robô pode ser conduzido a diferentes posicionamentos. A determinação dos posicionamentos possíveis de serem alcança-

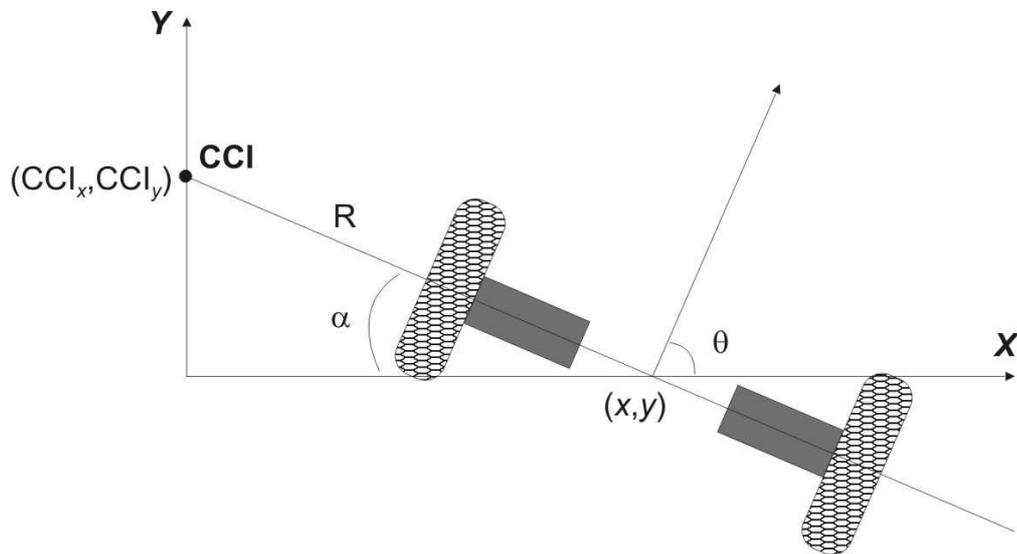


Figura 2.9: Geometria da cinemática direta num robô com transmissão diferencial.

dos, dados os parâmetros de controle, é conhecido como problema de cinemática direta para

o robô. Conforme ilustrado na Figura 2.9, onde o robô se encontra localizado na posição (x, y, θ) , tem-se que

$$\alpha = \frac{\pi}{2} - \theta \quad (2.20)$$

e, pelas relações trigonométricas do sistema,

$$\text{CCI}_x = x - R \cos(\alpha) \quad (2.21)$$

$$\text{CCI}_y = y - R \text{sen}(\alpha). \quad (2.22)$$

Substituindo-se 2.20 em 2.21 e 2.22

$$\text{CCI}_x = x - R \cos\left(\frac{\pi}{2} - \theta\right) \quad (2.23)$$

$$\text{CCI}_y = y - R \text{sen}\left(\frac{\pi}{2} - \theta\right) \quad (2.24)$$

o que resulta em

$$\text{CCI}_x = x - R \text{sen}(\theta) \quad (2.25)$$

$$\text{CCI}_y = y + R \cos(\theta) \quad (2.26)$$

ou

$$\text{CCI} = [x - R \text{sen}(\theta), y + R \cos(\theta)]. \quad (2.27)$$

Como ilustrado na Figura 2.10, ir-se-á agora calcular a posição do robô no tempo $t \rightarrow t + \delta t$. Das equações 2.25 e 2.26, para o instante de tempo t onde a posição do robô é representada por $P(t) = (x, y, \theta)$, tem-se que

$$R = \frac{x - \text{CCI}_x}{\text{sen}(\theta)} \quad (2.28)$$

$$R = -\frac{y - \text{CCI}_y}{\cos(\theta)}. \quad (2.29)$$

Para o instante de tempo $t = t + \delta t$, tem-se que $P(t + \delta t) = (x', y', \theta')$, e, como o CCI e R permanecem constante, de 2.25 e 2.26 tem-se que

$$\text{CCI}_x = x' - R \text{sen}(\theta') \quad (2.30)$$

$$\text{CCI}_y = y' + R \cos(\theta'). \quad (2.31)$$

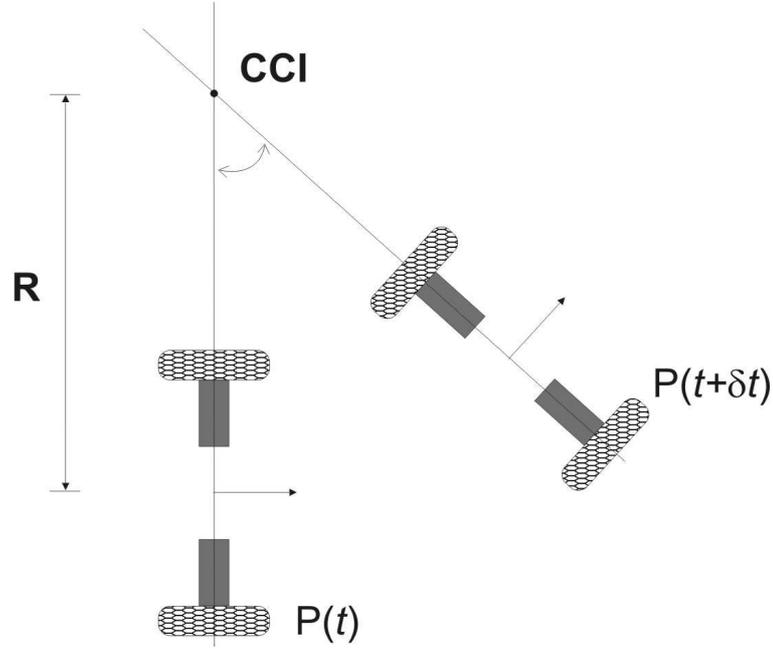


Figura 2.10: Cinemática direta num robô com transmissão diferencial em $t \rightarrow t + \delta t$.

Como o deslocamento angular no instante $t = t + \delta t$ é igual a $\omega \delta t$, então

$$\theta' = \theta + \omega \delta t. \quad (2.32)$$

substituindo-se a equação 2.32 nas equações 2.30 e 2.31 e as equações 2.28 e 2.29 nas equações 2.30 e 2.31 respectivamente, tem-se

$$x' = (x - CCI_x) \cos(\omega \delta t) + (x - CCI_x) \frac{\cos(\theta)}{\sin(\theta)} \sin(\omega \delta t) + CCI_x \quad (2.33)$$

$$y' = (y - CCI_y) \cos(\omega \delta t) - (y - CCI_y) \frac{\sin(\theta)}{\cos(\theta)} \sin(\omega \delta t) + CCI_y. \quad (2.34)$$

Das equações 2.28 e 2.29 tem-se a igualdade em R

$$R = \frac{x - CCI_x}{\sin(\theta)} = -\frac{y - CCI_y}{\cos(\theta)} \quad (2.35)$$

onde resulta que

$$\frac{\sin(\theta)}{\cos(\theta)} = -\frac{x - CCI_x}{y - CCI_y}. \quad (2.36)$$

Substituindo-se 2.36 nas equações 2.33 e 2.34, tem-se finalmente

$$x' = (x - CCI_x) \cos(\omega \delta t) - (y - CCI_y) \sin(\omega \delta t) + CCI_x \quad (2.37)$$

$$y' = (x - CCI_x) \text{sen}(\omega\delta t) + (y - CCI_y) \text{cos}(\omega\delta t) + CCI_y. \quad (2.38)$$

Desta maneira, como v_e e v_d e conseqüentemente R e ω são funções do tempo, está demonstrado que para o sistema ilustrado na Figura 2.10, que, se o robô está na posição (x, y, θ) no tempo t , e se a roda esquerda e direita têm velocidade de contato com o solo v_e e v_d respectivamente, então, no tempo $t \rightarrow t + \delta t$ a posição do robô é dado por

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \text{cos}(\omega\delta t) & -\text{sen}(\omega\delta t) & 0 \\ \text{sen}(\omega\delta t) & \text{cos}(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - CCI_x \\ y - CCI_y \\ \theta \end{bmatrix} + \begin{bmatrix} CCI_x \\ CCI_y \\ \omega\delta t \end{bmatrix} \quad (2.39)$$

A equação 2.39 descreve o movimento rotacional do robô a uma distância R do CCI com um velocidade angular dada por ω . Diferentes classes de robôs resultarão em diferentes expressões matemáticas para R e ω (SHIM; KIM; KOH, 1995).

Se integrarmos a equação 2.39 a partir das condições iniciais (x_0, y_0, θ_0) , é possível se calcular onde o robô estará em qualquer tempo t baseado nos parâmetros de controle $v_e(t)$ e $v_d(t)$, o que significa resolver o problema da cinemática direta para o sistema. De um modo geral, para um robô qualquer capaz de se mover em uma certa direção $\theta(t)$ a uma dada velocidade $V(t)$ tem-se

$$\begin{aligned} x(t) &= \int_0^t V(t) \text{cos}[\theta(t)] dt \\ y(t) &= \int_0^t V(t) \text{sen}[\theta(t)] dt \\ \theta(t) &= \int_0^t \omega(t) dt, \end{aligned} \quad (2.40)$$

e, para nosso caso particular de um sistema robótico com transmissão diferencial, tem-se que

$$\begin{aligned} x(t) &= \frac{1}{2} \int_0^t [v_d(t) + v_e(t)] \text{cos}[\theta(t)] dt \\ y(t) &= \frac{1}{2} \int_0^t [v_d(t) + v_e(t)] \text{sen}[\theta(t)] dt \\ \theta &= \frac{1}{L} \int_0^t [v_d(t) - v_e(t)] dt. \end{aligned} \quad (2.41)$$

Uma questão importante e mais difícil de se equacionar é, como selecionar os parâmetros de controle do robô móvel para que o mesmo alcance um posicionamento específico global ou siga uma trajetória específica. A solução deste problema se encontra na determinação da

cinemática inversa do robô móvel. A cinemática inversa inverte a relação entre as entradas do controle e seu comportamento. Está também relacionada com problemas de planejamento de trajetória.

2.3.2 Cinemática dos robôs móveis não holonômicos

As equações de 2.41 descrevem situações de restrição na velocidade do robô que não podem ser integradas num posicionamento onde exista esta restrição. Como pode ser visto no item 2.2, são conhecidas como *restrições não holonômicas* e, em geral, são muito difíceis de serem resolvidas, embora existam soluções para um classe limitada de funções de controle $v_e(t)$ e $v_d(t)$ (ZHAO; BEMENT, 1992). Por exemplo, se for assumido que $v_e(t) = v_e$, $v_d(t) = v_d$ e $v_e \neq v_d$, então as equações 2.41 resultam em

$$\begin{aligned} x(t) &= \frac{L v_d + v_e}{2 v_d - v_e} \operatorname{sen} \left[\frac{t}{L} (v_d - v_e) \right] \\ y(t) &= -\frac{L v_d + v_e}{2 v_d - v_e} \cos \left[\frac{t}{L} (v_d - v_e) \right] + \frac{L v_d + v_e}{2 v_d - v_e} \\ \theta(t) &= \frac{t}{L} (v_d - v_e), \end{aligned} \quad (2.42)$$

onde $(x, y, \theta)_{t=0} = (0, 0, 0)$. Dado um certo tempo t e uma certa posição (x, y) , as equações 2.42 podem ser resolvidas para v_d e v_e mas não fornecem um controle independente em termos de θ . Existem, na realidade, infinitas soluções para v_d e v_e a partir das equações 2.42, mas todas correspondem ao deslocamento do robô sob o mesmo círculo de raio R que passa através do ponto $(0, 0)$ no instante $t = 0$ e no ponto (x, y) no instante $t = t$, contudo, o robô executa a trajetória circular diferentes números de vezes em diferentes direções.

Se usarmos as equações 2.17 e 2.19, pode-se reescrever as equações 2.42 em função do raio de curvatura R com o CCI e da velocidade angular ω , dado por

$$\begin{aligned} x(t) &= R \operatorname{sen}(\omega t) \\ y(t) &= R[1 - \cos(\omega t)] \\ \theta(t) &= \omega t. \end{aligned} \quad (2.43)$$

Considerando-se dois casos especiais de deslocamento do veículo com transmissão diferencial. Se $v_e = v_d = v$, com $R \rightarrow \infty$ (i.e., o robô se movimenta numa linha reta), então as

equações 2.41 do deslocamento do robô simplificam-se para

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x + v \cos(\theta)\delta t \\ y + v \sin(\theta)\delta t \\ \theta \end{pmatrix}, \quad (2.44)$$

e, escolhendo-se $-v_e = v_d = v$, que é o segundo caso particular (i.e. o robô roda sobre seu próprio eixo não havendo deslocamento linear), as equações 2.41 resultam em

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta + 2v\delta t/L \end{pmatrix}. \quad (2.45)$$

Desta maneira, uma técnica de controle para posicionar o robô em algum ponto específico (x, y, θ) , é por exemplo, rotacionar o robô no lugar, sem deslocamento linear, até que esteja posicionado no sentido do ponto (x, y) . Então, deve-se conduzir o robô em linha reta ($v_e = v_d$) até que atinja o ponto (x, y) especificado, rotacionando-se então o robô sobre o seu próprio eixo até atingir a orientação θ desejada.

Não existem somente as soluções apresentadas anteriormente para a solução da cinemática dos sistemas robóticos móveis com transmissão diferencial. Outras soluções, como as baseadas em mudanças suaves na trajetória, são possíveis (SCHEUER; FRAICHARD, 1997) (LAVALLE, 2006).

2.4 Transmissão síncrona em robôs móveis

Na transmissão síncrona (*synchro drive*), cada roda é capaz de ser tracionada e rotacionada individualmente. Uma configuração típica envolve três rodas rotacionáveis arranjadas no robô formando um triângulo equilátero, muitas das vezes montadas em plataformas cilíndricas. Estas rodas permitem o controle da orientação rotacional. Todas as rodas viram e tracionam em sincronia. Todas as rodas sempre apontam para mesma direção e rotacionam com a mesma taxa. Isto é conseguido tipicamente através de mecanismos com correias que ligam fisicamente as rodas. Num robô com transmissão síncrona, o sistema controla a direção para onde as rodas devem apontar e a velocidade angular das mesmas.

Um arranjo mecânico comum para os veículos com transmissão síncrona é o uso de dois motores independentes, um que traciona todas as rodas para frente e outro que executa a rotação das mesmas. Pelo fato de que todas as rodas se mantêm em paralelo, estes veículos

sempre giram em redor do seu centro. A habilidade de controlar de modo independente a rotação e a velocidade linear do robô faz com que o controle seja mais simplificado permitindo que estes tipos de robôs sirvam de modelos convenientes para um robô pontual ideal. A Figura 2.11 ilustra um exemplo de robô móvel comercial que utiliza transmissão síncrona (SPROUSE, 2004).

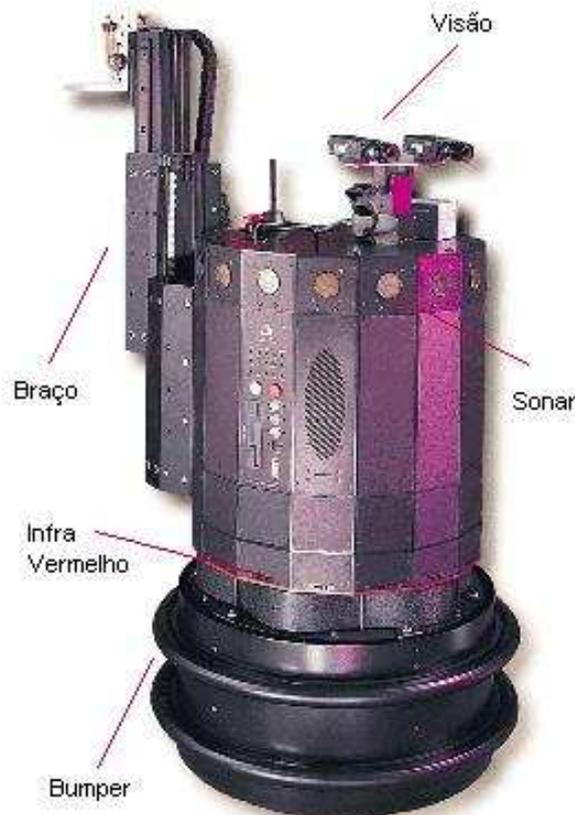


Figura 2.11: Robô Nomad 200 fabricado pela Nomadic Technologies. Utiliza transmissão síncrona para locomoção.

Robôs projetados a partir das tecnologias de transmissão diferencial ou síncrona são sensíveis a pequenas variações no plano do solo. No caso da transmissão diferencial, uma pequena variação no plano do solo pode fazer com que uma das rodas do robô perca o contato com o solo causando erros no posicionamento do robô, ao passo que nos robôs com transmissão síncrona, uma variação no plano da superfície de contato das rodas pode causar problemas de alinhamento das rodas pela razão de que a distância que cada roda percorre é diferente.

2.4.1 Cinemática direta para transmissão síncrona

Robôs com transmissão síncrona giram em redor do seu centro numa razão ω . A velocidade translacional v pode ser controlada diretamente. Para esta categoria de robôs, a cinemática direta vem das equações 2.40, onde

$$\begin{aligned}x(t) &= \int_0^t V(t) \cos[\theta(t)] dt \\y(t) &= \int_0^t V(t) \text{sen}[\theta(t)] dt \\ \theta(t) &= \int_0^t \omega(t) dt.\end{aligned}\tag{2.46}$$

Nota-se que o CCI para um robô com transmissão síncrona está sempre no infinito e que mudando a orientação das rodas manipula-se a direção do CCI. A Figura 2.12 ilustra um exemplo da cinemática de robôs com transmissão síncrona.

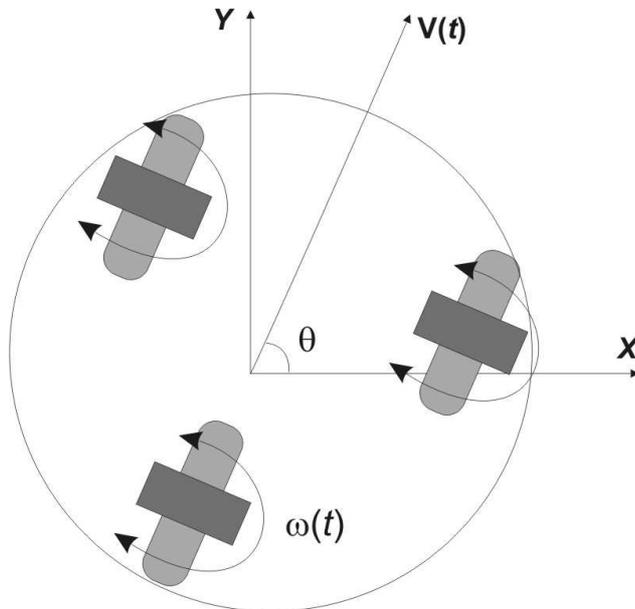


Figura 2.12: Cinemática de robôs com transmissão síncrona.

2.4.2 Cinemática inversa para transmissão síncrona

Pelo fato de que mudanças na orientação pode ser completamente independente da translação, a cinemática inversa dos robôs com transmissão síncrona é muito similar ao

caso especial da cinemática inversa para transmissão diferencial, apresentada no ítem 2.3.2. Pode-se considerar dois casos especiais de interesse: (1) Se $v(t) = 0$ e $\omega(t) = \omega$ num período de tempo δt , então o robô rodará no mesmo lugar pela quantia de $\omega\delta t$; e (2) se $\omega(t) = 0$ e $v(t) = v$ para um período de tempo δt , então o robô move-se na direção em que está apontando a uma distância de $v\delta t$.

2.5 Modelos cinemáticos de robôs móveis com rodas direcionáveis

Os robôs que não são projetados nem com tecnologia de transmissão diferencial nem com síncrona tipicamente possuem uma ou mais rodas que podem ser direcionadas rotacionalmente e uma ou mais rodas cujo eixo rotacional não pode ser mudado (são rodas fixas). Para estes robôs o processo de se calcular a região em potencial do CCI pode ser mais complexo (KELLY, 1994). A seguir é mostrado alguns exemplos destes sistemas.

2.5.1 Bicicletas

Considerando-se agora o problema de se determinar o CCI de uma bicicleta. O mesmo deve encontrar-se na intersecção da linha que passa através do eixo da roda traseira (perpendicular ao corpo da bicicleta) com a linha que passa através do eixo da roda dianteira, conforme ilustrado na Figura 2.13. No giro máximo da roda frontal, a bicicleta tem o raio mínimo de rotação e gira em torno do ponto que passa através do eixo da roda traseira.

2.5.2 Triciclos

Um robô projetado em formato de triciclo com três rodas tem, tipicamente, as duas rodas traseiras fixas (executam rotação em torno de seus eixos mas não distorcem) com odômetros e a roda dianteira executa a tração e o direcionamento. Conforme ilustrado na Figura 2.14 o robô triciclo é controlado pelo direcionamento α e velocidade v providos através da roda dianteira.

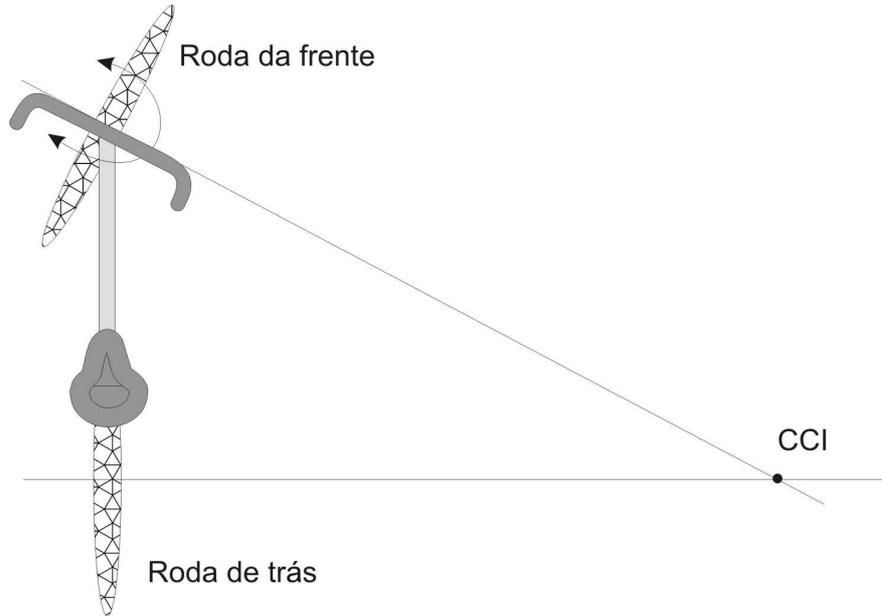


Figura 2.13: Localização do CCI para uma bicicleta.

2.5.2.1 Cinemática direta para robôs móveis com rodas direcionáveis

Se a roda dianteira (roda direcionável) for posicionada a um certo ângulo α com relação ao eixo frontal, o triciclo (ou a bicicleta) irá executar uma rotação com velocidade angular ω ao redor do ponto localizado a distância R ao longo da linha perpendicular às rodas traseiras (linha do eixo das rodas). R e ω são dados por

$$\begin{aligned} R &= d \tan\left(\frac{\pi}{2} - \alpha\right) \\ \omega &= \frac{v}{(d^2 + R^2)^{\frac{1}{2}}}, \end{aligned} \quad (2.47)$$

onde v é a velocidade linear do robô em relação ao solo e d é a distância do eixo da roda dianteira com o eixo das rodas traseiras, como ilustrado na Figura 2.14. Tomando-se como referência o ângulo θ que representa a orientação do robô num determinado instante de tempo, tem-se que

$$\begin{aligned} CCI_x &= x + R \sin(\theta) \\ CCI_y &= y - R \cos(\theta), \end{aligned} \quad (2.48)$$

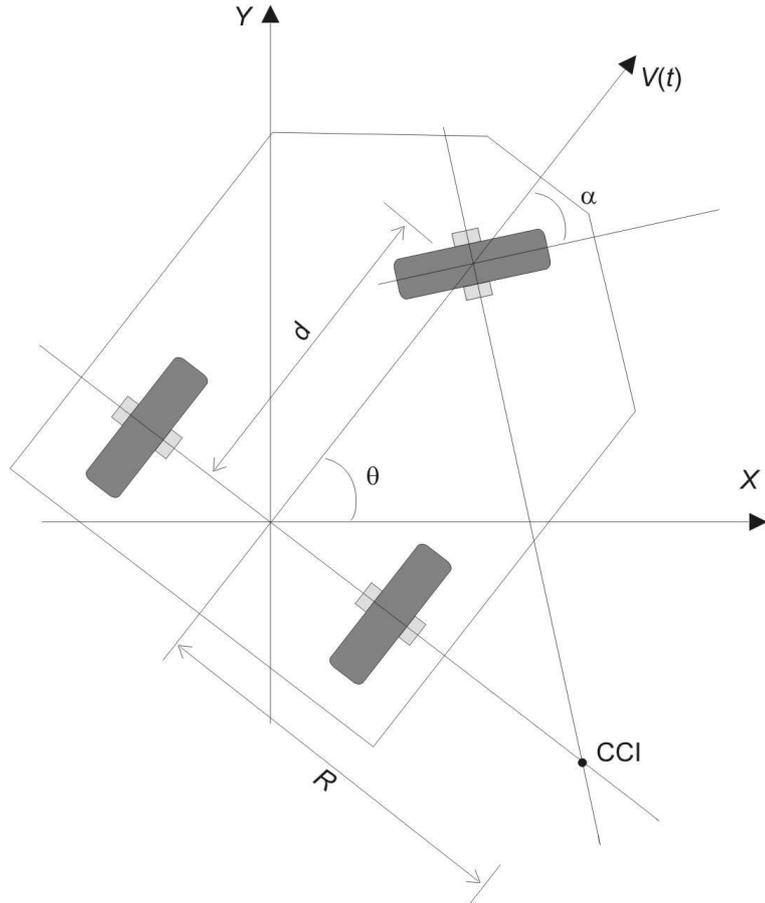


Figura 2.14: Cinemática de um robô triciclo.

e, de maneira análoga ao demonstrado no ítem 2.3.1, encontra-se facilmente que

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega\delta t) & -\text{sen}(\omega\delta t) & 0 \\ \text{sen}(\omega\delta t) & \cos(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - \text{CCI}_x \\ y - \text{CCI}_y \\ \theta \end{bmatrix} + \begin{bmatrix} \text{CCI}_x \\ -\text{CCI}_y \\ \omega\delta t \end{bmatrix}, \quad (2.49)$$

onde (x', y', θ') é a posição do robô no instante de tempo $t \rightarrow t + \delta t$.

As equações 2.47 podem ser reagrupadas em função da velocidade angular ω , da velocidade linear v e do ângulo α do robô

$$\omega = \frac{v}{\{d^2[1 + \tan^2(\frac{\pi}{2} - \alpha)]\}^{\frac{1}{2}}}. \quad (2.50)$$

Substituindo-se 2.50 na equação 2.49 obtém-se a equação da cinemática direta do robô.

2.5.2.2 Cinemática inversa para robôs móveis com rodas direcionáveis

Como no caso dos sistemas robóticos com transmissão diferencial, encontrar-se a cinemática inversa de uma forma geral é muito difícil. É mais profícuo olhar-se para os casos especiais. Dois casos são de particular importância. Se $\alpha = 0$, o robô se deslocará em linha reta para frente e a equação 2.49 se reduz à

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x + v \cos(\theta)\delta t \\ y + v \sin(\theta)\delta t \\ \theta \end{pmatrix}. \quad (2.51)$$

Se o veículo é capaz de executar manobras de $\pm 90^\circ$ com a roda dianteira, então pode-se rodar o robô em torno do seu próprio eixo, sem haver deslocamento linear, e

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \pm v\delta t/d \end{pmatrix}. \quad (2.52)$$

Note que se a roda dianteira não pode ser distorcida de $\pm 90^\circ$, será impossível mudar a orientação do robô sem mudar sua posição. Uma limitação no range de rotação da roda resulta em limites no raio de curvatura do círculo que o veículo pode executar. Esta limitação no raio de curvatura é característica na maioria dos robôs que utilizam rodas direcionáveis.

A cinemática inversa pode ser resolvida de maneira similar ao demonstrado para os veículos com tração diferencial (item 2.3.2), isto se o robô for capaz de executar manobras de $\pm 90^\circ$ com a roda dianteira. Se, contudo, este ângulo for mais limitado, o problema de conduzir o robô a uma determinada posição (x, y, θ) é consideravelmente mais complexo. Estratégias, como as utilizadas em estacionamento de veículos em vagas paralelas, devem ser consideradas para obtenção de certos posicionamentos.

2.5.3 Sistema automotivo de direção

O sistema de direção automotivo (*Ackerman steering*) é o mesmo utilizado na maioria dos automóveis. No modelo de direção de Ackerman, cada roda dianteira roda em seu próprio eixo podendo rotacionar diferentemente em relação ao CCI, que se localiza na linha que passa nas rodas traseiras do veículo, conforme ilustrado na Figura 2.15. A roda dianteira interior ao arco percorrido (mais próxima do CCI) deve rotacionar com um ângulo maior do que a outra roda dianteira ($\alpha_1 > \alpha_2$) e percorrer uma distância menor (o raio de rotação é menor).

O sistema de direção de Ackerman é o mecanismo preferido para a maioria dos automóveis comerciais operando tanto nas rodovias asfaltadas como nas estradas *off-road*. O tamanho do veículos comerciais permitem a instalação de extensivos sistemas computacionais embarcados de sensoriamento e controle, permitindo navegação na grande maioria dos terrenos.

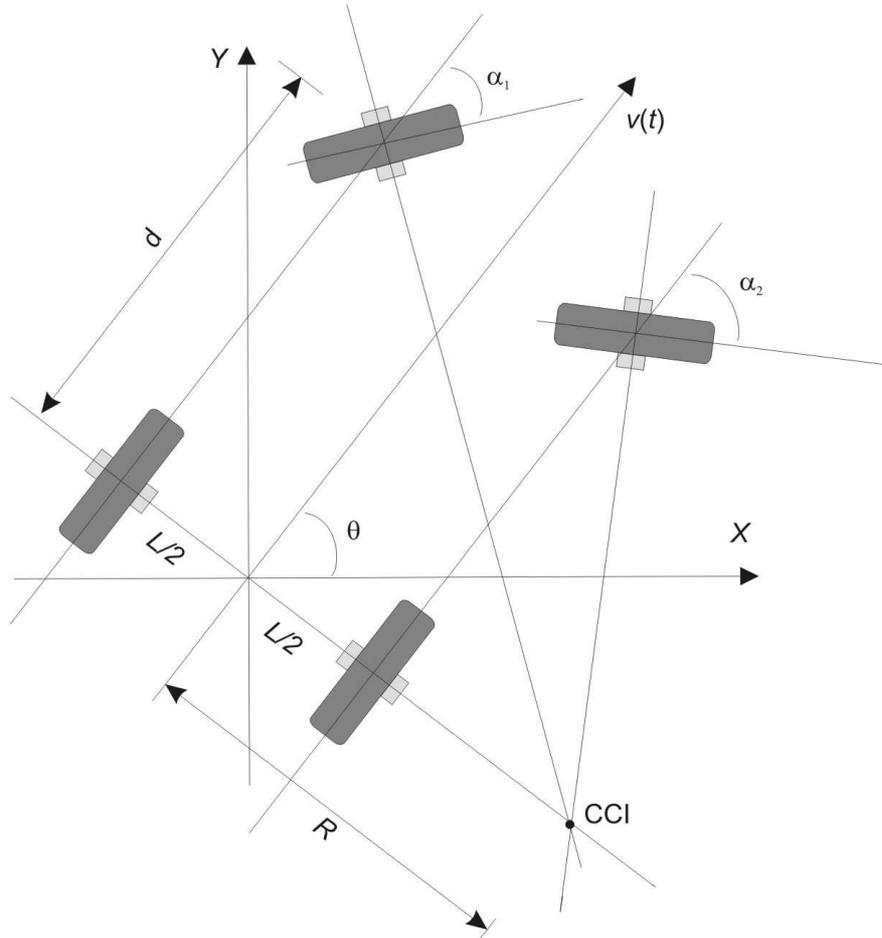


Figura 2.15: Cinemática de um robô como o mecanismo de Ackerman.

2.5.3.1 Cinemática direta para sistemas de direção de Ackerman

No sistema de direção de Ackerman, o veículo rotaciona ao redor do ponto localizado na linha que passa através dos eixos das rodas traseiras, a uma distância R da linha central do veículo. Desta maneira, conforme ilustrado na Figura 2.15, temos então para roda dianteira interna que

$$R + \frac{L}{2} = d \tan\left(\frac{\pi}{2} + \alpha_1\right). \quad (2.53)$$

Para roda exterior ao arco percorrido, que rotaciona com o ângulo α_2 , tem-se que

$$R - \frac{L}{2} = d \tan\left(\frac{\pi}{2} + \alpha_2\right). \quad (2.54)$$

Em geral, todas as quatro rodas viajam pelo solo com diferentes velocidades, e especificando-se a velocidade de uma das rodas tem-se a velocidade de todas elas. O sistema de direção de Ackerman é complexo e não é a intenção deste trabalho estar se aprofundando no tema, mas sim descrever de forma sucinta o seu princípio de funcionamento cinemático.

2.6 Considerações finais

Este capítulo apresentou os aspectos relacionados a modelagem cinemática de alguns dos sistemas robóticos móveis mais comumente encontrados. É feita uma revisão bibliográfica dos modelos cinemáticos de robôs móveis com tração diferencial nas seções 2.1 e 2.3. Todo desenvolvimento matemático do modelo cinemático do robô móvel utilizado como padrão no simulador virtual (capítulo 5) pode ser visto nas seções 2.3.1 e 2.3.2. É visto também, a título de revisão bibliográfica, os aspectos cinemáticos dos sistemas robóticos móveis com transmissão síncrona e os modelos cinemáticos de robôs móveis com rodas direcionáveis. Desta maneira, se no futuro optar-se por outro modelo a ser implementado no simulador virtual, que não o com tração diferencial, já se encontra disponível seus respectivos equacionamentos cinemáticos.

Capítulo 3

Proposta de arquitetura aberta de supervisão e controle embarcada

Entre os campos relacionados para a realização de projetos de sistemas embarcados, as tecnologias de hardware e software estão tendo um crescimento significativo. Isto é particularmente verdadeiro para a evolução de motores, sensores, microcontroladores, microprocessadores, interfaces de comunicação e interfaces de potência. Partindo deste fato, a idéia é elaborar estruturas abertas, que se adaptem facilmente ao desenvolvimento destas tecnologias.

Neste capítulo, é apresentada uma proposta de arquitetura aberta de supervisão e controle embarcada a ser utilizada em sistemas robóticos móveis, onde é descrito o projeto estruturado em pequenos módulos independentes, com interfaces de comunicação, inseridos dentro de uma arquitetura aberta orientada à hierarquia. Para isto, se faz necessário uma pequena introdução, a título de revisão bibliográfica, das técnicas e dispositivos utilizados no projeto. Esta introdução é vista nas primeiras seções deste capítulo. A aplicação final visa o desenvolvimento de um sistema de controle embarcado para dispositivos robóticos móveis que terá por finalidade a navegação, dentro de um ambiente interno ou externo, possibilitando a aplicação do conceito de prototipagem rápida.

3.1 Sistemas reconfiguráveis

Sistemas reconfiguráveis são sistemas que apresentam a característica de, através da substituição de parte de seu software ou hardware, adaptarem-se a tarefas específicas. Eles têm por objetivo obter alto desempenho com baixo custo de produção sendo uma alternativa às máquinas de Von Neuman implementadas pelos sistemas com microprocessadores. Sistemas

de software reconfigurável são bastante comuns, sendo exemplos os sistemas embarcados em automóveis, em eletrodomésticos ou mesmo em vídeo-games. Nestes sistemas, a mudança de uma memória ROM (*Read-Only Memory*) ou de um CD-ROM leva a reconfiguração de funções responsáveis pela operação dos mesmos (MIYAZAKI, 1998). Sistemas de hardware reconfigurável são mais recentes e estão associados ao aparecimento das FPGA (*Field Programmable Gate Array*) na década de 90. O advento de dispositivos como FPGAs mudou o ponto de equilíbrio entre compromisso entre flexibilidade e performance. Com as FPGA os objetivos de performance são mantidos com o aumento de flexibilidade (KALRA, 2001). Page (1996) propõe a idéia de unir em um único circuito integrado um microprocessador e uma FPGA para atender dinamicamente novas aplicações.

Comparados aos sistemas de software reconfigurável os sistemas de hardware reconfigurável apresentam um maior potencial em termos de performance e adaptabilidade. Outros nomes estão associados aos sistemas de hardware reconfigurável: CCM (*Custom Computing Machines*) ou FCCM (*FPGA-based Custom Computing Machines*). Os termos sistema reconfigurável, computação reconfigurável ou lógica reconfigurável também estão associados a sistemas de hardware reconfigurável.

Tradicionalmente, a execução de um algoritmo na computação convencional pode seguir dois métodos: usar uma tecnologia de hardware, a exemplo de ASICs (*Application Specific Integrated Circuits*) ou de placas de circuito impresso, ou utilizando microprocessadores programáveis por software. O primeiro método apresenta como vantagem a grande velocidade de execução, mas não é flexível a modificações posteriores. O segundo método, embora apresentando alta flexibilidade para modificações, não é executado com a mesma velocidade dos algoritmos executados por hardware (COMPTON, 2002). O fato dos microprocessadores executarem de forma seqüencial suas tarefas (máquina de Von Neumann) fazem com que os mesmos tenham tempos de processamento que não são aceitáveis para muitas aplicações (PIMENTEL; LE-HUY, 2000)(DIDO et al., 2002). A computação reconfigurável tem por objetivo suprir a lacuna entre a solução por software e a solução por hardware, atingindo desempenhos muito superiores aos desempenhos obtidos por software, mas obtendo uma flexibilidade muito maior que a flexibilidade obtida por uma solução por hardware (CHEN; CHEN; CHEN, 2000)(COMPTON, 2002)(CORIC; LEESER; MILLER, 2002).

De outro modo, sistemas embarcados apresentam grande dinâmica tecnológica, ou seja, estão sujeitos a grandes variações tecnológicas em curto espaço de tempo, seja por demandas por novas tarefas e performances, seja por disponibilidade de novas tecnologias de sensores e atuadores. Restrições de recursos em sistemas embarcados acentuam a necessidade de

novas idéias de projeto. É comum a utilização de blocos de software funcionalmente testados para acelerar o tempo de projeto de um sistema (API - *Application Program Interface*). Em hardware existem blocos funcionais que podem assumir a mesma função. Estes blocos podem ser combinados com outros circuitos para implementar um determinado algoritmo. Este aspecto modular permite facilitar a manutenção e modernização de projetos (RENNER et al., 2002). A utilização de IP-CORE (*Intellectual Property Cores*), permite a integração de soluções já desenvolvidas por diversos fornecedores para minimizar o tempo de projeto. Barramento PCI, interfaces de comunicação, e funções de processamento de sinal, como FFT (*Fast Fourier Transform*), codificação e decodificação de imagens digitais e mesmo Microprocessadores e DSPs são exemplos de IP-CORE disponíveis (ITO; CARRO, 2000)(KEAN, 2000).

As PLDs (*Programmable Logic Devices*), também chamadas de EPLD (*Erasable Programmable Logic Devices*), ou ainda CPLD (*Complex Programmable Logic Devices*), e as FPGA ou DFGA (*Dinamicallly Field Programmable Gate Array*) são dispositivos que permitem a execução de algoritmos diretamente em hardware. Com estes dispositivos programáveis é possível executar os algoritmos explorando o paralelismo inerente da solução por hardware, executando-os muito mais rápido do que se o mesmo algoritmo fosse executado de forma seqüencial por microcontroladores ou por DSPs, sujeitos ao modelo de Von Neumann. As PLDs e FPGAs apresentam diferenças quanto a aplicação e quanto a estrutura interna. As FPGA são geralmente aplicadas em sistema com menor sensibilidade ao custo e de maior complexidade (DINIZ; PARK, 2002).

Paralelamente ao desenvolvimento de dispositivos que permitiram a implementação da computação reconfigurável, ferramentas de projeto, depuração, simulação e testes foram desenvolvidos pelos diversos fabricantes. Estes ambientes permitem a criação de módulos desenvolvidos com linguagens de alto nível de abstração chamadas linguagens de descrição de hardware, a exemplo de VHDL e AHDL (ALTERA, 2007)(NAVABI; DAY, 1991)(D'AMORE, 2005). De outro modo, é possível implementar blocos com representações de mais baixo nível de abstração, a exemplo de esquemáticos. A integração de blocos criados com diferentes linguagens permite a criação de projetos de forma bastante flexível, permitindo uma fácil interação de uma equipe de trabalho.

O tópico de arquitetura de sistemas abertos vem sendo estudado nos últimos anos por várias instituições de pesquisas nas áreas de engenharia aplicada à máquinas ferramentas e engenharia de produção, visando os aspectos da modularidade bem como seus efeitos na performance do sistema (BHASKER, 1999). O objetivo do conceito de arquitetura reconfigurável

é permitir uma fácil e rápida adaptação da estrutura de comando do robô móvel a estas evoluções tecnológicas, para uma melhor portabilidade e permutabilidade do sistema final. Através da divisão da estrutura em pequenos blocos funcionais, com interfaces dedicadas bem específicas, a modularização do projeto se torna eficiente. Isto permite primeiro uma melhor especificação das tarefas de desenvolvimento para uma equipe de pesquisadores multidisciplinar, então a adaptação de um bloco particular a uma nova evolução tecnológica. Tudo isto não pode ser encontrado até agora nos robôs móveis comerciais mais comuns, que são bastante fechados para permitirem estas manipulações. Os robôs móveis, como plataformas para consolidação do conhecimento em várias áreas de ensino e pesquisa, tais como modelagem, controle, automação, sistemas de potência, sensores, atuadores, transmissão de dados, sistemas eletrônicos embutidos e engenharia de software, estão sendo mais e mais utilizados em institutos de ensino e pesquisa que trabalham nesta área (CHANG, 1997).

O objetivo deste conceito aplicado à arquitetura reconfigurável é permitir uma fácil e rápida adaptação de dispositivos embarcados a novas evoluções tecnológicas, para uma melhor portabilidade e capacidade de intercâmbio para o sistema final. A divisão da estrutura em pequenos blocos funcionais, com interfaces bem especificadas, permite uma melhor especificação das tarefas de uma equipe de projeto multidisciplinar, bem com a rápida adaptação de um determinado bloco a uma nova evolução tecnológica. Este tipo de solução não pode ser alcançada em muitos dos sistemas comerciais disponíveis, pois estes não fornecem informações ou não apresentam recursos para permitir este tipo de manipulação (MELO; ROSÁRIO, 2007).

3.2 Processamento digital de sinais

O processamento digital de sinais é um método de processamento de sinais e dados com a finalidade de se intensificar ou modificar estes sinais, ou de analisar estes sinais para se determinar informações específicas que contém (OSHANA, 2005). Os DSPs (*Digital Signal Processor*) são microprocessadores especializados em processamento digital de sinal usados para processar sinais de áudio, vídeo, etc., quer em tempo real quer em *off-line*. A grande característica do DSP é sua alta velocidade de processamento comparada a outros microcontroladores. A capacidade que os DSPs têm de repetir (com uma instrução prévia, por exemplo "RPT") em extrema velocidade uma instrução complexa (como por exemplo a "MPYA": "*Multiply and Accumulate Previous Product*") faz com que sejam rapidamente resolvidas, por exemplo, as Transformadas Rápidas de Fourier (FFT), permitindo a implementação de filtros

digitais. Estes filtros são muito usados em telecomunicações, por exemplo para decodificar a sinalização (DTMF, MFC, etc) (SMITH, 2003).

Em 1980 surgiram os primeiros DSPs completos, o NEC μ PD7720 e o AT&T DSP1. Ambos foram inspirados nas pesquisas de telecomunicações. O primeiro DSP produzido pela Texas Instruments (TI), o TMS32010 apresentado em 1983, provou ser um grande sucesso e a TI é hoje em dia a líder do mercado de DSPs de uso geral. O desenvolvimento da ferramenta RTDX (*Real Time Data Exchange*) foi uma grande inovação da TI para a aplicação em tempo real de DSPs (TI, 2007).

Um DSP pode ser programado tanto em assembly como em alguma linguagem de alto nível. Cada família de DSP tem seu próprio assembly e suas próprias ferramentas de desenvolvimento fornecidas pelo fabricante. Pode-se também utilizar as ferramentas Matlab/Simulink e LabView para a criação de diagramas de blocos e criação de projetos em DSP, desde que um compilador adequado seja utilizado em conjunto.

As aplicações mais usuais em que se empregam DSP são o processamento de áudio e vídeo, no entanto pode ser usado em qualquer outra aplicação que requeira o processamento em tempo real, como controle e automação de dispositivos, demonstrando excelente desempenho em sistemas embarcados (ABUT, 2004).

Os DSPs são projetados levando-se em consideração que as operações mais habituais do processamento digital são as de adição, multiplicação e transferência de memória consecutivos. Para tanto existem instruções de repetição que precedem as instruções acima, tornando possível a execução destas usando-se muitas vezes apenas um ciclo de memória, por ser desnecessária a instrução de retorno (*branch*) para se permanecer no laço (*loop*). Os DSPs têm arquitetura Harvard (barramento de programa e memória separados) e alguns podem operar com números de ponto flutuante (HAYES, 1998).

A escolha de um processador DSP como controlador principal do sistema embarcado proposto deve-se ao fato de, dentre várias características apresentadas, possui uma velocidade de processamento muito alta, satisfazendo as necessidades do projeto de controle, uma vez que o sistema embarcado deve ter a capacidade, dentre outras, da navegação autônoma em ambientes internos e externos. Para tanto, o processador escolhido deve ser rápido o suficiente para tratar todos os dados recebidos pelos sensores, através dos dispositivos de hardware reconfigurável, enviar os comandos aos atuadores, executar os algoritmos de navegação, de triangulação com marcadores, de geração de trajetórias, inteligência artificial, etc. Deve também gerenciar toda comunicação com o sistema supervisor através de links de Wi-

Fi. Desta maneira, optou-se pela utilização de um processador da Texas Instruments da família C6400, mais especificamente, o modelo TMS320C6416 que opera na frequência de 1 GHz executando até 8000 milhões de instruções por segundos (MIPs) na sua performance máxima. O processador TMS320C6416 é baseado na plataforma da família C6400 projetado para aplicações que necessitem alta performance e intensiva utilização da memória. Como exemplo de aplicações típicas que utilizam a família C6400 da TI tem-se implementações de softwares de redes, vídeo, tratamento de imagens, controladores de sistemas embarcados e sistemas multi-canais. A implementação pode ser feita utilizando-se o kit de desenvolvimento da Texas Instruments TMDSDSK6416 DSP *board Kit* (TI, 2007) que contém o processador e todos periféricos necessários para integração do sistema.

3.3 Proposta do sistema embarcado

Segundo Alami et al. (1998) um robô autônomo é precedido de uma arquitetura integrada que permite ao robô planejar suas tarefas. Neste sentido, apresenta-se uma proposta de uma arquitetura aberta para sistemas robóticos móveis ilustrada na Figura 3.1. Os diferentes blocos são implementações tanto de software como de hardware. A arquitetura, sob o ponto de vista de sistemas de robôs móveis, é organizada em vários blocos independentes, conectados através de um barramento de dados, com endereçamento e controle dedicados. Um bloco mestre gerencia vários blocos subordinados. Blocos relacionados às interfaces dos sensores e atuadores, comunicação e memória auxiliar estão sujeitos ao controle direto através do gerenciador de blocos.

A vantagem do uso do barramento comum é a facilidade em ocasionais necessidades de expansão do sistema. Por exemplo, quando limitações de recursos forem superadas, novos blocos podem ser adicionados, permitindo uma configuração adequada do robô para cada tarefa (MELO; ROSÁRIO, 2006).

O ambiente proposto é um conjunto de módulos de hardware e software, implementados com ênfase na utilização de processadores DSPs de última geração a nível de controle de trajetória e de lógica reconfigurável a nível de controle local, integrados para dar suporte ao desenvolvimento de protótipos de sistemas embarcados. A estruturação do ambiente utilizando tecnologias DSP associada a lógica reconfigurável tem por finalidade a utilização das melhores características de processamento em tempo real que se obtém através dos processadores DSPs com as qualidades inerentes dos sistemas de hardware reconfiguráveis.

A lógica reconfigurável, utilizada neste contexto, apresenta as seguintes justificativas (MELO;

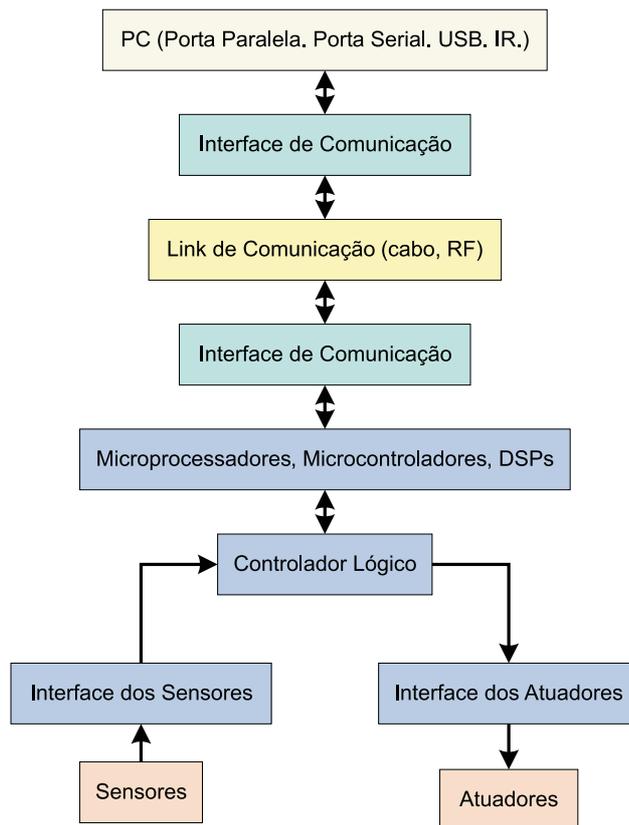


Figura 3.1: Proposta de arquitetura aberta global para sistema robótico móvel.

ROSÁRIO, 2005):

- **Facilidade e rapidez na execução de projeto.** Com a utilização de uma série de ferramentas fornecidas pelo mercado é possível a execução, o teste, a simulação e a depuração de um projeto com bastante eficiência. A divisão do projeto em pequenos blocos funcionais permite a melhor distribuição das tarefas em uma equipe de trabalho. Blocos previamente desenvolvidos podem ser aproveitados em novos projetos, minimizando o tempo de execução dos mesmos.
- **Minimização de custo.** O desenvolvimento de um novo hardware para atender exigências de um novo projeto ou as novas exigências de um projeto antigo, implica em custos que podem ser evitados pela utilização de lógica reconfigurável.
- **Fácil expansão.** Novos módulos podem ser gradativamente adicionados ao sistema original permitindo que novas funcionalidades sejam implementadas sem a necessidade de realizar um novo projeto de hardware.
- **Velocidade de operação.** Além da característica de operar em paralelo, algoritmos implementados com lógica reconfigurável são tipicamente mais rápidos que algoritmos implementados com lógica convencional (execução de linhas de código através de um processador).
- **Domínio tecnológico.** Lógica reconfigurável é apresentada como solução para inúmeros problemas práticos de engenharia. Dentro desta realidade é justificável o esforço para dominar esta área de conhecimento, buscando suas virtudes e limitações.

A utilização de processadores DSPs na estrutura do sistema a nível de controle de trajetória tem, dentre outras, as seguintes justificativas:

- **Velocidade de processamento.** Os processadores DSPs têm velocidade de processamento bem superior aos processadores com outras tecnologias, utilizados com frequência para análise de sinais em tempo real.
- **Rapidez na comunicação interna.** São rápidos o suficiente para tratar todos os dados recebidos pelos sensores, através dos dispositivos de hardwares reconfigurável.
- **Capacidade de processamento.** Têm capacidade de processamento suficiente para executar os algoritmos de navegação, de triangulação com marcadores, de geração de trajetórias, inteligência artificial, etc.

São objetivos do ambiente proposto:

- **Flexibilidade.** O ambiente deve ser capaz de adaptar-se facilmente à aplicação desejada. Um novo conjunto de requisitos deve ser facilmente atendido sem implicar em novos projetos de hardware.
- **Expansibilidade.** O ambiente deve ser de fácil expansão, permitindo que novas funcionalidades sejam adicionadas a um sistema pré-existente.
- **Transparência.** O ambiente deve ser totalmente aberto, permitindo a acesso de novos usuários a todos os aspectos do mesmo. Isto vem de encontro as diferentes necessidades acadêmicas de graduação e pós-graduação a que este ambiente visa prestar suporte.
- **Adequação a tarefa.** O ambiente deve poder ser configurado para resolver problemas de maior ou menor complexidade. Isto permite a utilização adequada de recursos para resolver um determinado problema, evitando o desperdício de recursos.
- **Baixo Custo.** São considerados aspectos que visem a minimização do custo no desenvolvimento de um protótipo de sistemas embarcados. São utilizadas ferramentas de projeto de baixo custo ou sem custo associado. A própria utilização de lógica reconfigurável permite a minimização do custo total de um projeto por razões já apresentadas anteriormente.
- **Capacidade de comando e monitoração remota.** Em muitas aplicações de sistemas embarcados a monitoração e modificação de parâmetros internos aos sistemas pode ser de grande utilidade, permitindo um menor tempo de projeto e depuração de erros ou atendendo a pré-requisitos deste projeto.
- **Estrutura para a prototipagem rápida.** Estar estruturado e ser capaz de se comunicar com os sistemas de simulação, como por exemplo o MatLab, para execução das técnicas de prototipagem rápida e de HIL (*Hardware In-the-Loop*).

O sistema proposto pode ser visualizado como um sistema multi-níveis, dividido em três níveis de controle hierárquicos, conforme ilustrado na Figura 3.2.

3.3.1 Descrição dos níveis de controle

Controle de supervisão. Neste nível de controle mais alto, a supervisão de um ou mais robôs móveis podem ser conduzidas, através da execução de estratégias globais de

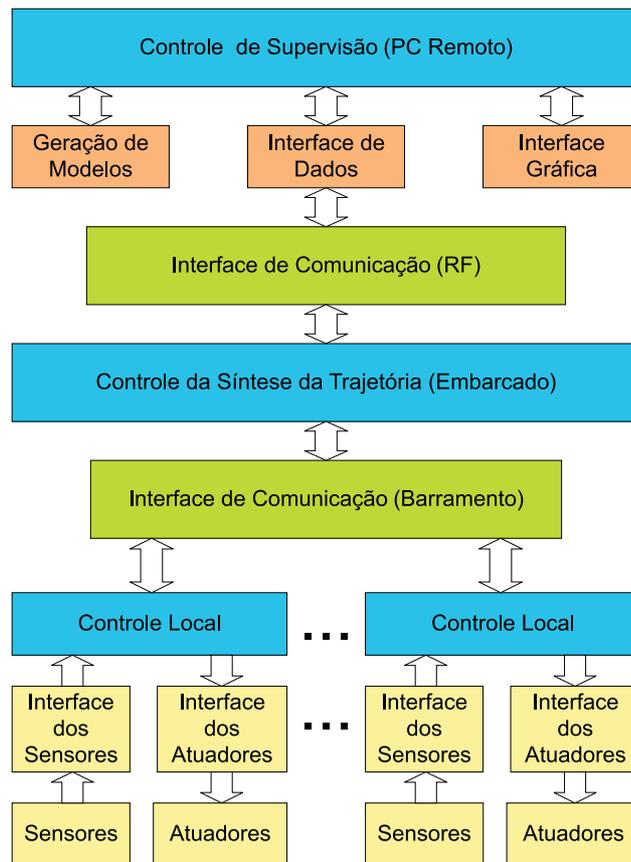


Figura 3.2: Diagrama de blocos ilustrando os níveis de controle para o sistema robótico móvel.

controle. Este nível também permite o gerenciamento das tarefas do robô móvel de tal forma a estabelecer correções nas conduções das tarefas de acordo com as informações recebidas dos sensores, ou modificar, por exemplo, a informação requisitada pelo nível de controle local, como será mencionada adiante.

Controle da síntese da trajetória. Neste nível, o software embutido do robô móvel processa o controle. A estratégia de controle permite que decisões sejam tomadas em nível local, com ocasionais correções feitas pelo nível de controle mais alto, o controle de supervisão. Sem comunicação com o nível de controle de supervisão, o robô móvel somente realiza um conjunto de ações baseadas em dados obtidos pelos sensores locais e em informações previamente armazenadas na memória. A transferência de dados entre este nível e o nível de supervisão é conseguida através do uso de estruturas de dados implementada numa memória compartilhada, que permite um melhor tratamento dos dados transmitidos e recebidos. Através do uso do mesmo recurso da memória compartilhada, as tarefas deste nível podem ser divididas entre vários diferentes processadores.

Controle local. Este nível é restrito às estratégias de controle local associadas com a interface dos sensores e atuadores. Nesta etapa é sugerida a estratégia de implementação em hardware do sistema de controle com utilização de sistemas de hardware reconfigurável (PLDs e FPGAs). Os controladores locais podem ser implementados sob vários tipos de equações, uma delas o formato RST que tem uma estrutura que pode ser aplicável em ambientes de arquitetura abertas, incluindo os controladores PID clássicos assim como mais avançadas técnicas de controle, tais como os sistemas de controle preditivos, por exemplo, como descritos por Boucher e Dumur (1998). A implementação de um controlador PID que atua a nível de controle local é apresentada do capítulo 4 assim como a proposta do simulador virtual que é apresentada no capítulo 5. Este simulador atua na estratégia de controle local facilitando a otimização do sistema de controle embarcado.

3.3.2 Comunicação remota

Um dos aspectos necessários à implementação da arquitetura proposta é a comunicação remota ou operação remota entre o nível de controle supervisor ou remoto e nível de controle embarcado, ou seja, a capacidade de realizar operações remotas sobre os sistemas embarcados considerados. Esta funcionalidade pode ser implementada através de rádio frequência (RF) através de um link *WiFi*, infravermelho (IR), fibra ótica ou fio metálico. Na Figura 3.3 é

apresentado um exemplo onde é considerado o uso de RF para implementar a comunicação remota entre o nível de controle supervisor e nível de controle embarcado.

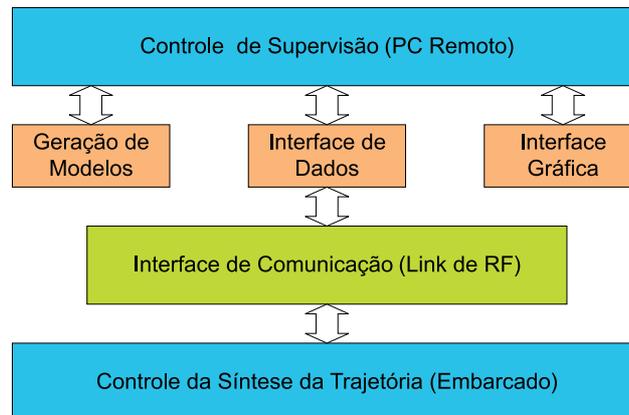


Figura 3.3: Diagrama de blocos mostrando a interligação da camada de controle supervisor e a camada de controle embarcado através de um link de rádio frequência.

A operação remota tem por função a comunicação bi-direcional entre o sistema embarcado (nível de controle embarcado) e um computador remoto (nível de controle supervisor), este link de dados *wireless* e também o link de RF para o comando manual do robô móvel são ilustrados na Figura 3.4. Maiores detalhes destes sistemas de comunicação remoto é apresentado na seção 3.3.4

3.3.3 Descrição da estrutura hierárquica

Nesta aplicação, os blocos de controle locais, as interfaces dos sensores e as interfaces de potência serão implementados através da utilização de PLDs ou FPGAs. A Figura 3.5 ilustra um descrição dos blocos que serão implementados considerando-se a arquitetura hierárquica e estruturada proposta neste projeto. É analisado a seguir, com mais detalhes, a estrutura de cada nível.

Nível de supervisão. O nível de supervisão deverá ser implementado utilizando-se de uma GUI (*Graphical User Interface*) num sistema Windows com um microcomputador PC. O link de comunicação entre este nível e o nível de síntese da trajetória utilizará a tecnologia de comunicação sem fio por RF (*wireless*) usada em redes Ethernet com link Wi-Fi. Wi-Fi é uma marca licenciada originalmente pela Wi-Fi Alliance para descrever a tecnologia de redes sem fio embarcadas (WLAN) baseadas no padrão IEEE 802.11. Este padrão de comunicação sem fio é utilizado com o protocolo TCP/IP e garante

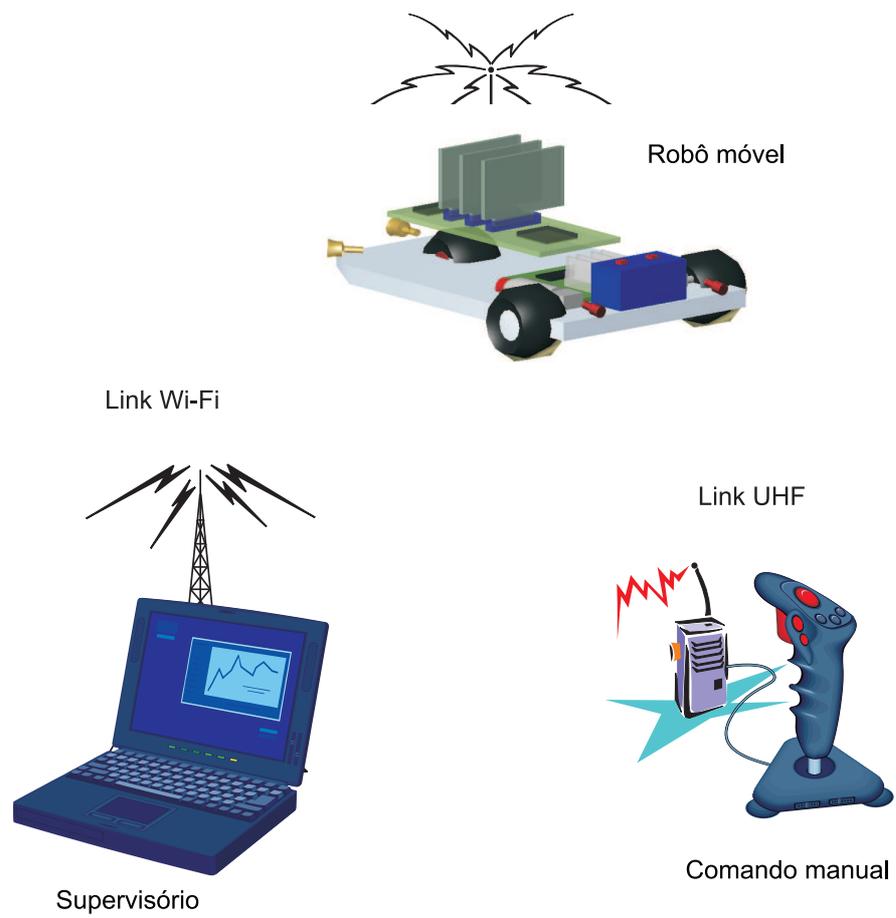


Figura 3.4: Ilustração das comunicações do módulo supervisor e do comando manual.

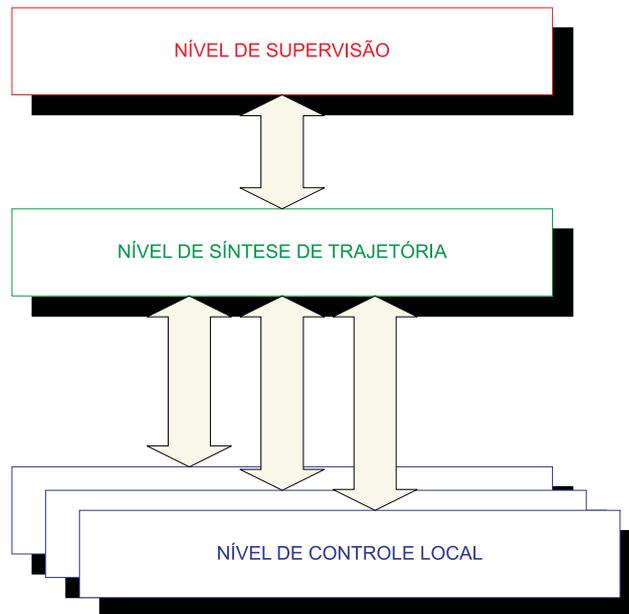


Figura 3.5: Diferentes níveis de controle do sistema proposto.

uma taxa de transmissão de até 11 Mbps operando na frequência de 2,4 GHz e requer poucos acessórios para sua implementação. Hoje em dia, adquire-se placa de interface Wi-Fi a um custo bem acessível haja vista sua popularização na utilização em *Lap Tops* e microcomputadores. Os 14 canais disponíveis do padrão Wi-Fi em 2,4 GHz garantem uma expansibilidade no sistema com a implementação de estratégias de controle de múltiplos robôs.

Nível de síntese da trajetória. Este nível inclui uma placa de controle da Texas Instruments TMDSDSK6416 DSP *board Kit* (TI, 2007) que contém o processador DSP TMS320C6416 e todos periféricos necessários para integração do sistema. Esta placa utiliza um sistema de operação em tempo real, que será utilizado para aplicações no sistema embarcado. Através de rotinas em linguagem padrão ANSI C, compatível com o processador, a execução e correção da trajetória associada com outras tarefas definidas pelo nível de supervisão deverão ser realizadas com sucesso.

Nível de controle local. Como mencionado previamente, este nível inclui todas as interfaces necessárias com os sensores e atuadores, e o controlador de cada motor. Num ponto de vista de uma arquitetura aberta, a implementação destes controladores locais têm que ser gerais o bastante para permitirem uma fácil reconfiguração da lei de controle de acordo com as especificações do usuário.

Para concluir, considere a Figura 3.6 onde são apresentados detalhes dos blocos que serão implementados em *hardware* e *software*, distribuídos de acordo com os níveis de controle propostos. O interesse por um projeto modular é óbvio considerando-se um *framework* para uma arquitetura aberta. Até as leis de controle locais permanecem abertas.

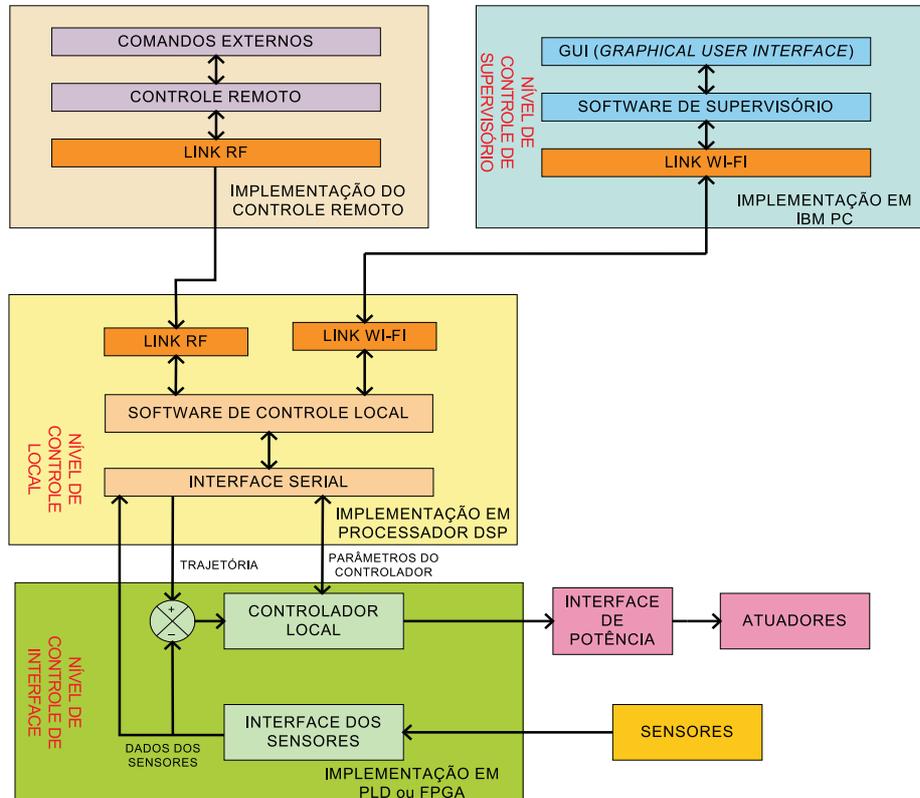


Figura 3.6: Representação das estruturas de níveis de controle do robô móvel.

3.3.4 Detalhes da implementação do hardware

A arquitetura, do ponto de vista do robô móvel, esta organizada em vários blocos independentes, conectados através de um barramento serial local, conforme ilustrado na Figura 3.7.

O bloco principal de gerenciamento comanda vários blocos escravos. Os blocos associados com a interface dos sensores, atuadores, comunicação e memória auxiliar estão sujeitos ao controle direto do bloco de gerenciamento. Uma das vantagens da utilização de um barramento comum é a facilidade de expansão do sistema. Dentro das limitações de pesquisas, é possível adicionar blocos, permitindo uma configuração adaptada do robô para cada diferente tarefa. A seguir, são descritos as funcionalidades de cada bloco.

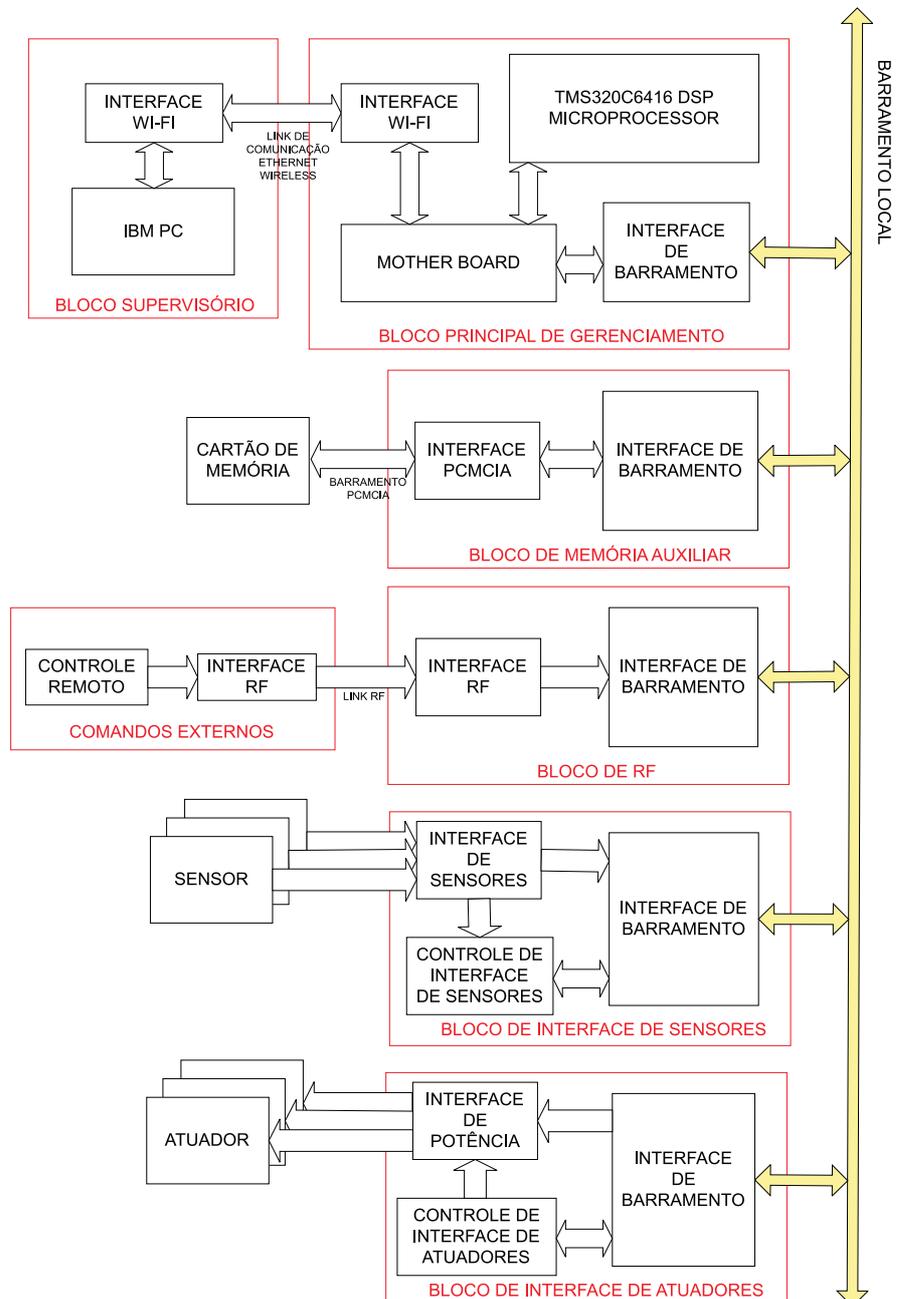


Figura 3.7: Ilustração da arquitetura de hardware do sistema proposto.

Bloco supervisorio. É implementado um microcomputador PC e se comunica com o robô móvel através de um link *wireless* Wi-Fi. A implementação de uma GUI (*Graphical User Interface*) num sistema Windows pode ser feita utilizando-se o software LabView da National.

Bloco principal de gerenciamento. É implementado em hardware com a utilização do processador DSP da família C6400 da Texas Instruments, conforme descrição do nível de síntese de trajetória na seção 3.3.3. A comunicação com o bloco supervisorio é feita através de link de comunicação *wireless* utilizando-se padrão Wi-Fi, conforme detalhado na seção 3.3.3. A comunicação com os demais blocos do sistema embarcado é feita através de um barramento serial de alto desempenho, que pode ser implementado utilizando-se o padrão Firewire (norma IEEE 1394). Sua capacidade de comunicação pode atingir até 30 vezes a velocidade do USB (*Universal Serial Bus*). Possui uma interface simples capaz de receber até 63 dispositivos. Os sistemas Firewire atuais podem operar a uma taxa de 400 Mbps (50 MB/s), podendo operar a 800 Mbps (100 MB/s) sob a nova especificação IEEE 1394b. A ligação original (IEEE 1394-1995) é constituída por 6 condutores, 4 dos quais agrupados em 2 pares cruzados, 1 para transmissão de dados em modo half-duplex e outro para o sinal de relógio. Os restantes 2 são usados para a alimentação dos dispositivos a que se encontram ligados.

Bloco de memória auxiliar. Armazena as informações dos sensores e tem a funcionalidade de uma biblioteca para as possíveis estratégias de controle para os sensores e atuadores. Pode ser utilizado como opção de registro de erros. Uma opção de implementação é a utilização da interface PCMCIA, pois é facilmente encontrada no mercado, e pode ser bem adaptada para aplicações em robôs móveis, devido ao baixo consumo, pouco peso, dimensões reduzidas e alta capacidade de armazenamento assim como boa imunidade a vibrações mecânicas. O padrão PCMCIA surgiu em 1990 como um padrão para a expansão de memória em notebooks. A idéia era permitir a instalação de memória RAM adicional sem precisar abrir o notebook e instalar novos módulos o que, na maioria dos modelos da época, era bem mais complicado do que hoje em dia.

Bloco de RF. Permite o estabelecimento de um canal simplex de comunicação *wireless* através de um link de RF. É utilizado para que o controlador do robô móvel receba comandos, permitindo o aprendizado de trajetórias, por exemplo. Opera em paralelo com o link Wi-Fi utilizado na comunicação do bloco supervisorio com o bloco de gerenciamento. O objeto deste link de comunicação é permitir o uso de um controle re-

moto. Todos os comandos executados pelo controle remoto são analisados pelo bloco de gerenciamento principal para garantir a estratégia de melhor trajetória. O controle remoto tem prioridade na execução da trajetória em relação outros blocos, como por exemplo o bloco supervisor, e pode assumir o controle do robô para executar, por exemplo, movimentos de paradas emergenciais. A implementação deste bloco pode ser feita utilizando-se módulos *transceivers* UHF (*Ultra High Frequency*) de baixa potência, como por exemplo o módulo BIM-433-40 do fabricante Radiometrix (RADIOMETRIX, 2007). Este é um módulo de rádio UHF miniatura capaz de transmitir dados em canal *half duplex* com taxa de até 40 kbps através de distâncias de até 30 metros no interior de construções e de até 120 metros ao ar livre. A opção de escolha deste dispositivo se justifica pois integra um transmissor de FM na faixa de UHF com um receptor super heterodino com reconstrução de dados e mudança de TX/RX no circuito que se traduz numa solução de baixo custo e alto desempenho na implementação de um link de radio bidirecional de curta distância. A Figura 3.8 ilustra este dispositivo.

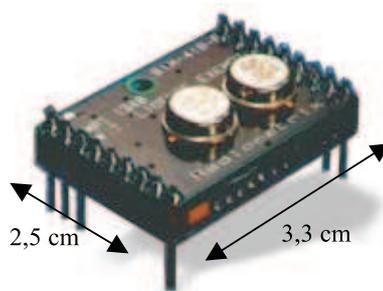


Figura 3.8: Módulo de RF BiM-433 usado para implementação da comunicação remota.

Bloco de interface de sensores. Implementado em hardware reconfigurável, PLD ou FPGA, este módulo faz a captação de todos os dados transmitidos pelos diversos sensores do robô móvel, tais como odômetros, sensores de ultra-som, sensores laser, etc. Os dados são separados, classificados e enviados ao bloco principal de gerenciamento que utiliza estes dados para montagem da melhor estratégia de trajetória.

Bloco de interface de atuadores. A nível de controle local, este bloco pode ser implementado em PLD ou FPGA para atuar, junto com o bloco de interface de sensores, no sentido da implementação de um controlador, como por exemplo o PID. Este controlador pode ser executado também em software no processador DSP do bloco de gerenciamento e, neste caso, o bloco de interface de atuadores serviria de intermediário entre o comando central e os atuadores, analisando e executando os comandos recebidos.

A idéia da implementação do controlador em hardware utilizando-se PLDs ou FPGAs é liberar o processador principal desta tarefa de controle de baixo nível deixando esta responsabilidade aos controladores locais concentrando sua capacidade de processamento em tarefas de alto nível, tais como as estratégias de geração de trajetória, execução de rotinas de navegação autônoma, algoritmos de inteligência artificial, gerenciamento da memória, comunicação com o módulo supervisor, etc.

A Figura 3.9 ilustra a implementação física no robô móvel do sistema proposto, podendo-se observar o *lay-out* dos dispositivos mecânicos, elétricos e eletrônicos.

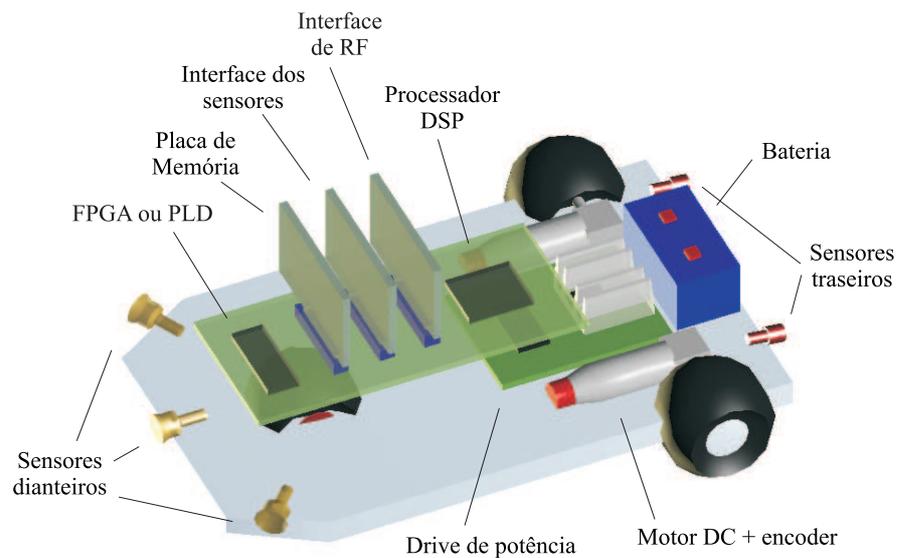


Figura 3.9: Implementação física da arquitetura proposta no robô móvel.

Um exemplo da flexibilidade obtida pelos PLDs e FPGAs é ilustrada através do controle de posição e velocidade do eixo de tração do robô móvel, cujo alvo é o controle de um motor DC acionado por um *drive* PWM. Este controle programável trabalha diretamente com sinais digitais gerados pelo encoder, que está acoplado ao eixo do motor, e pelo nível de controle local, representando a trajetória (MELO et al., 2005). Uma ilustração deste sistema é apresentado na Figura 3.10.

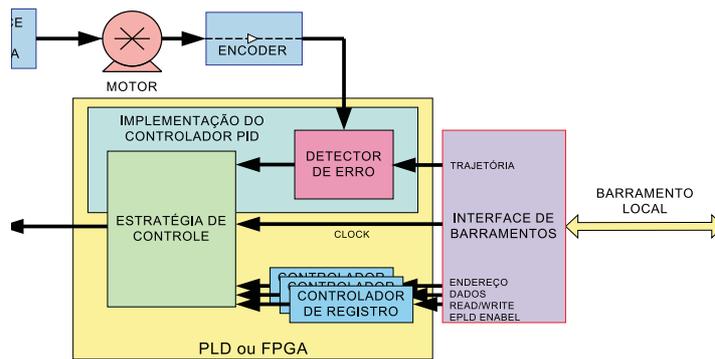


Figura 3.10: Blocos de controle do motor DC do robô móvel.

3.4 Considerações finais

Neste capítulo foi descrita uma proposta de arquitetura aberta de supervisão e controle embarcada, apresentando uma visão geral do ambiente proposto. Procurou-se enfatizar os aspectos conceituais do ambiente, bem como aspectos de implementação de hardware que podem ser úteis para execuções práticas. Contudo, os blocos desenvolvidos podem (e devem) ser modificados, pois um dos objetivos do sistema é que seja aberto a modificações de seus blocos, permitindo o aprimoramento e a evolução do mesmo. O sistema proposto é utilizado na técnica de prototipagem rápida e HIL (*hardware-in-the-loop*) que é descrito no capítulo 5. Este modelo de arquitetura aberta apresentado aqui é também utilizado como base para o desenvolvimento do simulador cinemático e dinâmico do robô móvel, que é mostrado na seção 5.3, onde os elementos básicos desenvolvidos neste capítulo serão utilizados e expandidos para implementação de sistemas práticos. Na apresentação destes projetos práticos, os novos elementos adicionados ao sistema, demonstrarão a necessidade de um ambiente de apoio a desenvolvimento de sistemas embarcados ser o mais aberto e flexível possível.

Capítulo 4

Implementação de um simulador de movimento com ênfase na dinâmica e controle

Este capítulo tem como objetivo apresentar as implementações dos modelos cinemático e dinâmico a serem utilizados no simulador do robô móvel e na técnica de prototipagem rápida, seções 5.3 e 5.5 respectivamente. As equações matemáticas do modelo cinemático adotado foram desenvolvidas na seção 2.2 que têm como plataforma o sistema robótico móvel não holonômico utilizado como base no desenvolvimento dos modelos cinemáticos e dinâmicos, assim como na implementação do simulador robótico a ser visto na seção 5.3. É apresentado também aspectos relacionados à implementação do controlador PID, utilizado nos controles dos dois eixos, assim como os detalhes de sintonia. A escolha do controlador PID recai principalmente em dois fatores básicos: primeiro, o controle das rodas de tração do sistema robótico é relativamente simples e segundo, a facilidade de implementação em software e hardware deste tipo de controlador. Nada impede que, como estamos tratando de um sistema de arquitetura aberta, conforme descrito no capítulo 3, venham a ser empregados outros tipos de controladores, dependendo da aplicação e da necessidade. Isto porque, todo o sistema foi concebido para ser o mais genérico possível, podendo-se adaptar, com poucas modificações, a qualquer necessidade implementacional.

4.1 Modelagem matemática

As tarefas realizadas pelos robôs móveis baseiam-se no movimento independente de cada grau de liberdade, coordenados a partir de um planejador de trajetórias baseado no seu modelo cinemático. Na maioria dos casos, a programação das tarefas é planejada com antecipação e um mapa do ambiente carregado na memória do robô. O robô móvel realiza a

trajetória com seqüência de movimentos independentes de cada junta, até alcançar a posição final desejada. A partir do conhecimento destas posições articulares, é facilmente implementado um gerador de referências (perfil de velocidades) baseado nas características cinemáticas das juntas.

Para realização de tarefas em nível de sistema de coordenadas (sistema cartesiano) e para geração dos sinais de referência para o controlador de posição de cada junta robótica do sistema mecatrônico em estudo, torna-se necessário o estabelecimento de um modelo matemático baseado na cinemática do sistema. Portanto, o controle de um robô necessita de procedimentos para transformar os dados de referência de posicionamento, tais como a velocidade linear e o raio de curvatura, em coordenadas cartesianas, quando se deseja realizar o controle através de um referencial cartesiano. A Figura 4.1 ilustra a estrutura de controle do robô móvel com os blocos representativos da geração de trajetória, modelo dinâmico e cinemático do sistema.

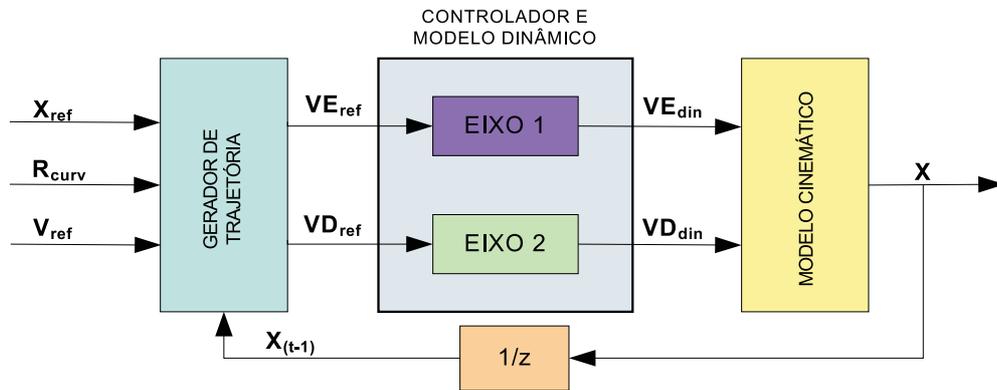


Figura 4.1: Estrutura de controle do robô móvel.

O gerador de trajetória recebe os dados de referência, tais como o vetor posicionamento $\mathbf{X}_{ref} = [x_{ref}, y_{ref}, \theta_{ref}]$, a velocidade linear de referência do robô V_{ref} e o raio de trajetória instantâneo do robô R_{curv} , que são convertidos em VE_{ref} (velocidade linear da roda esquerda) e VD_{ref} (velocidade linear da roda direita). Estas velocidades diferenciadas são recebidas pelo controlador, e no modelo dinâmico do sistema, são enviadas às respectivas rodas do robô, através de seus atuadores. São então gerados pelo controlador os vetores VE_{din} (velocidade linear dinâmica da roda esquerda) e VD_{din} (velocidade linear dinâmica da roda direita). No bloco do modelo cinemático, estes dados são convertidos no vetor posicionamento final do robô $\mathbf{X} = [x, y, \theta]$.

4.1.1 Modelo cinemático

O modelo cinemático implementado no controlador, para movimentos circulares do robô, é o descrito na seção 2.3.1 que resulta em (equação 2.39)

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega\delta t) & -\text{sen}(\omega\delta t) & 0 \\ \text{sen}(\omega\delta t) & \cos(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - \text{CCI}_x \\ y - \text{CCI}_y \\ \theta \end{bmatrix} + \begin{bmatrix} \text{CCI}_x \\ \text{CCI}_y \\ \omega\delta t \end{bmatrix}. \quad (4.1)$$

A equação 4.1 descreve o movimento rotacional do robô a uma distância R do CCI com um velocidade angular dada por ω .

Quando o robô está se deslocando em linha reta, então $v_e = v_d = v$, com $R \rightarrow \infty$. O modelo cinemático implementado no controlador, conforme descrito na seção 2.3.2, é dado por

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x + v \cos(\theta)\delta t \\ y + v \text{sen}(\theta)\delta t \\ \theta \end{pmatrix}. \quad (4.2)$$

4.1.2 Modelagem dinâmica do sistema

O modelo matemático de um sistema dinâmico é definido como um conjunto de equações que representa com precisão ou, pelo menos, razoavelmente bem a dinâmica do sistema. Uma vez obtido o modelo matemático de um sistema, podem ser utilizadas várias ferramentas analíticas e de computação para efeito de análise e síntese. A seguir é apresentada a modelagem dinâmica do atuador utilizado neste trabalho.

4.1.2.1 Modelo do atuador

Cada eixo de tração, localizado nas rodas esquerda e direita do robô móvel, inclui um motor CC, um redutor e um *encoder* de posição. Considerando as equações de modelagem

do motor CC, tem-se que

$$\begin{aligned}
 e_a(t) &= L_a \frac{di(t)}{dt} + R_a i(t) + K_b \frac{d\theta_m(t)}{dt}, \\
 T_m(t) &= J_m \frac{d^2\theta_m(t)}{dt^2} + B_m \frac{d\theta_m(t)}{dt}, \\
 T_m(t) &= K_t i(t),
 \end{aligned}
 \tag{4.3}$$

onde $e_a(t)$ é a tensão de alimentação da armadura, $T_m(t)$ o torque do motor, $\theta_m(t)$ a posição angular do rotor do motor, $i(t)$ a corrente de armadura do motor, L_a a indutância e R_a a resistência elétrica da bobina do motor, B_m é o coeficiente de atrito viscoso, K_b é a constante de força contra-eletromotriz, K_t a constante de torque e J_m a inércia do eixo do motor. A Figura 4.2 mostra o esquema equivalente dos motores CC utilizados para o tracionamento das rodas do robô móvel.

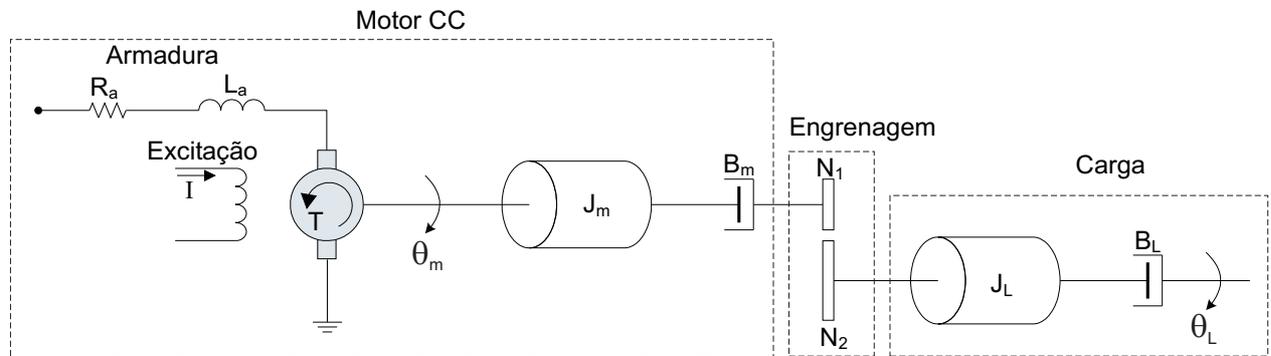


Figura 4.2: Esquema equivalente do motor CC com acionamento da carga.

O diagrama de blocos da Figura 4.3 ilustra o modelamento do motor de CC com o sistema redutor.

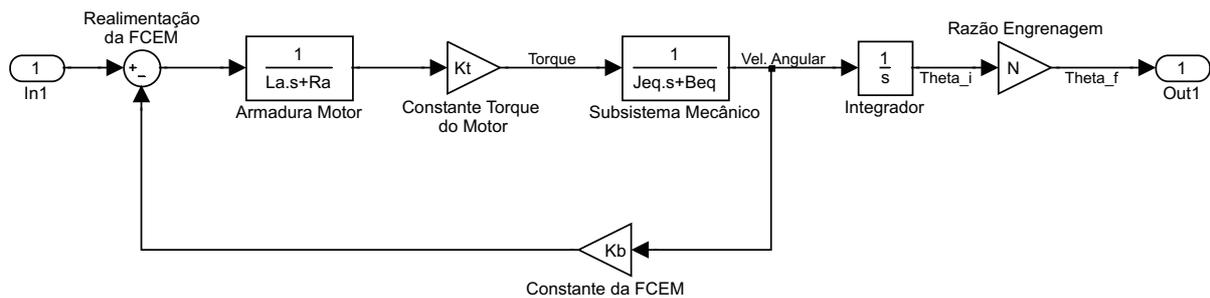


Figura 4.3: Modelamento do motor CC.

A função de transferência do motor CC em vazio utilizado no protótipo do robô móvel, e para validação e testes, é dada por:

$$\frac{\theta_m(s)}{E_a(s)} = \frac{K_t}{L_a J_m s^3 + (R_a J_m + L_a B_m) s^2 + (R_a B_m + K_b K_t) s} \quad (4.4)$$

Com o sistema redutor, temos que

$$J_{eq} = J_m + J_L \left(\frac{N_1}{N_2} \right)^2, \quad (4.5)$$

$$B_{eq} = B_m + B_L \left(\frac{N_1}{N_2} \right)^2, \quad (4.6)$$

onde $\frac{N_1}{N_2} = N$ é a razão da engrenagem do redutor, B_{eq} é o coeficiente de atrito viscoso equivalente do sistema motor-carga e J_{eq} a inércia equivalente. A função de transferência do motor com a carga é dada por:

$$\frac{\theta_L(s)}{E_a(s)} = \frac{K_t}{L_a J_{eq} s^3 + (R_a J_{eq} + L_a B_{eq}) s^2 + (R_a B_{eq} + K_b K_t) s} \quad (4.7)$$

Os dois motores CC utilizados no protótipo do robô móvel são iguais e têm os parâmetros apresentados na Tabela 4.1.

Tabela 4.1: Parâmetros dos motores CC.

Parâmetro	Variável	Valor
Resistência da armadura	R_a	2Ω
Indutância da armadura	L_a	180 mH
Constante de torque	K_t	$1 \times 10^{-6} \text{ N.m/A}$
Constante da FCEM	K_b	56,5 V.(rad/s)
Momento de inércia	J_m	$1 \times 10^{-4} \text{ kg.m}^2$
Atrito viscoso	B_m	$5 \times 10^{-5} \text{ N.m/(rad/s)}$
Velocidade angular máxima	ω_{max}	300 rad/s
Relação de transmissão	N	1/10
Momento de inércia da carga	J_L	$1 \times 10^{-2} \text{ kg.m}^2$
Atrito viscoso da carga	B_L	$1,15 \times 10^{-2} \text{ N.m/(rad/s)}$

Aplicando-se os valores da Tabela 4.1 na equação 4.4, resulta na função de transferência do motor CC em vazio:

$$\frac{\theta_m(s)}{E_a(s)} = \frac{1 \times 10^{-5}}{1,8 \times 10^{-5}s^3 + 2,09 \times 10^{-4}s^2 + 6,65 \times 10^{-4}s}, \quad (4.8)$$

e, com a carga, a equação 4.7 resulta em

$$\frac{\theta_m(s)}{E_a(s)} = \frac{1 \times 10^{-5}}{3,6 \times 10^{-5}s^3 + 4,297 \times 10^{-4}s^2 + 8,95 \times 10^{-4}s}. \quad (4.9)$$

A equação 4.9 é a função de transferência da planta sendo utilizada no desenvolvimento do controlador.

A Figura 4.4 ilustra a rotina de inicialização dos parâmetros dos atuadores. Esta rotina é utilizada para inicializar o sistema simulador do robô móvel que é apresentado no capítulo 5.3. O anexo D mostra a listagem completa com todos os parâmetros e características dos atuadores utilizados no simulador.

4.2 Controlador PID

Para controlar a velocidade angular ω dos motores CC e, conseqüentemente, a velocidade e o deslocamento linear de cada roda, foi implementado um controlador PID. A Figura 4.5 mostra um diagrama de blocos com a planta e o controlador PID atuando sobre o motor CC.

4.2.1 Ajustes de parâmetros para controladores PID

A Figura 4.6 mostra o controlador PID de uma planta. Se um modelo matemático da planta pode ser obtido, então é possível aplicar várias técnicas de projeto na determinação dos parâmetros do controlador que vão impor as especificações do regime transitório e do regime permanente do sistema de malha fechada (OGATA, 2003).

O processo de selecionar parâmetros do controlador que garantam uma dada especificação de desempenho é conhecido como sintonia do controlador. Ziegler e Nichols (1942) sugeriram regras para a sintonia de controladores PID (o que significa ajustar os valores de K_p , T_i e T_d) baseadas na resposta experimental ao degrau ou no valor de K_p que resulta em uma estabilidade marginal, quando somente uma ação proporcional é utilizada.

```

Editor - C:\Leo\work\inicializacao_3.m
File Edit Text Cell Tools Debug Desktop Window Help
D [Icons] Stack: Base [Icons] [Icons] [Icons]
1 %Inicializa os parâmetros dos motore DC e do controlador PID
2
3 %Motor 1 --> Roda Direita
4 %Motor 2 --> Roda Esquerda
5
6 disp('Inicialização');
7
8 c1c;
9 %close all;
10 %clear all;
11
12 %Raio das rodas de tração do robô
13 Rd = 0.04; %m
14
15 %Comprimento do eixo das rodas de tração do robô
16 L = 0.387; %m
17
18 %Posicionamento inicial do robô
19 %X_inic = 2; %m
20 %Y_inic = 3; %m
21 %Teta_inic = pi/2; %rad
22
23 %Parametros dos motores
24 Ra = [2 2]; % Resistência da bobina em Ohm
25 La = [0.18 0.18]; % Indutância da bobina em Henry
26
27 %Constante de torque do motor
28 Kt = [0.00001 0.00001]; % N.m/A
29
30 %Constante da FCEM
31 Kb = [56.5 56.5]; % V.(rad/s)
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
script Ln 1 Col 1 OVR

```

Figura 4.4: Exemplo da rotina de inicialização dos parâmetros dos atuadores.

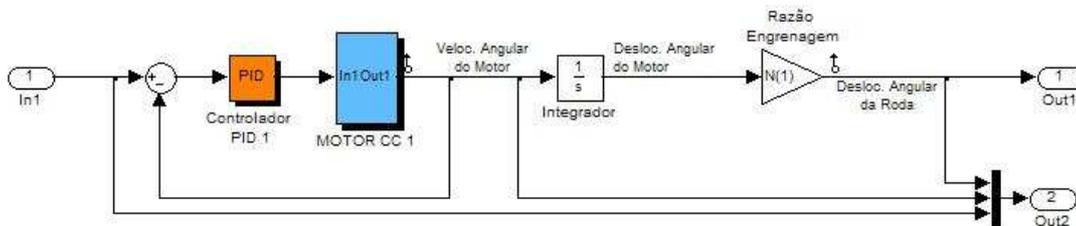


Figura 4.5: Controlador PID do motor CC.

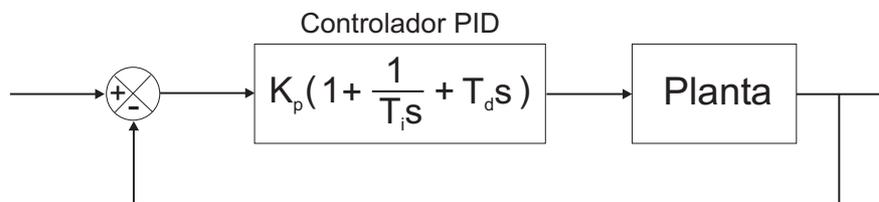


Figura 4.6: Controlador PID de uma planta genérica.

4.2.1.1 Primeiro método de Ziegler-Nichols para sintonia de controladores PID

No primeiro método, obtém-se experimentalmente a resposta da planta a uma entrada em degrau unitário. Se a planta não possui integradores nem pólos complexos conjugados dominantes, então essa curva de resposta ao degrau unitário pode ter o aspecto de um “S”, como mostrado na Figura 4.7.

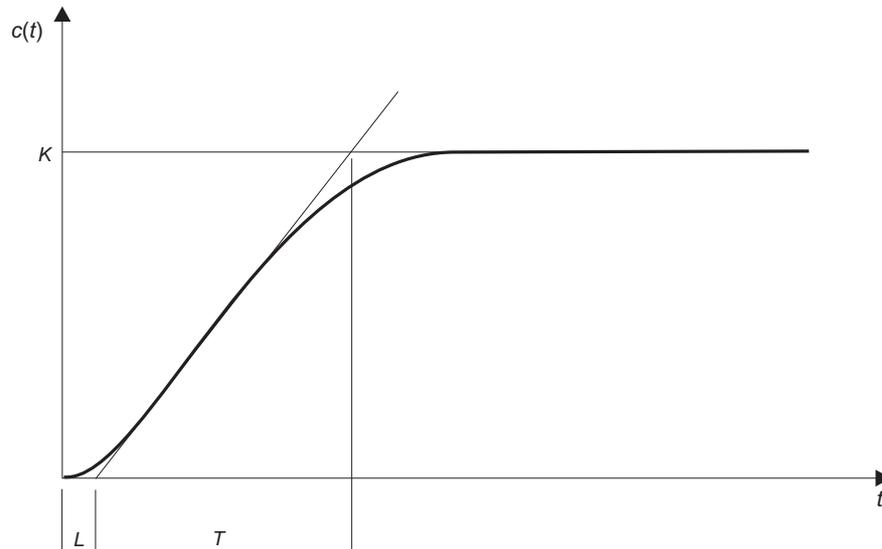


Figura 4.7: Curva de resposta ao degrau.

A curva com o formato “S” pode ser caracterizada por duas constantes, o atraso L e a constante de tempo T . O atraso e a constante de tempo são determinados desenhando-se uma linha tangente no ponto de inflexão da curva e determinando-se a intersecção da linha tangente com o eixo dos tempos e a linha $c(t) = K$, conforme mostra a Figura 4.7. A função de transferência $\frac{C(s)}{U(s)}$ pode ser aproximada por um sistema de primeira ordem com um atraso de transporte, como segue:

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts + 1} \quad (4.10)$$

Ziegler e Nichols (1942) sugeriram escolher os valores de K_p , T_i e T_d de acordo com a fórmula que aparece na Tabela 4.2

O controlador PID sintonizado pelo primeiro método das regras de Ziegles-Nichols fornece:

Tabela 4.2: Regra de sintonia do controlador PID baseada na resposta ao degrau.

Controlador	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0,9\frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2\frac{T}{L}$	$2L$	$0,5L$

$$\begin{aligned}
 G_c(s) &= K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \\
 &= 1,2\frac{T}{L} \left(1 + \frac{1}{2Ls} + 0,5Ls \right) \\
 &= 0,6T \frac{(s + \frac{1}{L})^2}{s}.
 \end{aligned} \tag{4.11}$$

Portanto, o controlador PID tem um pólo na origem e zeros duplos em $s = -\frac{1}{L}$.

4.2.1.2 Segundo método de Ziegler-Nichols para sintonia de controladores PID

No segundo método, define-se primeiro $T_i = \infty$ e $T_d = 0$. Utilizando somente a ação do controle proporcional, aumenta-se K_p de 0 ao valor crítico K_{cr} , no qual a saída exibe uma oscilação sustentada pela primeira vez. Portanto, o ganho crítico K_{cr} e o correspondente período P_{cr} são determinados experimentalmente. A Figura 4.8 ilustra o resultado obtido através da simulação da planta do robô móvel aplicando-se este segundo método.

Ziegler e Nichols (1942) sugerem escolher os valores dos parâmetros K_p , T_i e T_d de acordo com a fórmula mostrada na Tabela 4.3.

Nota-se que o controlador PID sintonizado por este segundo método fornece:

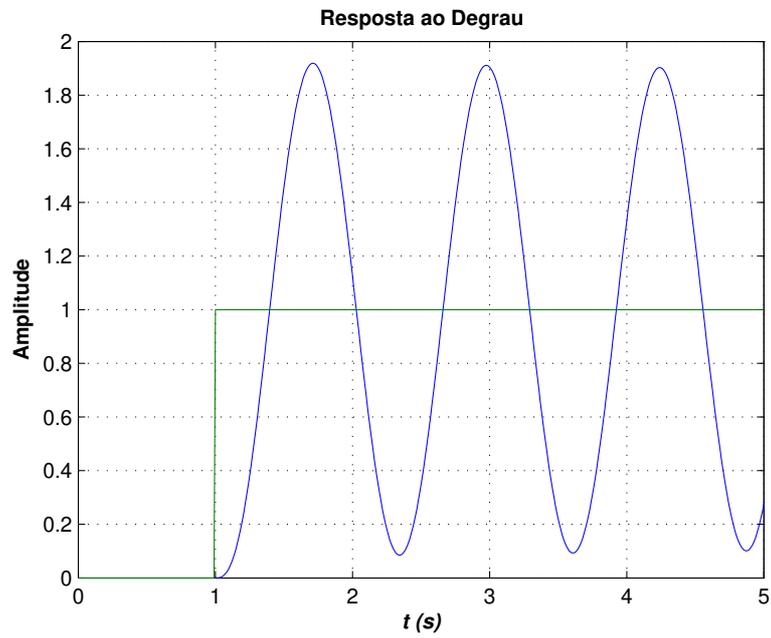


Figura 4.8: Oscilação sustentada com período P_{cr} .

Tabela 4.3: Regra de sintonia do controlador PID baseada no ganho crítico.

Controlador	K_p	T_i	T_d
P	$0,5K_{cr}$	∞	0
PI	$0,45K_{cr}$	$\frac{1}{1,2}P_{cr}$	0
PID	$0,6K_{cr}$	$0,5P_{cr}$	$0,125P_{cr}$

$$\begin{aligned}
G_c(s) &= K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \\
&= 0,6 \frac{T}{L} \left(1 + \frac{1}{0,5 P_{cr} s} + 0,125 P_{cr} s \right) \\
&= 0,075 K_{cr} P_{cr} \frac{\left(s + \frac{4}{P_{cr}} \right)^2}{s}.
\end{aligned} \tag{4.12}$$

Portanto, o controlador PID tem um pólo na origem e zeros duplos em $s = -\frac{4}{P_{cr}}$. Nota-se que, como a planta tem função de transferência conhecida (equação 4.9), então pode-se utilizar o método do lugar das raízes para se encontrar o ganho crítico K_{cr} e a frequência de oscilações sustentadas ω_{cr} , onde $\frac{2\pi}{\omega_{cr}} = P_{cr}$. Esses valores podem ser encontrados a partir dos pontos de cruzamento dos ramos do lugar das raízes com o eixo $j\omega$.

4.2.2 Determinação das constantes do controlador PID

Para a determinação das constantes do controlador PID a ser adotado no sistema, optou-se pela utilização do segundo método de Zeigler-Nichols, conforme pode ser visto na seção 4.2.1.2. A Figura 4.8 ilustra o resultado da simulação do controlador fazendo-se $T_i = \infty$ e $T_d = 0$. Como resultado desta simulação, temos que $P_{cr} = 2,5$ s e $K_{cr} = 10000$. Desta maneira, conforme pode ser visto na Tabela 4.3, chegou-se aos seguintes valores:

$$\begin{aligned}
K_p &= 0,6 K_{cr} = 6000, \\
T_i &= 0,5 P_{cr} = 1,25, \\
T_d &= 0,125 P_{cr} = 0,3125.
\end{aligned} \tag{4.13}$$

4.3 Análise dinâmica do sistema

Uma vez obtidos o modelamento dinâmico do sistema, conforme pode ser visto na seção 4.1.2, é possível analisar o desempenho do sistema a partir de vários métodos disponíveis. Na prática, o sinal de entrada de sistemas de controle não é conhecido previamente: ele é de caráter aleatório e seus valores instantâneos não podem ser expressos de maneira analítica. Na análise e no projeto de sistemas de controle, deve-se ter uma base de comparação do de-

sempenho de vários sistema de controle. Essa base pode ser estabelecida detalhando-se sinais de entrada de teste específicos e, em seguida, comparando-se as respostas dos vários sistemas com esses sinais. Os sinais de testes geralmente utilizados são as funções degrau, rampa, parábola, impulso, senoidais, etc. Com estes sinais de teste, tanto a análise experimental como a análise matemática dos sistemas de controle podem ser obtidas facilmente.

A Figura 4.9 apresenta a implementação do sistema no Simulink®, de onde é feita toda a análise dinâmica do sistema e obtém-se os resultados experimentais da simulação. A Figura 4.9 ilustra a expansão dos subsistemas criados. No nível mais baixo temos o diagrama de blocos no motor CC. Um nível acima o controlador PID, e em primeiro plano o sistema de tração de uma das rodas do robô móvel, com entrada de referência em velocidade linear da roda e saída deste sinal passando pelo sistema.

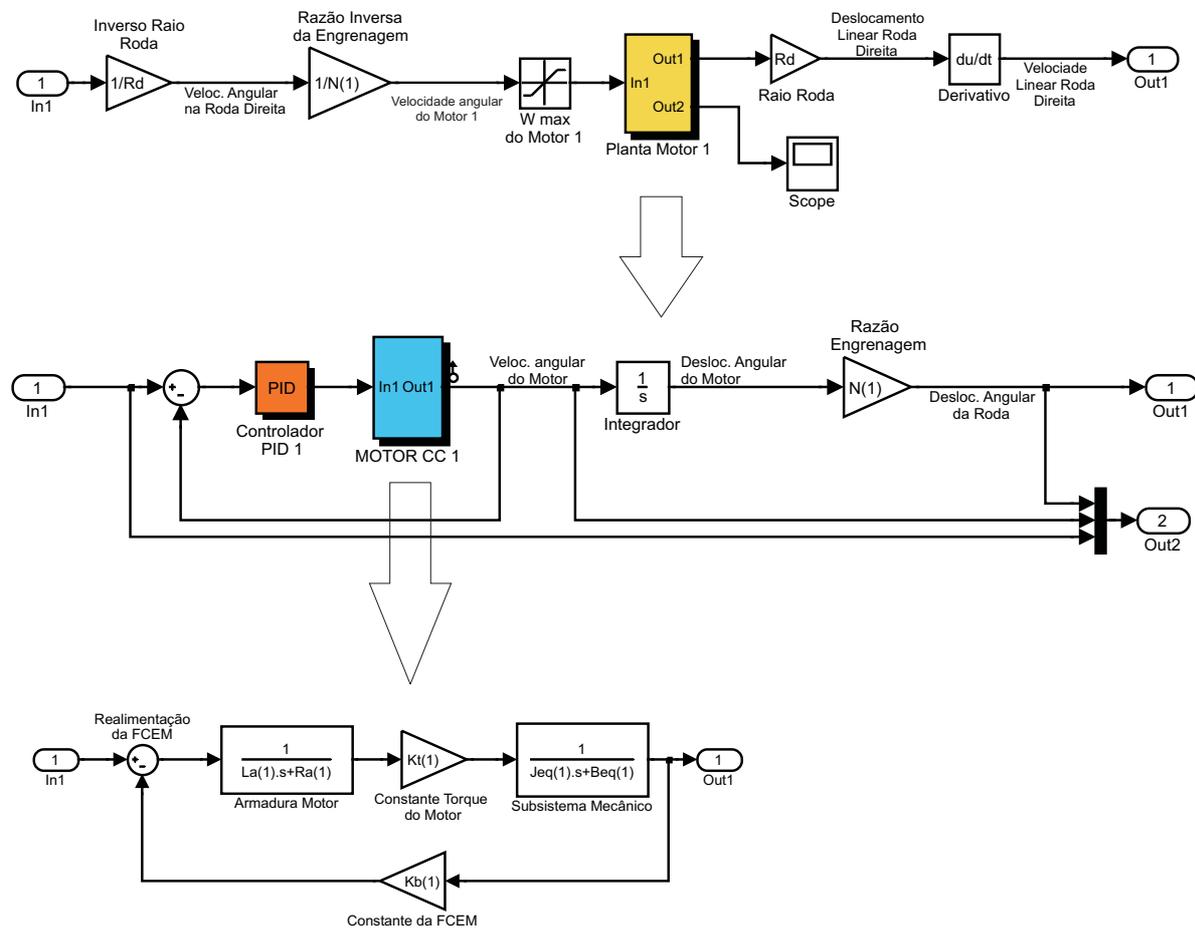


Figura 4.9: Análise dinâmica do sistema no Simulink®.

A seguir são apresentadas as técnicas de análise dinâmica do sistema implementado.

4.3.1 Análise de resposta transitória e de regime estacionário

A resposta temporal de um sistema de controle é constituída de duas partes: a resposta transitória e a resposta estacionária. Para respostas transitórias, entende-se aquela que vai do estado inicial ao estado final. Por resposta estacionária, entende-se o comportamento do sinal de saída à medida em que t tende ao infinito.

Implementando-se os valores das equações 4.13 no controlador PID do sistema, obtém-se como resposta ao degrau unitário, o gráfico ilustrado na Figura 4.10.

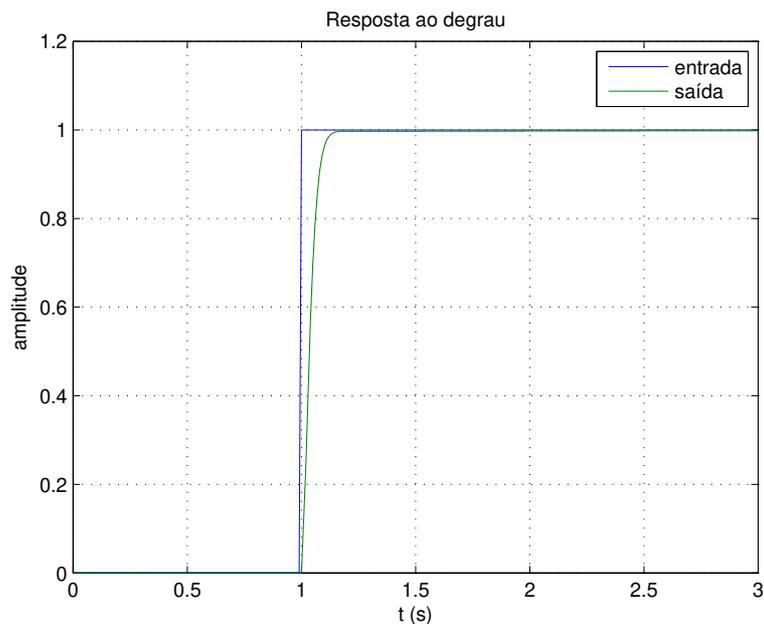


Figura 4.10: Resposta ao degrau.

Não é do escopo deste trabalho um estudo mais aprofundado a respeito do controlador PID utilizado, mas apenas uma apresentação de suas características e os resultados experimentais obtidos, através de simulações do modelo da planta. A Figura 4.11 ilustra o gráfico da velocidade angular ω_m do motor e o deslocamento angular θ_L da carga em função de uma entrada em degrau unitário.

O controlador PID atua no controle da velocidade angular ω_m do motor. Nota-se, através dos resultados obtidos pelas Figuras 4.10 e 4.11, que tanto na resposta transitória a um sinal degrau na entrada, como em regime permanente, o controlador atua de modo efetivo e eficiente.

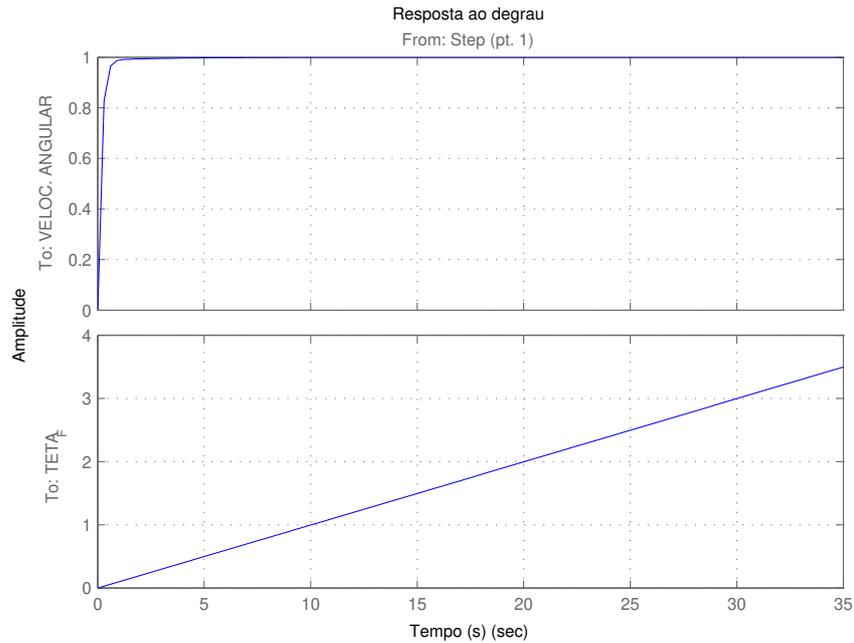


Figura 4.11: Gráficos de ω_m e θ_L em resposta ao degrau.

4.3.2 Análise do lugar das raízes

A característica básica da resposta transitória de um sistema de malha fechada depende essencialmente da localização dos pólos de malha fechada. Se o ganho de malha do sistema for variável, então a localização dos pólos e malha fechada dependerá do valor do ganho de malha escolhido. É importante, então, que se conheça como os pólos de malha fechada se movem no plano s , à medida que o ganho de malha varia.

Um método simples para a determinação das raízes da equação característica foi desenvolvido por Evans (1948) e tem sido amplamente utilizado na engenharia de controle. Esse método, chamado *método do lugar das raízes*, permite que as raízes da equação característica sejam representadas graficamente para todos os valores de um parâmetro do sistema. As raízes correspondentes a um valor específico desse parâmetro podem, então, ser localizadas no gráfico resultante. Normalmente o parâmetro é o ganho, mas é possível utilizar qualquer outra variável da função de transferência de malha aberta. O gráfico do lugar das raízes mostra claramente as contribuições de cada pólo ou zero de malha aberta nas localizações dos pólos de malha fechada.

No projeto de um sistema de controle linear o método do lugar das raízes prova sua eficiência, pois indica o modo pelo qual os pólos e os zeros de malha aberta devem ser

modificados, para que a resposta satisfaça as especificações de desempenho do sistema. Esse método é, em particular, eficiente para a obtenção rápida de resultados aproximados (NISE, 2002).

A Figura 4.12 ilustra os valores dos pólos e zeros obtidos pelo simulador. Tem-se dois pólos em $-5,8 \pm 40,7j$ e um pólo e um zero em $-0,333$.

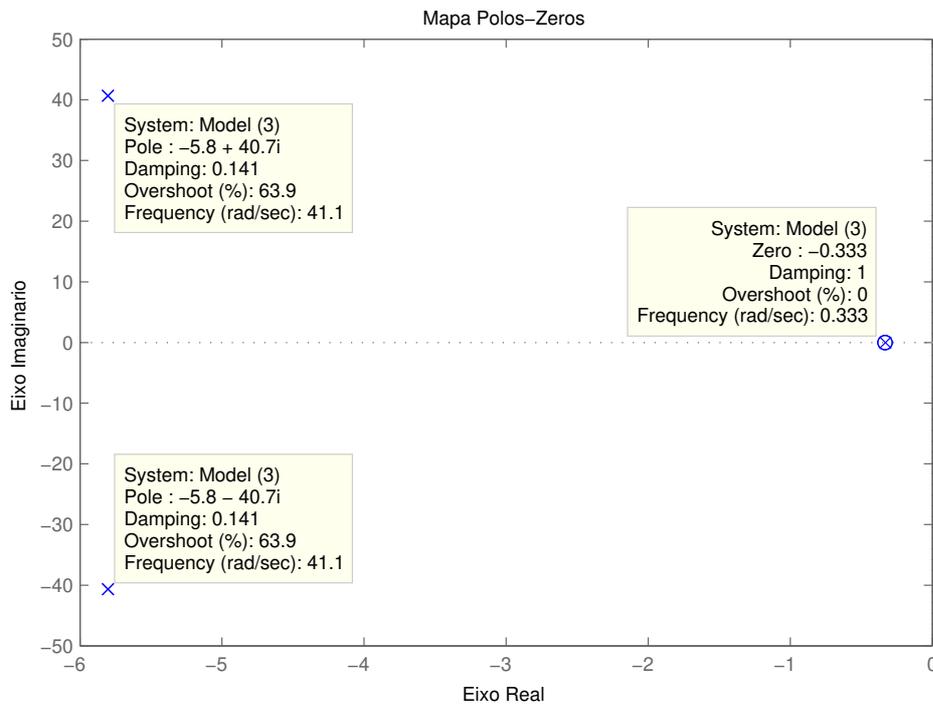


Figura 4.12: Pólos e zeros do sistema.

A Figura 4.13 e ilustra o gráfico do lugar das raízes do sistema. Percebe-se a localização dos pólos e zeros do sistema, mostrando sua estabilidade.

4.3.3 Análise de resposta em frequência

O termo *resposta em frequência* significa a resposta em regime permanente de um sistema a uma entrada senoidal. Nos métodos de resposta em frequência, varia-se a frequência do sinal de entrada dentro de um certo intervalo e estuda-se a resposta resultante (OGATA, 2003). Os métodos de resposta em frequência foram desenvolvidos entre 1930 e 1940 por Nyquist, Bode, Nichols e outros. Os métodos de resposta em frequência são dos mais poderosos na teoria de controle convencional, sendo indispensáveis também na teoria de controle robusto.

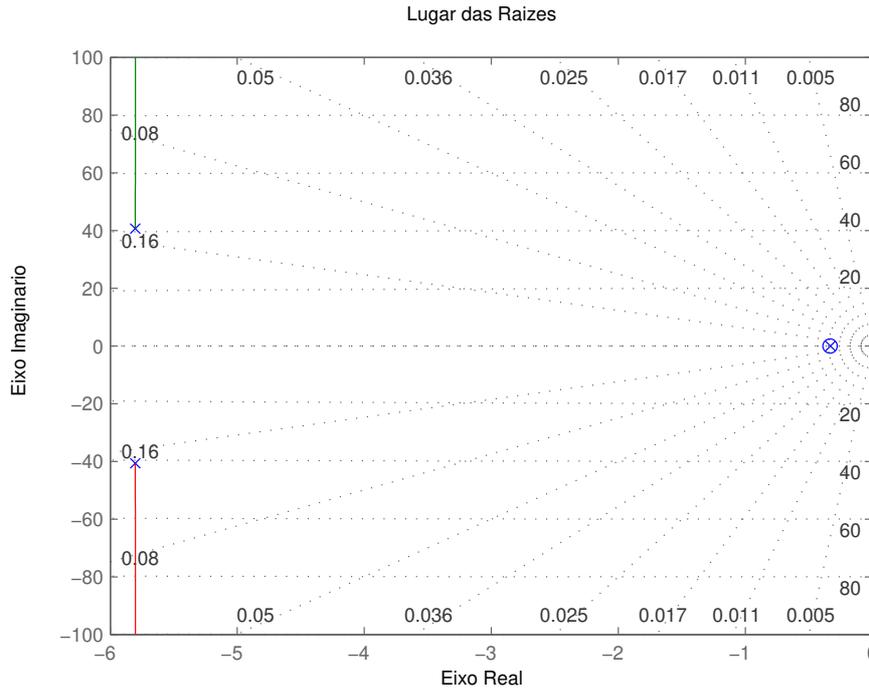


Figura 4.13: Gráfico do lugar das raízes do sistema.

O critério de estabilidade de Nyquist nos possibilita pesquisar tanto a estabilidade absoluta como a relativa dos sistemas lineares de malha fechada, com base no conhecimento de suas características de resposta em frequência de malha aberta. Uma vantagem do método de resposta em frequência é que os testes da resposta em frequência são, em geral, simples e podem ser realizados com exatidão, com a utilização de geradores de sinais senoidais ou através de simuladores, se for conhecido a função de transferência do sistema. Muitas vezes, as funções de transferência de componentes complexos podem ser determinadas experimentalmente por meio de testes de resposta em frequência. Além disto, o enfoque da resposta em frequência apresenta a vantagem de permitir que se projete um sistema de maneira que os efeitos de ruídos indesejáveis sejam desprezíveis e que a essa análise e esse projeto possam ser estendidos a certos sistemas de controle não-lineares (NISE, 2002).

4.3.3.1 Diagramas de Bode

Um diagrama de Bode é constituído de dois gráficos: um é o gráfico do módulo em dB de uma função de transferência senoidal; o outro é o gráfico do ângulo de fase. Ambos são traçados em relação à frequência em escala logarítmica.

A Figura 4.14 ilustra o resultado da análise do comportamento do sistema em resposta a frequência. O sinal de entrada é a velocidade linear da roda que é dado por um degrau unitário. Como saída tem-se a velocidade angular ω_m do motor. No diagrama de Bode traçado, tem-se a limitação da frequência na velocidade angular máxima do motor CC.

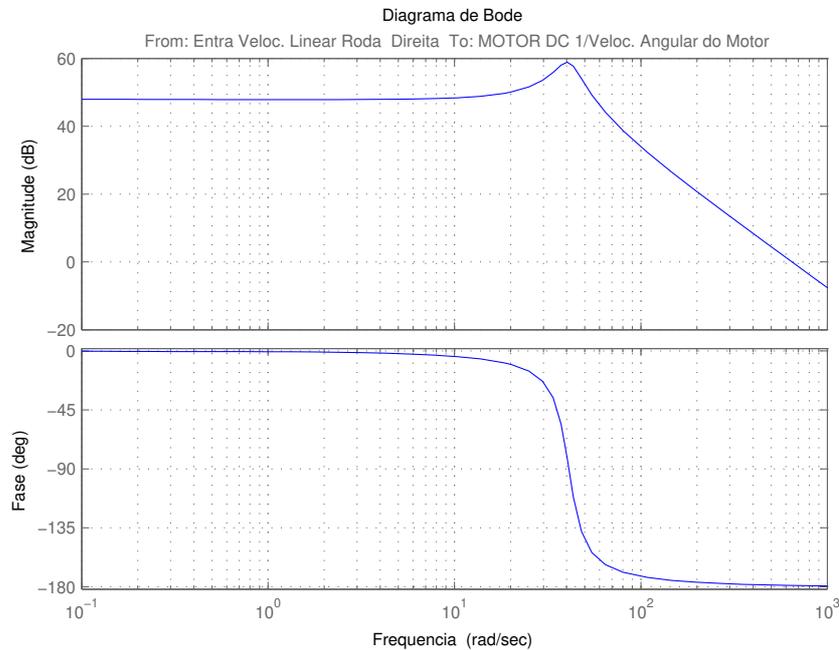


Figura 4.14: Análise de resposta em frequência com o módulo em dB e o ângulo de fase.

4.3.3.2 Diagramas de Nyquist

O diagrama polar de uma função de transferência senoidal $G(j\omega)$ é um gráfico do módulo de $G(j\omega)$ versus o ângulo de fase de $G(j\omega)$ em coordenadas polares, com ω variando de zero a infinito. Assim, o diagrama polar é o lugar dos vetores $|G(j\omega)|\angle G(j\omega)$, com ω variando de zero ao infinito. O diagrama polar é frequentemente chamado de diagrama de Nyquist.

Uma vantagem em utilizar um diagrama polar é que este representa as características da resposta em frequência de um sistema em toda a faixa de frequências em um único gráfico. Uma desvantagem é que o diagrama não indica claramente as contribuições de cada fator individual sobre a função de transferência de malha aberta (OGATA, 2003).

Os diagramas de Nyquist, assim como os diagramas de Bode, são comumente utilizados para a representação da resposta em frequência de sistemas de controle com realimentação, lineares e invariantes no tempo. Os diagramas de Nyquist são diagramas polares, enquanto

dos diagramas de Bode são retangulares. Um dos diagramas pode ser mais conveniente para uma operação em particular, mas uma dada operação pode sempre ser conduzida por qualquer um dos dois diagramas (D'AZZO; HOUPIS; SHELDON, 2003).

A Figura 4.15 ilustra o Diagrama de Nyquist do sistema.

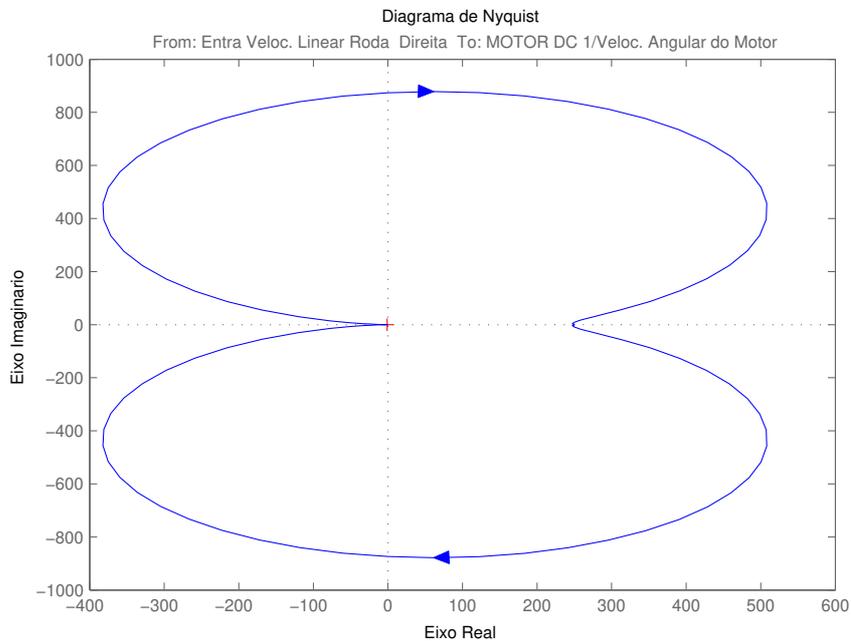


Figura 4.15: Diagrama de Nyquist.

4.3.3.3 Carta de Nichols

Uma outra maneira de representar graficamente as características da resposta em frequência é com a utilização do diagrama do módulo em dB *versus* ângulo de fase. A curva é graduada em termos da frequência do sinal de entrada. Esses diagramas são chamados normalmente de carta de Nichols.

No diagrama de Bode, as características de resposta em frequência de $G(j\omega)$ são representadas em papel semilog por duas curvas separadas, a curva do módulo em dB e a curva do ângulo de fase, enquanto que, na carta de Nichols, as duas curvas do diagrama de Bode são combinadas em uma única. As vantagens da carta de Nichols são que a estabilidade relativa do sistema de malha fechada pode ser determinada rapidamente e que a compensação pode ser realizada com facilidade.

A Figura 4.16 ilustra a carta de Nichols do sistema.

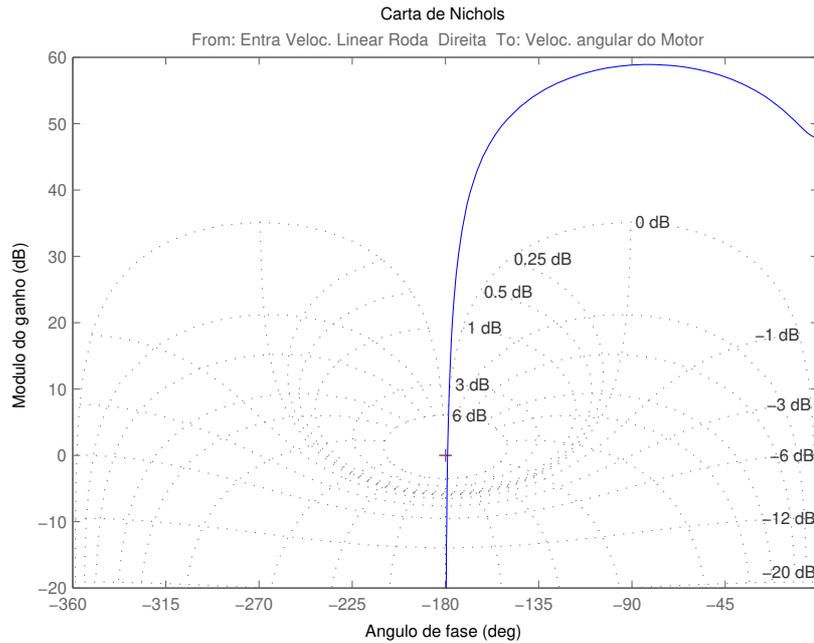


Figura 4.16: Carta de Nichols.

4.4 Considerações finais

Neste capítulo apresentou-se a modelagem matemática cinemática adotada para o robô móvel, assim como sua modelagem dinâmica. Optou-se pela escolha do controlador PID que é um controlador eficiente e simples para o atuar no modelo dinâmico do sistema. Foram apresentadas as regras de sintonia do controlador assim como toda a análise dinâmica do sistema, com ilustrações de gráficos analíticos. Através da análise dos resultados encontrados, verifica-se a estabilidade e eficiência do controlador PID escolhido para atuar na planta, demonstrando que este tipo de controlador é suficientemente eficiente para o controle dinâmico do robô móvel com as características apresentadas. Não obstante, fica claro que outro controlador pode ser utilizado, uma vez que o sistema tem arquitetura aberta e trabalha com implementação em prototipagem rápida, bastando para isto, substituir alguns blocos no simulador para se efetuar a troca da tecnologia do controlador.

Nos capítulos subseqüentes serão abordados os temas relacionados ao simulador virtual e a prototipagem rápida. Nestes contextos serão inseridos todos os estudos verificados neste capítulo concernentes ao modelamento cinemático e dinâmico do sistema robótico móvel.

Capítulo 5

Implementação do simulador virtual

Este capítulo tem como objetivo apresentar as principais características de um simulador de sistemas robóticos móveis. Ele foi implementado a partir do modelo cinemático e dinâmico dos sistemas de acionamento mecânico dos eixos robóticos, para a simulação de diferentes técnicas de controle no campo da robótica móvel, permitindo-se aprofundar os conceitos de sistemas de navegação, planejamento de trajetórias e sistemas de controle embarcado. Esse simulador, implementado em arquitetura modular e aberta, conforme apresentado no capítulo 3, permite a aplicação direta de vários conceitos dentro da área da robótica móvel, sendo utilizado para sua validação, e como objetivo principal deste estudo, o modelo de um protótipo de robô móvel com restrições não holonômicas e tração diferencial com dois graus de liberdade (movimento de deslocamento linear e de rotação).

Para o desenvolvimento do simulador, foram considerados os aspectos construtivos do protótipo do robô móvel, incluindo a modelagem cinemática e dinâmica do sistema de acionamento e controle. O simulador apresenta o módulo gerador de trajetórias que é o primeiro bloco do sistema e foi implementado com a funcionalidade de gerar uma trajetória para o robô móvel a partir de uma matriz de pontos fornecidas inicialmente. Outro bloco apresentado é o controlador implementado na forma PID, conforme descrito na seção 4.2.

Para a implementação final do controlador utiliza-se o conceito de prototipagem rápida, onde após validação e testes do sistema completo de controle e supervisão através do simulador, a interface de supervisão e hardware de controle poderão ser implementados de maneira rápida, segura e aberta, eliminando custos envolvidos na confecção de protótipos.

5.1 Visão geral do sistema

A Figura 5.1 apresenta um visão geral do sistema simulador proposto. A utilização do sistema tem início pela captação de pontos principais para geração da trajetória do robô móvel. A idéia é utilizar um sistema de câmara de vídeo ou fotográfico, que capte a imagem do ambiente em que o robô móvel irá navegar. Este sistema inicial deve ser capaz de, através da imagem captada, identificar os obstáculos do ambiente e gerar uma matriz com alguns pontos estratégicos que servirão de entrada para o sistema de geração de trajetória.

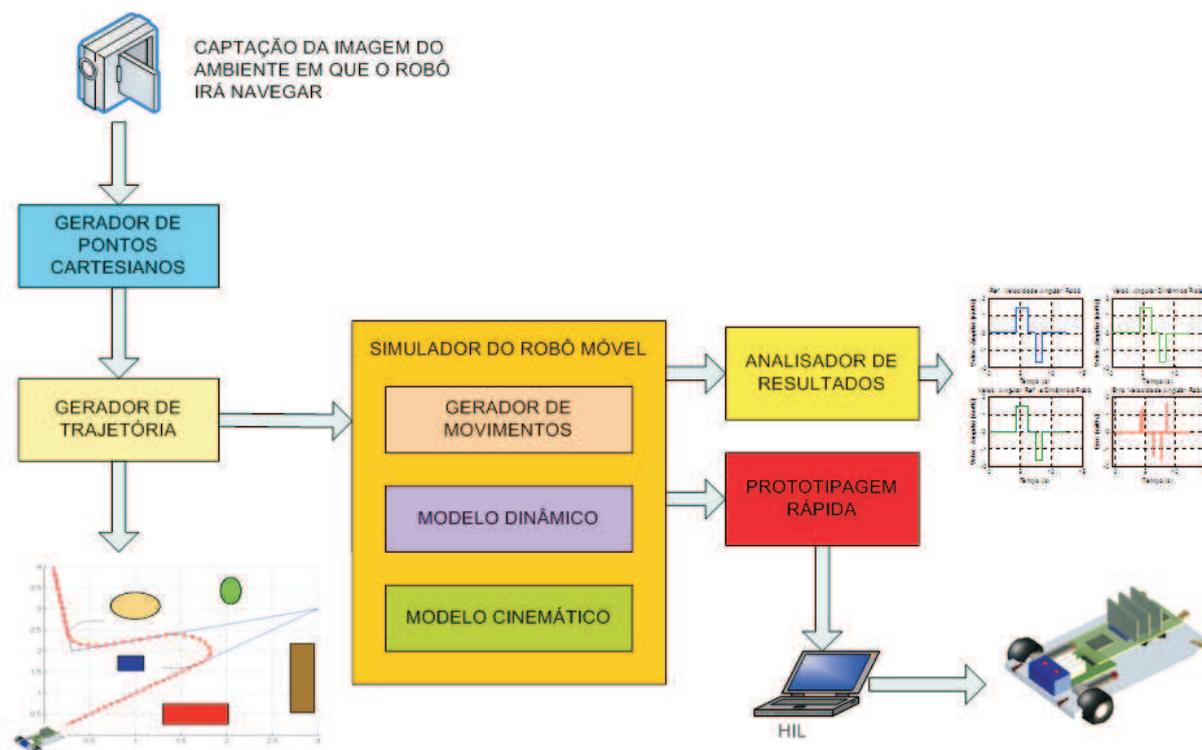


Figura 5.1: Uma visão geral do sistema.

A Figura 5.2 ilustra um exemplo de um ambiente com alguns obstáculos onde o robô deve navegar. Neste ambiente, o robô está localizado inicialmente no ponto P1 e tem como objetivo alcançar o ponto P4. O sistema gerador de pontos cartesianos iniciais, deve então fornecer ao módulo de geração de trajetória, os pontos cartesianos P1, P2, P3 e P4, que são os pontos principais da rota a ser traçada.

Este sistema é particularmente interessante e pode ser empregado, por exemplo, em futebol robóticos, onde as estratégias de navegação são feitas a partir de imagens do ambiente (campo de futebol) e os obstáculos são os outros robôs jogadores. Como é descrito a seguir,

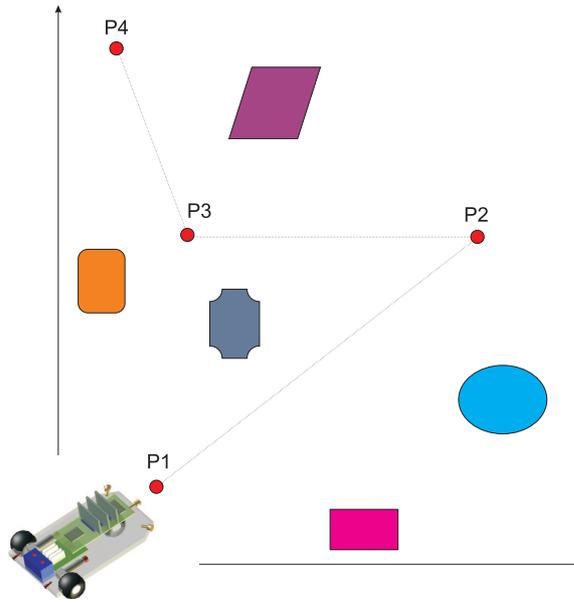


Figura 5.2: Exemplo de um ambiente com obstáculos para navegação do robô móvel.

com este sistema, pode-se definir a melhor trajetória a ser traçada, respeitando-se sempre as restrições holonômicas ou não holonômicas dos sistemas robóticos em questão, e fazer toda a simulação do sistema prevendo falhas e analisando resultados antes da implementação final da sistema de controle no robô móvel (MELO; ROSÁRIO; LIMA, 2004a).

Uma vez gerados os pontos principais da trajetória a ser seguida pelo robô móvel, e conhecendo-se as restrições a que se está sujeito o sistema robótico móvel, começa-se a primeira etapa da utilização do sistema virtual de simulação, com a utilização do bloco gerador de trajetória, que é descrito a seguir.

Na Figura 5.1 pode-se observar também o bloco simulador do robô móvel, que recebe uma matriz de dados da trajetória, gerados pelo módulo gerador de trajetória, e faz a simulação cinemática e dinâmica do sistema. Os detalhes serão apresentados na seção 5.3. Por fim, temos os blocos de análise de resultados e de prototipagem rápida, que finalizam os sistema. As análise de resultados são feitas através do bloco analisador de resultados, que gera uma série de gráficos a partir de todos os dados coletados na simulação. O bloco da prototipagem rápida, através da técnica de HIL (*hardware-in-the-Loop*) prepara e carrega o sistema embarcado do robô móvel com os dados de comando do simulador, deixando o controlador embarcado pronto para a navegação no ambiente proposto. Estes módulos serão detalhados mais adiante neste capítulo.

5.2 Bloco gerador de trajetória

O planejamento básico de trajetória para os robôs móveis refere-se à determinação de um caminho no espaço-C (*espaço de configuração*) entre uma configuração inicial do robô e uma configuração final, de tal forma que o robô não colida com nenhum obstáculo no ambiente, e que a movimentação planejada seja consistente com as restrições cinemáticas do veículo. As digressões a este respeito podem ser vistas na seção B.2. A seguir é apresentada a descrição de implementação e operacionalidade do bloco gerador de trajetória.

O bloco gerador de trajetória, como pode ser visto na seção 5.1, recebe alguns pontos importantes para que seja traçada a trajetória a ser realizada pelo robô móvel. Estes pontos formam uma matriz cartesiana que podem conter mais ou menos pontos, dependendo da complexidade do ambiente. A título de testes e para validação do sistema, o número de pontos a ser alimentado o sistema foi fixado em quatro. Não obstante, o número de pontos podem ser aumentados dependendo da complexidade do ambiente em que o robô móvel irá navegar. Outro dado importante a ser utilizado pelo sistema tem relação as restrições holonômicas ou não do robô móvel modelado. Deve-se informar ao sistema o raio de curvatura a ser realizado na trajetória. Uma vez que, a título de testes, foi fixada em quatro o número de pontos cartesianos de entrada, deve-se fornecer também o raio das duas curvas a serem executadas pelo robô. A informação de raios distintos torna o sistema flexível, podendo a trajetória ser traçada com curvaturas diferentes dependendo do ângulo de deslocamento de direção e das restrições do robô.

A Figura 5.3 ilustra a implementação do bloco gerador de trajetória feita como uma *script* no MatLab[®].

O bloco gerador de trajetória fornece a trajetória final do robô móvel sob várias formas. Pode informar o posicionamento do robô (x, y, θ) em cada intervalo de tempo Δt da trajetória. Esta informação fica armazenada em forma de uma matriz de três linhas versus o número de colunas em que se estabeleceu inicialmente. Outra forma de se passar a informação da trajetória do robô ao sistema simulador virtual, é através de uma matriz com os pontos inicial e final de cada segmento de reta com suas respectivas inclinações angulares θ . Esta segunda maneira foi a escolhida como dados de entrada para no simulador virtual do robô móvel, como é mostrado na seção 5.3.

```
Editor - C:\Leo\work\gerador_trajetorias10.m
File Edit Text Cell Tools Debug Desktop Window Help
Stack: Base
1 %Função geradora de trajetória para o robô móvel não holonômico
2 % Entradas: Xtraj, Ytraj, R, ds
3 % (Xtraj(i), Ytraj(i)) indicam pontos da trajetória
4 % R(i) é o raio de transições entre trajetórias
5 % ds(i) tamanho do deslocamento na trajetória
6
7 - inicializacao_3
8
9 - tj2
10 - figure;
11 - hold on;
12 - for i=1:4
13 -     plot(Xtraj(i),Ytraj(i),'go','markerfacecolor','g');
14 - end
15
16 - Xres = [];
17 - Yres = [];
18 - R_cur = [];
19 - Temp = [];
20 - P1 = [ 1, -1, -1];
21 - P2 = [ 1,  1,  1];
22 - P3 = [-1,  1,  1];
23 - P4 = [-1, -1, -1];
24 %clc;
25
26 - Xi = Xtraj(1);
27 - X_inic = Xi;
28 - Yi = Ytraj(1);
29 - Y_inic = Yi;
30 %tet(1) = atan2(Ytraj(2) - Ytraj(1), Xtraj(2) - Xtraj(1));
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
263
```

5.2.1 Considerações matemáticas da implementação do gerador de trajetória

A idéia básica é utilizar os pontos cartesianos e, em conjunto com os raios de curvatura, traçar uma trajetória que combine retas e semicírculos. As retas serão trechos de retas entre os pontos fornecidos e os semicírculos irão depender dos raios de curvaturas informados.

O primeiro passo é encontrar o ângulo de inclinação θ inicial do robô. A equação 5.1 é utilizada para encontrar o θ_i .

$$\theta_i = \arctan \frac{(y_f - y_i)}{(x_f - x_i)} \quad (5.1)$$

As inclinações das outras retas da trajetória são encontradas de maneira análoga. O ponto tangente é determinado pelas equações 5.2 e 5.3 onde r é o raio de curvatura.

$$x = \frac{r(\sec \theta_f + \sec \theta_i) + y_i - y_f - x_i \tan \theta_i + x_f \tan \theta_f}{\tan \theta_f - \tan \theta_i} \quad (5.2)$$

$$y = r \sec \theta_i + x \tan \theta_i + y_i - x_i \tan \theta_i \quad (5.3)$$

O cálculo do centro da circunferência e os coeficientes é determinado pelas equações 5.4.

$$\begin{aligned} a &= 1 + \tan^2 \theta_i \\ b &= 2y_i \tan \theta_i - 2x - 2y \tan \theta_i - 2x_i \tan^2 \theta_i \\ c &= y_i^2 - 2y_i y - 2y_i x_i \tan \theta_i + 2y x_i \tan \theta_i + x_i^2 \tan^2 \theta_i - r^2 + x^2 + y^2 \end{aligned} \quad (5.4)$$

As raízes da equação do segundo grau representam o ponto (x_{ti}, y_{ti}) onde a reta tangencia a circunferência, conforme equações 5.5.

$$\begin{aligned} \Delta_1 &= b^2 - 4ac \\ x_1 &= -\frac{b}{2a} \\ \lambda &= (r^2 - (x_1 - x))(x_1 - x) \\ x_{ti} &= x_1 \\ y_{ti} &= y_i - (x_i - x_1) \tan \theta_i \end{aligned} \quad (5.5)$$

A escolha da circunferência certa para a mudança de trajetória é feita com a identificação inicial da inclinação a partir do ponto de intersecção das trajetórias retas e circulares. A

inclinação da trajetória circular θ_c é dada por

$$\theta_c = \arctan \frac{(y - y_e)}{(x - x_e)}, \quad (5.6)$$

onde x_e e y_e são os pontos de intersecção das duas retas.

O comprimento da trajetória reta entre o ponto inicial e o começo do semicírculo do raio de curvatura é dado por

$$t_r = \sqrt{(x_{ti} - x_i)^2 + (y_{ti} - y_i)^2}, \quad (5.7)$$

e os ângulos inicial α_i e final α_f da parte da circunferência utilizada na trajetória é dado por

$$\alpha_i = \arctan \frac{(y_{ti} - y)}{(x_{ti} - x)}, \quad (5.8)$$

$$\alpha_f = \arctan \frac{(y_{tf} - y)}{(x_{tf} - x)}, \quad (5.9)$$

e o comprimento do arco p é dado por

$$p = r|\alpha_i - \alpha_f|. \quad (5.10)$$

O programa gerador de trajetória é apresentado em sua forma integral no anexo A.

5.2.2 Gráficos do gerador de trajetória

O gráfico da Figura 5.4 ilustra os pontos que servem de início para o gerador de trajetória. Neste exemplo, escolheu-se o vetor $\mathbf{x} = [.1 \ 3 \ .3 \ .1]$ e o vetor $\mathbf{y} = [.1 \ 3 \ 2 \ 4]$, com os raios de curvatura para o primeiro e o segundo semicírculo dados pelo vetor $\mathbf{r} = [.4 \ .3]$. Todas as medidas são em metros.

Na Figura 5.5 vê-se o traçado das retas, dos semicírculos e de um seguimento de reta intermediário indicando o começo e o fim do traçado de cada movimento circular a ser executado pelo robô móvel.

O traçado final da trajetória do robô móvel pode ser observado na Figura 5.6.

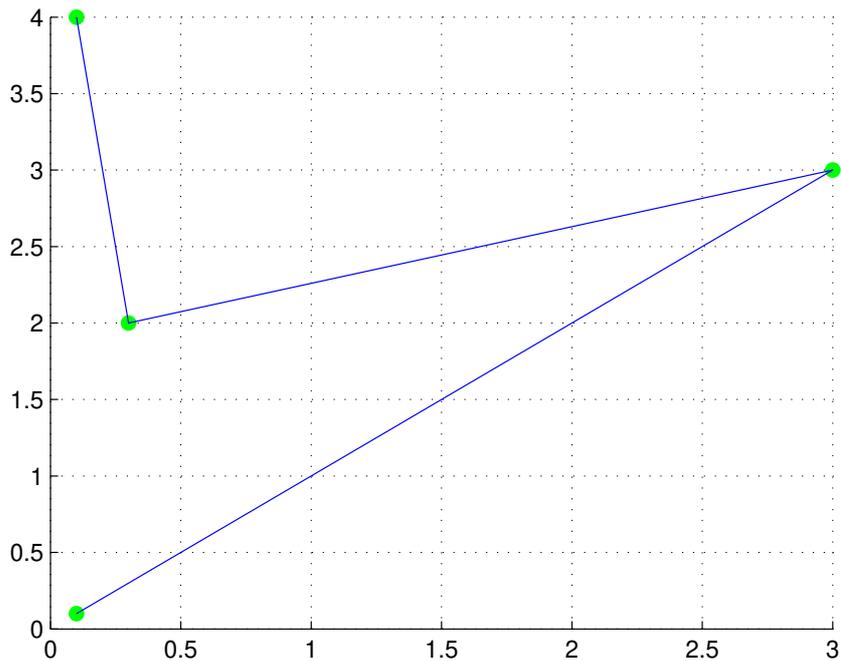


Figura 5.4: Pontos iniciais fornecidos ao gerador de trajetória.

5.3 O sistema simulador virtual

O sistema simulador virtual do robô móvel é formado por três blocos principais. O primeiro é chamado de bloco de geração de movimentos. O segundo é o bloco do controlador e modelo dinâmico do robô móvel. O terceiro é o bloco do modelo cinemático. Esta estruturação modular é ilustrada na Figura 5.7 onde pode-se observar as variáveis de entrada e saída do sistema (MELO; ROSÁRIO; LIMA, 2004b).

Os objetivos principais do sistema simulador virtual são, primeiro, executar a simulação dinâmica e cinemática do sistema robótico móvel. Segundo, gerar todos os dados para análise gráfica dos resultados obtidos e erros e terceiro, operar técnicas de prototipagem rápida em conjunto com a arquitetura aberta e reconfigurável implementada no robô móvel e explanada no capítulo 3.

As variáveis de entrada do sistema são:

- Δt . Uma das entradas do sistema simulador virtual é o tempo de simulação com o valor dos intervalos Δt .

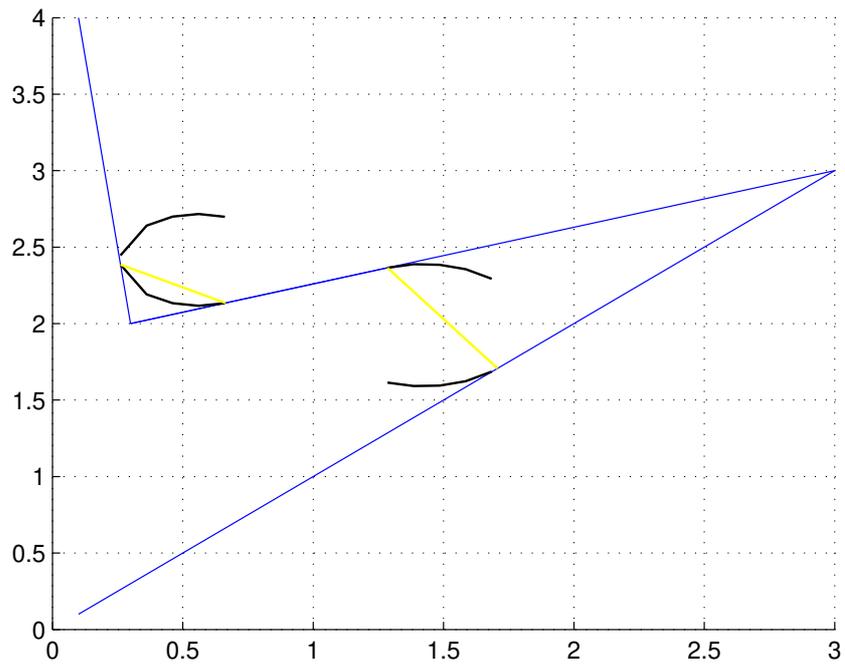


Figura 5.5: Ensaio do traçado da trajetória.

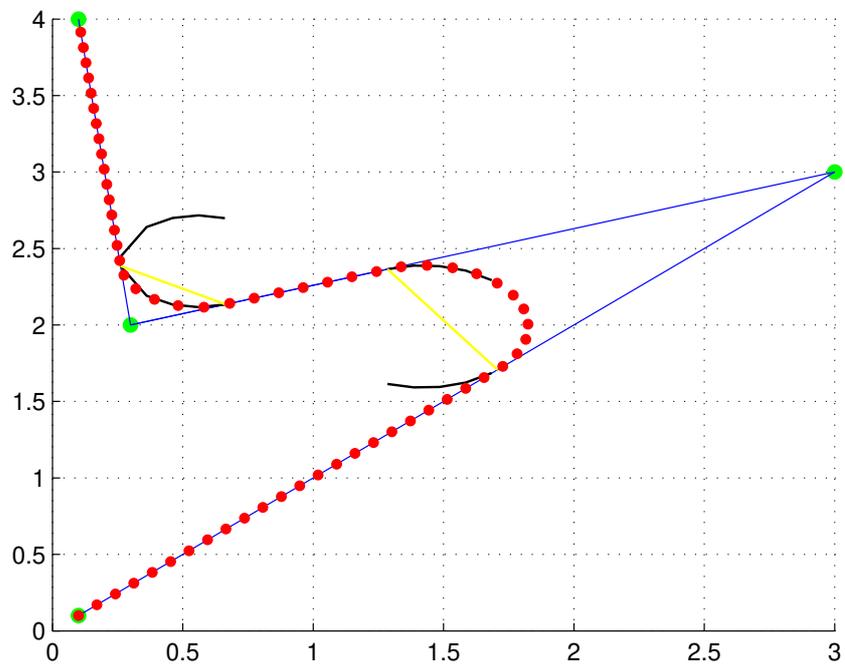


Figura 5.6: Traçado final da trajetória do robô móvel.

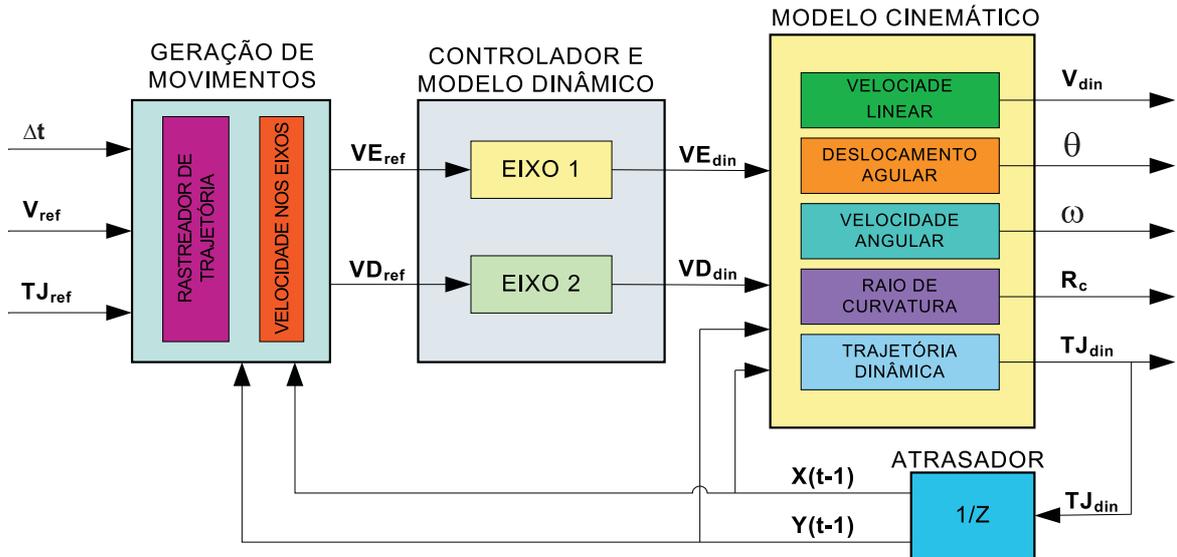


Figura 5.7: Diagrama de Blocos representando os módulos de geração de movimentos, controlador, modelo cinemático e modelo dinâmico.

- \mathbf{TJ}_{ref} , que é a matriz da trajetória de referência.
- V_{ref} , que é a velocidade linear de referência do robô móvel.

Conforme pode ser visto na Figura 5.1, da seção 5.1, o sistema simulador virtual opera em conjunto com o sistema gerador de trajetória, que recebe deste uma matriz de pontos \mathbf{TJ}_{ref} contendo coordenadas de posicionamento (x, y, θ) do robô móvel.

As variáveis de saída do sistema são:

- \mathbf{TJ}_{din} , que é a matriz da trajetória dinâmica do robô móvel, fornecida na forma de uma matriz de coordenadas cartesianas.
- \mathbf{V}_{din} , que é um vetor velocidade dinâmica do robô móvel.
- \mathbf{R}_c , que é um vetor raio de curvatura dinâmico do robô móvel.
- θ , que é vetor ângulo de posicionamento do robô móvel.
- ω , que é um vetor velocidade angular do robô móvel.

Outras saídas também são disponibilizadas tais como o ângulo de deslocamento instantâneo θ_r das rodas do robô móvel, a velocidade angular ω_r das rodas do robô móvel e do eixo do motor ω_e . Estas saídas, e outras mais, serão abordadas na seção que descreve as análises dos resultados da simulação, seção 5.4.

O sistema simulador ilustrado na Figura 5.7 mostra também a presença de um bloco atrasador $1/z$ que tem a função de fornecer a posição da coordenada cartesiana do robô no tempo anterior ao da amostragem atual, resultando em $x(t - 1)$ e $y(t - 1)$. Estes sinais são utilizados nos blocos de modelagem cinemática e de geração de movimentos, como é mostrado a seguir, nas seções 5.3.3 e 5.3.1.

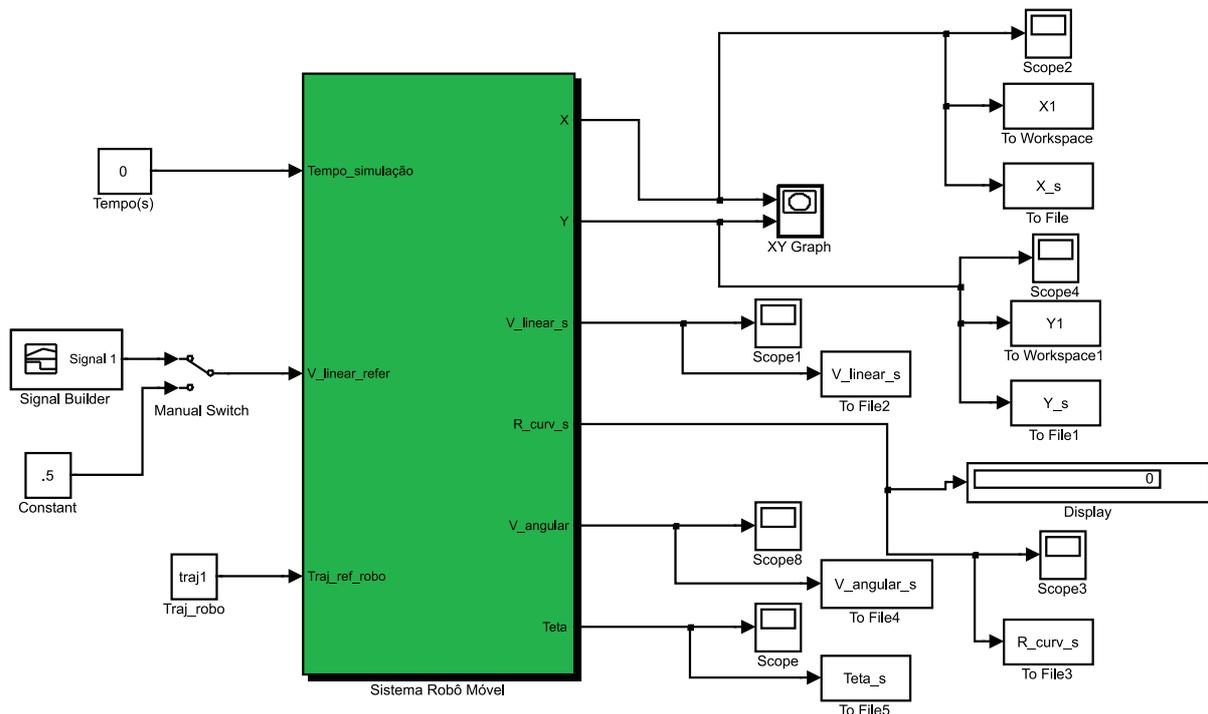


Figura 5.8: Janela inicial do simulador do sistema robótico móvel.

A Figura 5.8 mostra a janela inicial do sistema simulador do robô móvel em sua implementação no Simulink[®]. Como descrito anteriormente, as entradas do sistema são o tempo de simulação, a velocidade linear de referência e a matriz trajetória de referência. A velocidade linear de referência do robô móvel pode ser uma constante ou uma curva em função do tempo. Esta flexibilidade de escolha da velocidade linear de referência do robô móvel é importante para análise do comportamento dinâmico do sistema. A Figura 5.9 exemplifica um sinal de velocidade linear utilizado como entrada no sistema.

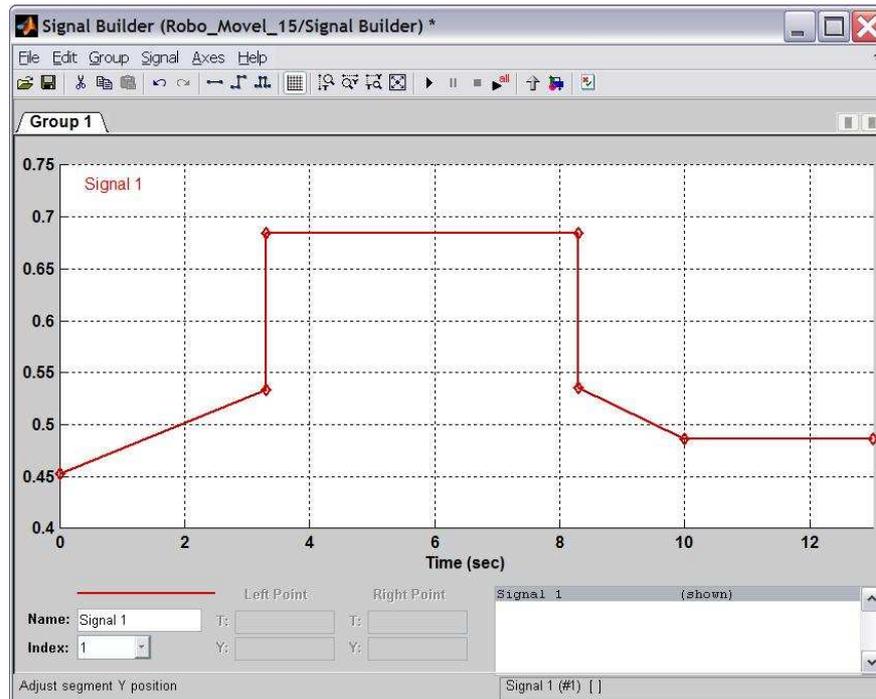


Figura 5.9: Exemplo de um sinal utilizado como entrada para velocidade linear do robô móvel.

As variáveis de saída deste bloco inicial do simulador são os vetores \mathbf{x} e \mathbf{y} , que são utilizados para se traçar a trajetória dinâmica do robô móvel em tempo real de simulação. Outras variáveis de saída são a velocidade linear dinâmica do robô móvel, o raio de curvatura empreendido pelo robô móvel durante todo o trajeto, a velocidade angular e o deslocamento angular θ do robô móvel. Todas estas saídas estão na forma vetorial, indexadas com o tempo, e são armazenadas em arquivos para posterior análise pelo bloco de análise de resultados, seção 5.4.

Dentro do bloco do sistema de robô móvel tem-se o subsistema ilustrado pela Figura 5.10. Pode-se observar os três principais módulos do simulador do sistema robótico móvel que são o bloco gerador de movimentos, o bloco controlador e modelo dinâmico e o bloco de modelo cinemático. A seguir são apresentadas as funcionalidades de cada módulo.

5.3.1 Módulo geração de movimentos

O módulo de geração de movimentos é implementado internamente por um sistema rastreador de trajetória e um sistema gerador das velocidade lineares de referência para as duas rodas do robô móvel. O sistema rastreador de trajetória tem como uma das entradas de

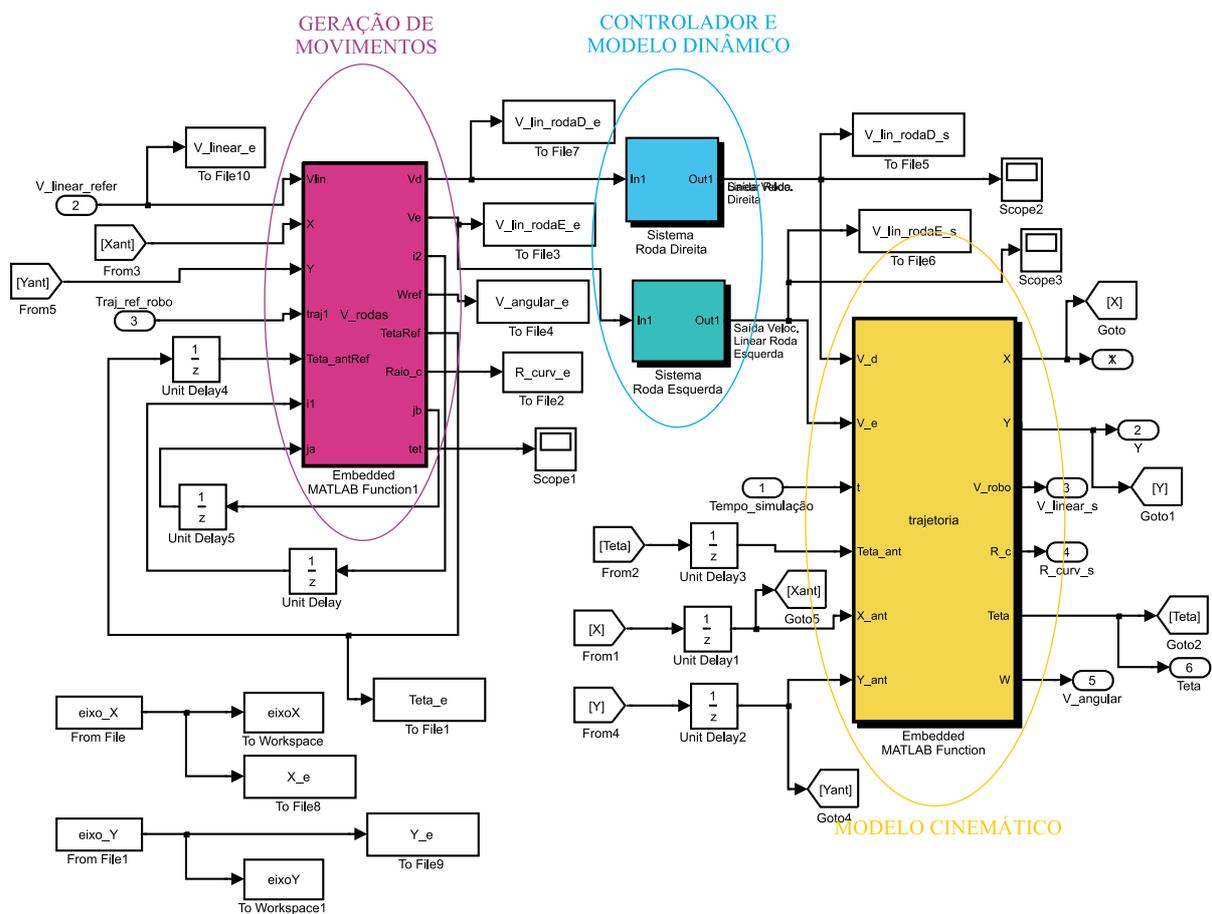


Figura 5.10: Simulador do sistema robótico móvel implementado no Simulink®.

dados a matriz de trajetória de referência, gerada pelo bloco gerador de trajetórias. A matriz trajetória (`traj1`) é composta de:

- coluna 1: pontos iniciais e finais das retas e arcos ref. ao deslocamento X do robô,
- coluna 2: pontos iniciais e finais das retas e arcos ref. ao deslocamento Y do robô,
- coluna 3: raios de curvaturas nos segmentos de reta e nos arcos,
- coluna 4: inclinações angulares (θ) ref. aos segmentos de reta.

```

1 function [Vd, Ve, i2, Wref, TetaRef, Raio_c, jb, tet]= V_rodas(Vlin, L, X, Y, traj1, Teta_antRef, i
2 Delta_t2 = .01;
3 %A matriz trajetória (traj1) é composta de:
4 %na coluna 1 dos pontos inicial e final das retas e arcos ref. ao deslocamento X do ro
5 %na coluna 2 dos pontos inicial e final das retas e arcos ref. ao deslocamento Y do ro
6 %na coluna 3 dos raios de curvaturas nos segmentos de reta e nos arcos
7 %na coluna 4 das inclinações angulares (teta) ref. aos segmentos de reta
8 Raio_c = traj1(i1,3);
9 tet=traj1(i1,4);
10 Xrs = ceil(traj1(i1,1)*6)/6;
11 Xri = floor(traj1(i1,1)*6)/6;
12 Yrs = ceil(traj1(i1,2)*6)/6;
13 Yri = floor(traj1(i1,2)*6)/6;
14
15 if ((X < Xrs) && (X > Xri) && (Y < Yrs) && (Y > Yri))
16     i2 = i1 + 1;
17 else
18     i2 = i1;
19 end
20
21 %Velocidade linear da roda direita e esquerda
22 alfa = L / (2 * Raio_c);
23 Vd = Vlin * (1 + alfa);
24 Ve = Vlin * (1 - alfa);
25
26 %Wref = velocidade angular de referência do robô em rad/s
27 Wref = (Vd - Ve) / L;
28 %Teta = deslocamento angular do robo no tempo em rad
29 Delta_TetaRef = Wref * Delta_t2;
30 TetaRef = Teta_antRef + Delta_TetaRef;

```

Figura 5.11: Bloco gerador de movimentos.

O sistema rastreador de trajetória tem seu princípio de funcionamento no rastreamento da trajetória dinâmica do robô móvel. Ele compara dinamicamente a posição atual do robô com a posição a ser alcançada. A posição a ser alcançada está armazenada na matriz trajetória. Nesta matriz estão relacionados também se a trajetória atual é uma reta ou uma curva. Uma vez rastreado e encontrada a posição cartesiana alvo, o sistema recarrega novos valores

de posicionamento (x, y, θ) da matriz trajetória e recomeça novo rastreamento. Observa-se que a posição cartesiana alvo é um valor de referência e os dados de comparação para o rastreamento são os dados dinâmicos fornecidos pelo sistema, então estes valores não vão coincidir exatamente mas vão estar numa faixa de valores aceitável e ajustável pelo sistema. Se o sistema rastreador de trajetória não conseguir encontrar o alvo, dentro da variação de valores programada, o robô se perde da trajetória e ficará executando uma reta ou um círculo infinito, até o final da simulação. Esta situação é particularmente interessante e acontece na implementação prática do sistema, bastando tentar executar movimentos ou velocidades fora da faixa máxima permitida pelas restrições que o modelo do robô permite. Por exemplo, se for tentado executar uma curva com raio menor do que a permitida pelas restrições não holonômicas do robô o mesmo não vai conseguir executar a trajetória e ficará perdido em loop circular! Outro exemplo seria a execução de uma curva com velocidade linear maior do que a permitida pelo sistema robótico. Isto irá resultar numa tentativa de velocidade linear em uma das rodas maior do que a permitida, causando um erro de trajetória.

A interface de velocidade do bloco gerador de movimentos executa dinamicamente a conversão das variáveis de entrada: Raio de curvatura da trajetória (R_c) e velocidade linear do robô (V_{lin}) nas variáveis de saída a serem utilizadas nos módulos controle e acionamento dos eixos: Velocidade linear da roda direita (V_d) e velocidade linear da roda esquerda (V_e). A Figura 5.11 ilustra a implementação do bloco gerador de movimentos desenvolvida em linguagem *script* do *Embedded MatLab® Editor* para *Simulink®*. No Anexo B encontra-se a listagem completa do software do bloco gerador de movimentos.

5.3.2 Módulo do controlador e modelo dinâmico

O módulo do controlador e modelo dinâmico apresenta os componentes relativos ao modelamento elétrico e mecânico do sistema de acionamento (neste caso motores CC), e ainda a implementação do controlador utilizado. Os parâmetros de entrada deste bloco são velocidade linear de referência da roda direita (V_d) e velocidade linear de referência da roda esquerda (V_e). As saídas são as respectivas velocidades dinâmicas de cada roda. A Figura 5.12 ilustra os componentes do bloco do controlador referente à roda direita. A roda esquerda tem um bloco idêntico. Observa-se que todos os dados referentes à dinâmica do sistema estão disponíveis para análise. Tem-se a velocidade angular da roda, a velocidade angular do motor, o deslocamento angular dinâmico da roda, o deslocamento linear dinâmico da roda e a velocidade linear dinâmica da roda. Todas as análises dinâmicas destas variáveis estão disponibilizadas no módulo analisador de resultados, descrito na seção 5.4.

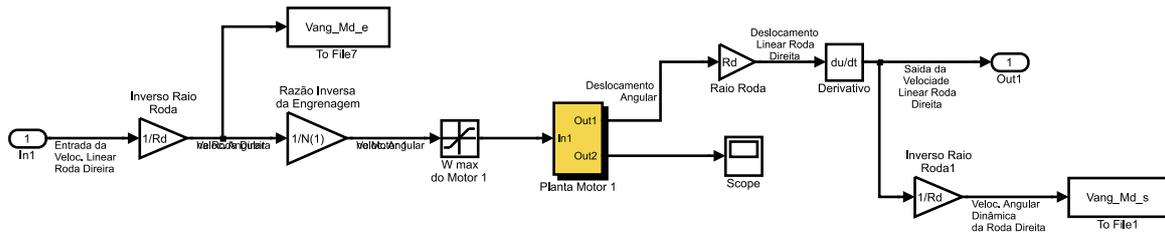


Figura 5.12: Módulo do controlador e modelo dinâmico da roda direita.

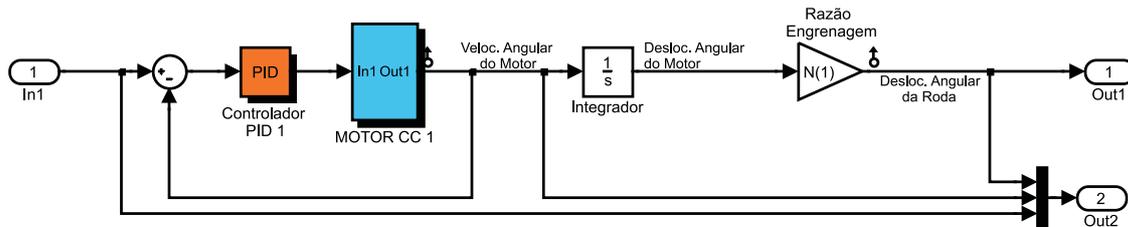


Figura 5.13: Controlador PID dos atuadores do robô móvel.

A Figura 5.13 ilustra a implementação do controlador PID. Os detalhes do modelamento deste controlador com sua análise de desempenho pode ser visto na seção 4.2. Cada eixo do robô móvel tem um controlador PID implementado para atuar independentemente. Os acionadores dos dois eixos são motores de CC cujo bloco representativo pode ser visto na Figura 5.13. O modelamento destes motores com seus respectivos diagramas de blocos e implementação pode ser visto na seção 4.1.2.1.

5.3.3 Módulo cinemático

O modelo cinemático do sistema mecatrônico em estudo foi implementado utilizando-se funções *S-function*, dentro do ambiente MatLab[®] integrado nos blocos do Simulink[®]. A Figura 5.14 apresenta a implementação da função do modelamento cinemático do sistema robótico móvel. O anexo C apresenta a rotina na sua íntegra. O desenvolvimento matemático do modelamento cinemático utilizado neste módulo, está descrito no capítulo 2, nas seções 2.3.1 e 2.3.2.

As variáveis de saída deste bloco são:

- \mathbf{x}_{din} . É um vetor de trajetória dinâmica do robô móvel no eixo x .
- \mathbf{y}_{din} . É um vetor de trajetória dinâmica do robô móvel no eixo y .

```

1 function [X,Y,V_roboto,R_c,Teta,W] = trajetoria(V_d,V_e,L,t,Teta_ant,X_ant,Y_ant)
2
3 % Monta os vetores X, Y e o Teta da trajetória do robô móvel
4 Delta_t = .01;
5
6 if V_d == V_e %trajetória reta
7     V_roboto = V_d;
8     Teta = Teta_ant;
9     %X = posição do roboto no eixo X
10    X = X_ant + V_roboto * cos(Teta) * Delta_t;
11    %Y = posição do roboto no eixo Y
12    Y = Y_ant + V_roboto * sin(Teta) * Delta_t;
13    R_c = inf;
14    W=0;
15 else %trajetória circular
16    %W = velocidade angular do robô em rad/s
17    W = (V_d - V_e) / L;
18    %R_c = raio de curvatura da trajetória do robô
19    R_c = (V_d + V_e) / (2 * W);
20    %Teta = deslocamento angular do roboto no tempo em rad
21    Delta_Teta = W * Delta_t;
22    Teta = Teta_ant + Delta_Teta;
23    %Centro de curvatura instantânea
24    CCix = X_ant - R_c * sin(Teta);
25    CCiy = Y_ant + R_c * cos(Teta);
26    %Velocidade linear do robô
27    V_roboto = ((V_d - V_e) * R_c) / L;
28    %Posição X e Y do robô
29    X = (X_ant - CCix) * cos(Delta_Teta) - (Y_ant - CCiy) * sin(Delta_Teta) + CCix;
30    Y = (X_ant - CCix) * sin(Delta_Teta) + (Y_ant - CCiy) * cos(Delta_Teta) + CCiy;
31 end
32

```

Figura 5.14: Implementação do modelo cinemático do robô móvel.

- \mathbf{V}_{din} . É um vetor velocidade dinâmica do robô móvel.
- \mathbf{R}_c . É um vetor raio de curvatura dinâmico do robô móvel.
- θ . É um vetor ângulo de posicionamento dinâmico do robô móvel.
- ω . É um vetor velocidade angular dinâmica do robô móvel.

5.4 Analisador gráfico de resultados

O simulador implementado em ambiente Simulink[®] permite a visualização temporal das saídas e entradas do sistema em estudo. Para melhor compreender e analisar o comportamento do sistema torna-se imprescindível a implementação de um analisador gráfico de resultados que é descrito a seguir. Assim, após as simulações realizadas no domínio do tempo, são obtidos arquivos de dados temporais correspondentes às variáveis de estudo (posição angular e cartesiana, velocidade linear e angular e sinais de controle), que após tratamento conveniente, torna-se possível verificar importantes resultados para melhor análise do comportamento do sistema. A Figura 5.15 ilustra o menu principal do analisador gráfico do sistema robótico móvel em estudo.



Figura 5.15: Menu principal do analisador gráfico de resultados do sistema robótico.

O módulo analisador gráfico de resultados permite uma análise completa do comportamento cinemático e dinâmico do sistema robótico móvel em estudo, inclusive dos erros. No menu principal pode-se optar pelos gráficos do deslocamento linear do robô, da trajetória cartesiana do robô, da velocidade linear do robô, do deslocamento angular do robô com relação ao CCI, da velocidade angular do robô com relação ao CCI, do raio de curvatura nas trajetórias circulares do robô com relação ao CCI, da velocidade linear nas rodas (motor direito e esquerdo), da velocidade angular nas rodas e, finalmente, os gráficos para análise dos erros.

As Figuras 5.16, 5.17 e 5.18 ilustram os outros sub-menus do sistema. Cada sub-menu tem suas funções, como geração de gráficos de deslocamento linear, angular, velocidades, erros, etc.



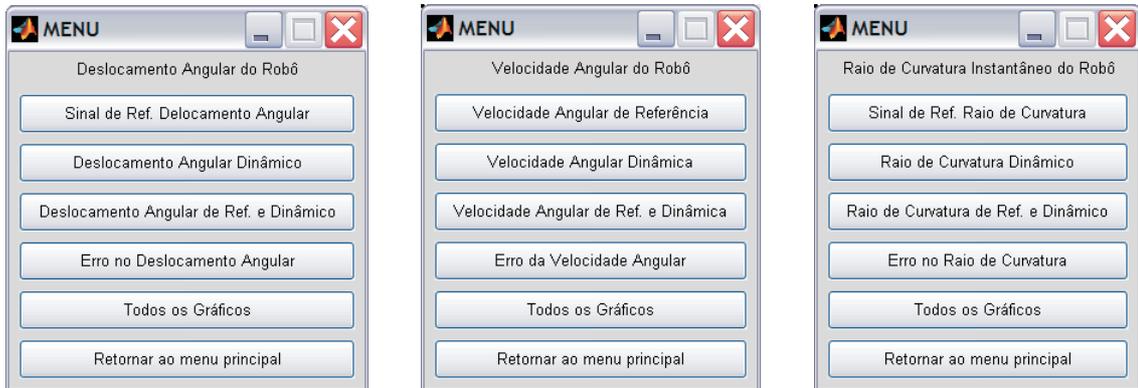
(a) Sub-menu deslocamento nos eixos cartesianos.

(b) Sub-menu da trajetória cartesiana.

(c) Sub-menu da velocidade linear do robô.

Figura 5.16: Sub-menus do sistema analisador gráfico.

A Figura 5.19 ilustra um gráfico com resultado de erros gerados na simulação do sistema robótico móvel. Maiores detalhes com exemplos esmiuçados são apresentados no capítulo 6 onde é feita a validação experimental de todo o sistema proposto.



(a) Sub-menu do deslocamento angular do robô.

(b) Sub-menu da velocidade angular do robô.

(c) Sub-menu do raio de curvatura do robô.

Figura 5.17: Sub-menus do sistema analisador gráfico.

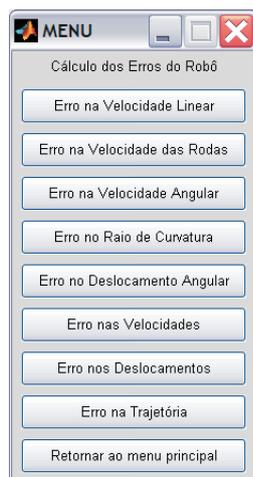


(a) Sub-menu da velocidade linear das rodas.

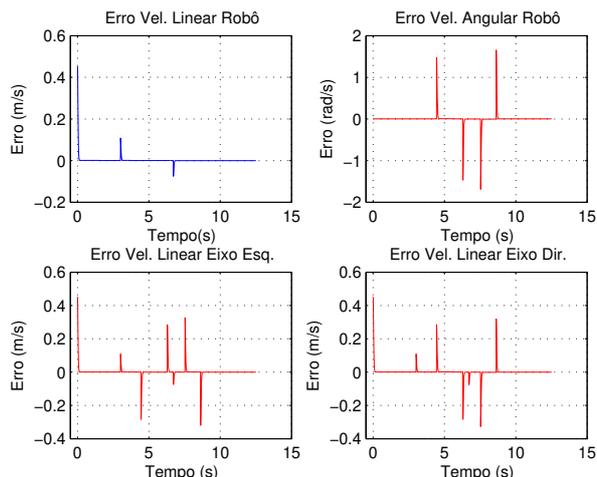
(b) Sub-menu da velocidade angular das rodas.

(c) Sub-menu estudos dos erros.

Figura 5.18: Sub-menus do sistema analisador gráfico.



(a) Submenu de análise dos erros do robô.



(b) Gráfico dos erros das velocidades do robô.

Figura 5.19: Submenu do módulo de estudos de erros do robô com um exemplo de gráfico gerado.

5.5 Prototipagem rápida

O termo prototipagem rápida, na sua versão tradicional da engenharia mecânica, designa um conjunto de tecnologias usadas para se fabricar objetos físicos diretamente a partir de fontes de dados gerados por sistemas de projeto auxiliado por computador (CAD¹). Tais métodos são bastante peculiares, uma vez que eles agregam e ligam materiais, camada a camada, de forma a constituir o objeto desejado. Eles oferecem diversas vantagens em muitas aplicações quando comparados aos processos de fabricação clássicos baseados em remoção de material, tais como fresamento ou torneamento (WOHLERS, 2001).

Segundo Wright (2001) a prototipagem rápida possibilita produzir modelos e protótipos diretamente a partir do modelo sólido 3D gerado no sistema CAD. Ao contrário dos processos de usinagem, que subtraem material da peça em bruto para se obter a peça desejada, os sistemas de prototipagem rápida geram a peça a partir da união gradativa de líquidos, pós ou folhas de papel. Camada por camada, a partir de seções transversais da peça obtidas a partir do modelo 3D, as máquinas de prototipagem rápida produzem peças em plásticos, madeira, cerâmica ou metais. Os dados para as máquinas de prototipagem podem ser gerados

¹Do inglês: *computer-aided design*

no sistema CAD no formato STL², que aproxima o modelo sólido por pequenos triângulos ou facetas. Quanto menor forem estes triângulos, melhor a aproximação da superfície, ao custo naturalmente de maior tamanho do arquivo STL e tempo de processamento. Um vez que o arquivo STL é gerado, as demais operações são executadas pelo próprio software que acompanha as máquinas de prototipagem rápida. Basicamente estes softwares irão, além de operações básicas de visualização, gerar as seções transversais do modelo que será construído. Tais dados são então descarregados para a máquina que irá depositar as camadas sucessivamente até que a peça seja gerada.

Tais métodos permitem aos projetistas criar rapidamente protótipos concretos a partir de seus projetos, ao invés de figuras bidimensionais. Esses modelos apresentam diversos usos. Eles constituem um auxílio visual excelente durante a discussão prévia do projeto com colaboradores ou clientes. Além disso, o protótipo pode permitir testes prévios como, por exemplo, ensaios em túnel de vento para componentes aeronáuticos ou análise fotoelástica para se verificar pontos de concentração de tensões na peça. A verdade é que os projetistas sempre construíram protótipos; os processos de prototipagem rápida permitem que eles sejam feitos mais depressa e de forma mais barata. De fato, estima-se que a economias de tempo e de custos proporcionada pela aplicação das técnicas de prototipagem rápida na construção de modelos sejam da ordem de 70 a 90% (GORNI, 2001).

As mesmas técnicas de prototipagem rápida podem ser usadas para a fabricação de ferramentas, um processo também conhecido como ferramentaria rápida, ou seja, a fabricação automática de ferramentas para uso na produção em série. A produção de ferramentas é uma das etapas mais lentas e caras no processo de manufatura, em função da qualidade extremamente alta que se exige delas. Ferramentas geralmente apresentam geometrias complexas e precisam ser dimensionalmente precisas, em torno de centésimos de milímetro. Além disso, elas devem ser duras, resistentes ao desgaste e apresentar baixa rugosidade. Por isso matrizes e moldes são tradicionalmente feitos por usinagem CNC, eletroerosão ou mesmo manualmente. Todos esses processos são caros e demorados, o que torna a implementação das técnicas de prototipagem rápida muito bem vinda. Estima-se que essas técnicas permitam economizar 75% do tempo e custos envolvidos na fabricação das ferramentas (DRIZO; PEGNA, 2006).

Elas também permitem a obtenção de peças com mesmo nível de qualidade da produção em série, na chamada manufatura rápida. De fato, a prototipagem rápida é o melhor processo

²STL é um formato de arquivo nativo dos *softwares* de estereolitografia em CAD, criados para os sistemas 3D. Este formato de arquivo computacional é suportado por vários outros sistemas de *software* e é muito utilizado nas técnicas de prototipagem rápida e CAM (*computer-aided manufacturing*).

de manufatura possível quando se precisa produzir pequenos lotes de peças e ou no caso de componentes complicados.

O termo “*rápido*” associado a esses processos é relativo. A construção de alguns protótipos pode levar de 3 a 72 horas, dependendo do tamanho e complexidade do objeto. Ainda assim esses processos são bem mais rápidos que os métodos tradicionais, tais como usinagem, que podem requerer dias ou mesmo meses para fabricar um único protótipo (DIMITROV et al., 2007).

5.6 Prototipagem rápida nos sistemas robóticos móveis

A utilização da técnica da prototipagem rápida nos sistemas robóticos móveis foge do escopo tradicional utilizado na engenharia mecânica, como pode ser visto na seção anterior, e entra em novo campo de pesquisa e desenvolvimento para projetos de sistemas mecatrônicos de robôs móveis. Neste sentido, a prototipagem rápida destes sistemas está associada não somente ao projeto do sistema físico, mas principalmente às implementações experimentais nos campos de *hardware* e *software* do sistema robótico. A arquitetura de *hardware* do sistema proposto é fundamental que seja aberta e flexível no sentido de se efetuar as modificações necessárias para otimização do sistema. Uma proposta de arquitetura aberta é apresentada no capítulo 3 (MELO; ROSÁRIO; LIMA, 2003). O *software* do sistema de controle embarcado do robô móvel, no contexto da prototipagem rápida, pode ser elaborado em simuladores e testados todos os parâmetros para ajustes que se fizerem necessários de acordo com o sistema físico a ser implementado, a arquitetura de hardware, os atuadores e os sensores. Desta feita, no contexto deste trabalho, a prototipagem rápida é então a metodologia que permite a criação de um ambiente virtual de simulação para o projeto de um controlador para robôs móveis. Depois de testado e validado no simulador, o sistema de controle é programado na memória da placa de controle do robô móvel. Desta maneira, uma economia de tempo e material é conseguida, validando primeiramente todo o modelo virtualmente para depois se operar a implementação física do sistema. A descrição da plataforma proposta é feita na seção 5.6.1.

Muitos pesquisadores já vêm utilizando a técnica da prototipagem rápida aplicada em sistemas robóticos móveis. MACHADO et al. (2002) descreve a implementação em MATLAB® de um ambiente de desenvolvimento rápido de aplicações (RAD) de navegação de robôs móveis baseadas em visão computacional. Neste trabalho é feita a descrição do desenvolvimento de um protótipo de software de controle para seguimento de objetos coloridos por um

robô móvel. Alguns detalhes da implementação, descritos neste artigo, permitiram a realização de testes em tempo real, mesmo sendo esta feita em código interpretado, o que geralmente acarreta em baixo desempenho quando comparado ao de uma linguagem compilada.

O desenvolvimento de técnicas de planejamento de trajetória, utilizando a abordagem da prototipagem rápida, que compõem um planejador seletivo num ambiente de realidade virtual é descrito por Aguinaga, Borro e Matey (2007) em seu trabalho. Os métodos utilizados têm sido adaptados e modificados, a partir de métodos disponíveis de planejamento de trajetória de robôs móveis, para utilização na reconstrução da trajetória.

Wisspeintener (2005) descreve um sistema robótico móvel chamado de VolksBot com construção modular utilizado em pesquisa e desenvolvimento na área acadêmica. O sistema utiliza a técnica da prototipagem rápida para a implementação de uma arquitetura com interfaces abertas de *hardware*, *software* e mecânica. A estrutura enxuta e aberta favorece a reusabilidade dos módulos, tanto em *hardware* como em *software*. São mostrados exemplos de como a técnica de prototipagem rápida nas variantes desta plataforma, podem ser utilizadas para adaptação da plataforma em novos campos de pesquisas urbanas, de resgate e de transportes autônomos.

Dudek e Jenkin (2000) descrevem uma alternativa de se implementar toda a computação embarcada que é processar a grande quantidade de dados nos sistemas supervisórios em vez de serem processados *onboard*. Sistemas embarcados que provêm um *link* de dados entre o sistema supervisório e o controle embarcado têm muitas vantagens. Uma delas é a redução do consumo de potência do controlador embarcado. Outra é a possibilidade de se utilizar processadores mais genéricos pois a maior parte do processamento pode ser feito nos processadores de maior porte do sistema supervisório (*offboard*). Isto permite que a técnica de prototipagem rápida para estes ambientes, produzam sistemas mais sofisticados em termos de programação.

Um projeto de um robô móvel adaptável que usa até três placas de controladores com servo motores e várias tecnologias de sensores é descrito por Hamblen, Furman e Hall (2006) em seu livro. Descrevem também a construção de robôs móveis de baixo custo utilizando a técnica da prototipagem rápida com utilização de componentes de hardware reprogramáveis, tais como PLDs e FPGAs, assim como a utilização de processadores em conjunto com FPGAs numa mesma placa de controle embarcado.

Ge e Lewis (2006) descrevem uma plataforma usada em robôs móveis chamada de ERSP (*Evolution Robotics Software Platform*). Este ambiente provê os componentes básicos e as

ferramentas necessárias para desenvolvimento rápido, prototipagem, e integração de aplicações voltadas aos sistemas robóticos móveis. A arquitetura de software que este sistema provê é chamada de ERSA (*Evolution Robotics Software Architecture*) sendo a infraestrutura e um dos principais componentes do ERSP. Esta estrutura em conjunto com técnicas de prototipagem rápida são explanados neste livro tendo como principal aplicação os sistemas robóticos móveis.

5.6.1 Descrição da plataforma

A Figura 5.20 ilustra a plataforma do protótipo do robô móvel com os elementos principais. A idéia é utilizar um sistema de arquitetura aberta e reconfigurável, conforme descrito no capítulo 3. Neste sentido, a proposta de implementação do *hardware* tem como dispositivo de processamento principal um microprocessador DSP TMS320C6416 da Texas Instruments, conforme descrito na seção 3.2. A utilização de dispositivos de *hardware* reconfigurável, tais como PLDs e FPGAs, tem sua fundamentação de escolha descrita na seção 3.1.

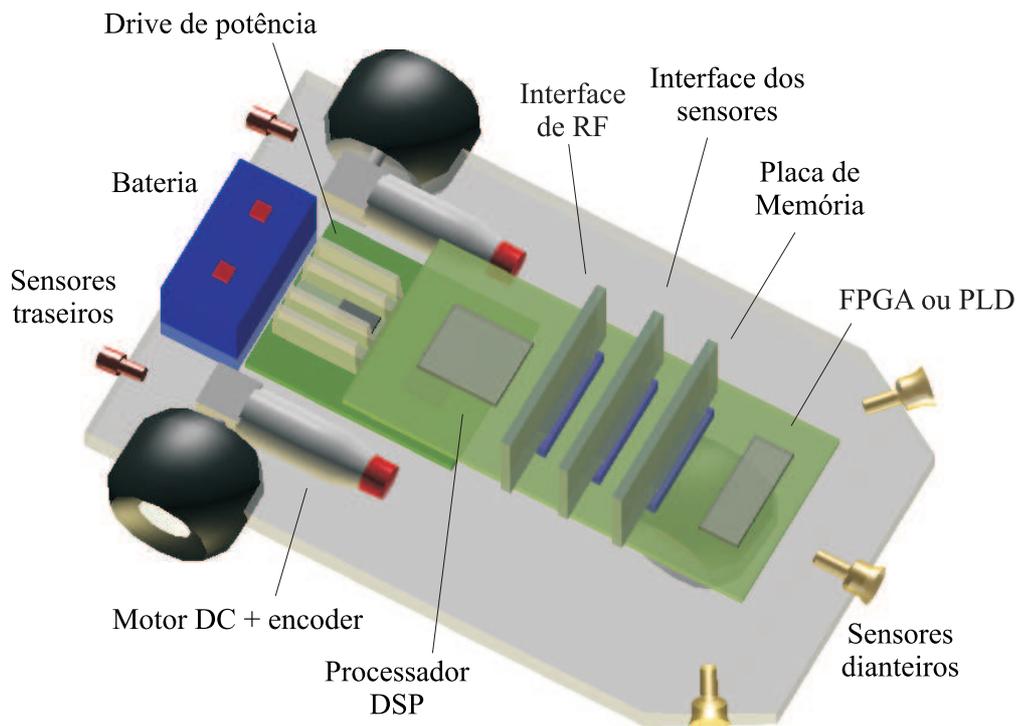


Figura 5.20: Plataforma do Robô móvel com os principais elementos.

A Figura 5.21 mostra o protótipo do robô móvel experimental que pode ser utilizado como plataforma de testes e validações deste trabalho. Todo modelamento cinemático e dinâmico

teve como base esta plataforma com os seus componentes elétricos e mecânicos.

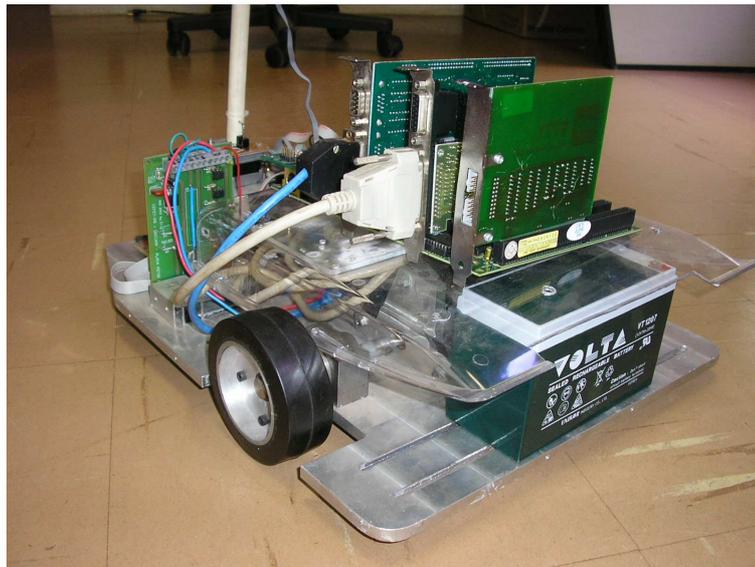


Figura 5.21: Protótipo do robô móvel experimental. LAIR - Laboratório de Automação Integrada e Robótica - FEM - UNICAMP.

A escolha do processador DSP TMS320C6416 como dispositivo gerenciador principal do sistema embarcado, recai em dois principais fatores. Primeiro, a sua grande capacidade de processamento de informações e instruções, operando na frequência de clock de 1 GHz e executando até 8 bilhões de informações por segundos na sua performance máxima, tornando o sistema eficiente o suficiente para implementações de técnicas de *software* complexas, requeridas nos sistemas de navegação robóticas. Em segundo porque tem um *tool box* específico que pode ser utilizado no MatLab[®] para sua programação em tempo real com técnicas de *hardware-in-the-loop* (HIL), que é uma das técnicas de simulação utilizada nos sistemas de prototipagem rápida para controladores robóticos móveis embarcados. Esta nova técnica de simulação HIL (anteriormente só disponíveis na indústria aeroespacial e aeronáutica) pode ser utilizada para o desenvolvimento e parametrização de controladores de sistemas embarcados (LEDIN, 2001).

5.6.2 Simulação *hardware-in-the-loop*

A simulação HIL é uma forma de simulação em tempo real. Ela difere das simulações em tempo real tradicionalmente encontradas, pela adição de um componente “real” no laço de testes. Este componente pode ser uma unidade de controle eletrônico (*ECU-Electronic*

controle unit) utilizado em sistemas embarcados automotivos ou aeroespaciais, um motor automotivo real, uma placa de controle embarcado utilizada em sistemas robóticos móveis, etc. A Figura 5.22 ilustra a utilização da técnica de simulação HIL para simulação em tempo real do sistema robótico móvel proposto.

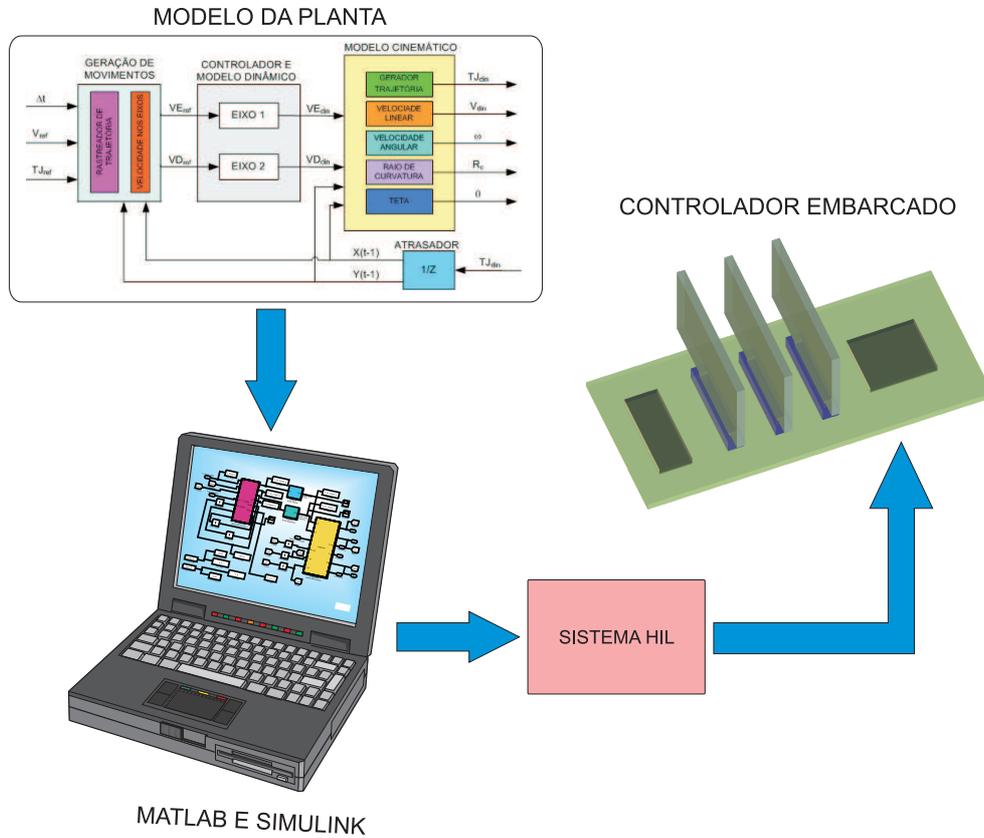


Figura 5.22: Simulação HIL do sistema robótico móvel.

A técnica de simulação HIL é utilizada no desenvolvimento e testes em tempo real de sistemas embarcados. As simulações HIL provêm uma plataforma efetiva de desenvolvimento por adicionarem a complexidade da planta sob controle à plataforma de testes. O sistema de controle é incluso nos testes e desenvolvimentos através de seus modelos de representações matemáticas e todos os respectivos modelamentos dinâmicos (AMBRÓSIO, 2005).

A simulação HIL inclui também a emulação elétrica de sensores e atuadores que atuam como interface entre a simulação da planta e o sistema embarcado que está sendo testado. O valor de cada sensor emulado eletricamente é controlado pelo simulador da planta e é utilizado pelo sistema embarcado. Outrossim, o sistema embarcado implementa os algoritmos de controle agindo nos atuadores do sistema. Variações nos sinais de controle resultam em variações nos valores das variáveis do simulador da planta (PAGNOTTELLI; VALIGI, 2006).

Em muitos casos, o modo mais efetivo de se desenvolver sistemas embarcados é conectando-se o sistema embarcado à planta real. Nestes casos, a simulação HIL é mais eficiente. O desenvolvimento e a eficiência dos testes são tipicamente enquadrados na fórmula que inclui os seguintes fatores: custo, duração e segurança. O custo da abordagem será uma medida do custo de todas as ferramentas e esforços. A duração do desenvolvimento e testes afeta no tempo de mercado para o planejamento comercial de um certo produto. O fator segurança e duração são tipicamente equacionados por uma medida de custo. Condições específicas que garantem o uso da simulação HIL incluem:

Prazos curtos para o desenvolvimento. Os prazos de desenvolvimentos de novos produtos nas indústrias automotivas, aeroespaciais e programas de defesa estão cada vez mais curtos e não permitem que sistemas embarcados em testes esperem até que seja feita uma avaliação final do protótipo. Na realidade, a maioria dos prazos de desenvolvimentos de novos produtos assumem que a simulação HIL será utilizada em paralelo com o desencaminhamento da planta. Por exemplo, enquanto um novo protótipo de motor de automóvel se torna disponível para testes de sistema de controle, 95% dos testes do controlador já estão sendo completados pela simulação HIL.

Alto custo da planta. Na maioria dos casos, a implementação física do sistema é muito mais cara do que um simulador de tempo real. Por conseguinte, é mais economicamente viável o desenvolvimento e teste de sistemas embarcados conectados a um simulador HIL do que a construção completa do protótipo.

Desenvolvimento de sistemas com fatores humanos (ergonômicos). A simulação HIL é muito importante no processo de desenvolvimento de sistemas ergonômicos. É um método que pode garantir a usabilidade e consistência de sistemas ergonômicos. Para a tecnologia em tempo real, o desenvolvimento de fatores humanos é a tarefa de se coletar e testar dados de componentes que terão uma interface com interação humana.

5.6.3 Prototipagem de controle rápida

A prototipagem de controle rápida (RCP - *Rapid Control Prototyping*) é um processo que permite ao engenheiro ou ao desenvolvedor um teste rápido e interativo das estratégias de controle, num sistema computacional em tempo real, com dispositivos de entrada e saída de dados reais. RCP é uma variante da simulação HIL, mas é distinto e popular o suficiente para garantir seu próprio nome e acrônimo. A diferença fundamental da técnica RCP em

comparação com a técnica HIL é que, no primeiro, a estratégia de controle é simulada em tempo real e a planta, ou sistema a ser controlado, é real. Na simulação HIL, a planta é simulada mas o controlador é real (ABEL; BOLLIG, 2006). A Figura 5.23 ilustra o sistema RCP utilizado no contexto do simulador do robô móvel apresentado neste trabalho.

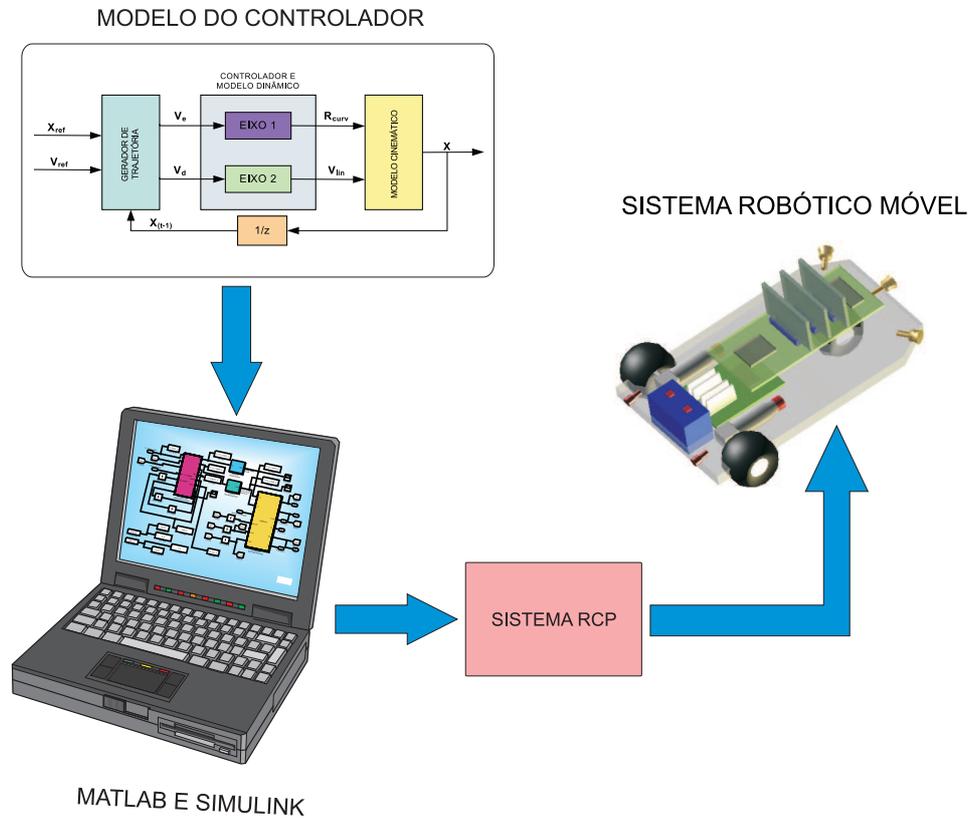


Figura 5.23: Sistema de prototipagem de controle rápida (RCP).

5.7 Considerações finais

Neste capítulo foram descritas as principais características do sistema simulador para robôs móveis. As bibliotecas desenvolvidas possibilitam facilmente adicionar novos módulos, aumentando o número de graus de liberdade do robô móvel, modificando o tipo de modelo cinemático do robô móvel, o tipo de controlador, a dinâmica e acoplamento dos eixos e outras características julgadas pertinentes à simulação destes sistemas mecatrônicos. Apresentou-se também as principais ferramentas utilizadas para o seu desenvolvimento. O sistema simulador foi idealizado em arquitetura modular e aberta em ambiente MatLab/Simulink®, facilitando a criação e modificação das bibliotecas nele utilizadas.

O módulo de controle foi implementado na forma PID, onde os parâmetros relativos a esse controlador foram calculados a partir das características dinâmicas do sistema em estudo, tornando possível, após sua validação e testes, ser implementado num ambiente de prototipagem rápida. Para tanto, pode-se utilizar o sistema de arquitetura aberta proposto no capítulo 3.

Conforme também pode ser visto neste capítulo, fica evidente as vantagens da utilização da técnica de simulação HIL e RCP na prototipagem rápida de sistemas robóticos móveis. Esta abordagem de desenvolvimento de controladores embarcados é feita neste trabalho de forma que todo o sistema seja testado e aprovado no simulador antes de ser implementado do robô móvel.

Para que o sistema tenha flexibilidade em suas alterações é necessário que sua arquitetura seja aberta e reconfigurável. Neste sentido, uma arquitetura que vem satisfazer as necessidades operacionais do sistema é apresentada no capítulo 3.

As técnicas de prototipagem rápida só podem ser executadas se a planta tiver seu modelamento cinemático, dinâmico e do controlador implementados num simulador virtual. Desta feita, o modelamento cinemático da planta está detalhado no capítulo 2, mais especificamente na seção 2.3.1. O modelamento dinâmico e do controlador é apresentado no capítulo 4. O *software* MatLab[®] provê as ferramentas computacionais necessárias para que todo sistema de simulação e prototipagem rápida seja implementado.

Capítulo 6

Validação experimental

Este capítulo tem por finalidade a validação experimental de parte do trabalho, apresentando resultados experimentais com exemplos de simulações, envolvendo diferentes situações de navegação de um sistema robótico móvel. A validação foi executada em duas etapas distintas, a primeira feita com o simulador virtual obtendo-se análise gráfica de resultados da simulação. A segunda é feita uma validação experimental cinemática onde são apresentados resultados experimentais através da implementação de um exemplo de trajetória cinemática de um robô móvel. Esta validação foi executada em robôs móveis do Laboratório de Robôs Móveis Autônomos da Faculdade de Ciências da UNESP-Bauru.

6.1 Analisador gráfico de resultados da simulação

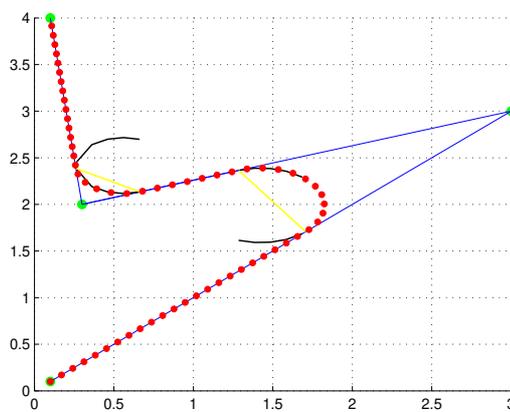
O simulador implementado em ambiente Simulink[®] permite a visualização temporal das saídas e entradas do sistema em estudo. Para melhor compreender e analisar o comportamento do sistema torna-se imprescindível a implementação de um analisador gráfico de resultados que é descrito a seguir. Assim, após as simulações realizadas no domínio do tempo, são obtidos arquivos de dados temporais correspondentes as variáveis de estudo (posição angular e cartesiana, velocidade linear e angular e sinais de controle), que após tratamento conveniente, torna-se possível verificar importantes resultados para melhor análise do comportamento do sistema.

A seguir, são apresentados dois estudos de casos onde são feitas as análises de desempenho e de erros do sistema robótico em questão.

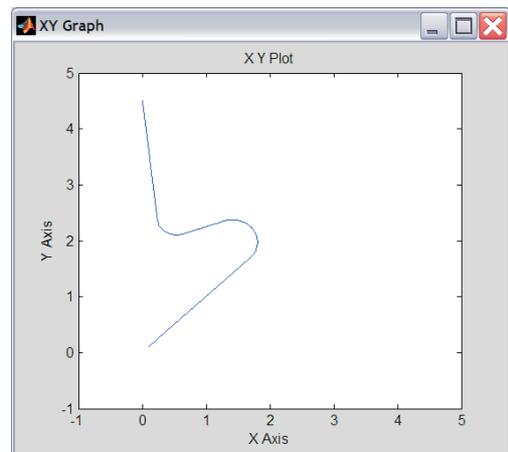
6.2 Primeiro estudo de caso

Neste primeiro exemplo da utilização do sistema simulador, far-se-á uma análise do comportamento do sistema robótico móvel seguindo uma trajetória inicial de referência ilustrado na Figura 6.1(a).

Todos os parâmetros de modelamento cinemático e dinâmico do controlador são ajustados para um modelo de robô móvel não holonômico com tração diferencial nas rodas traseiras, conforme descrito nos capítulos anteriores. A Figura 6.1(b), ilustra a trajetória dinâmica obtida pelo simulador após transcorrido o tempo final de simulação.



(a) Trajetória inicial de referência.



(b) Trajetória dinâmica gerada pelo simulador.

Figura 6.1: Trajetória de referência e dinâmica do robô móvel. Primeiro estudo de caso.

6.2.1 Velocidade linear do robô

Para este primeiro estudo de caso, tomou-se a velocidade linear de referência como uma constante no valor de 0,5 m/s. Os gráficos da Figura 6.2, mostram a velocidade linear de referência do robô, a velocidade linear dinâmica do robô e o erro. Nota-se a atuação eficaz do controlador no sentido da estabilização pronta e eficiente da velocidade, com a minimização do erro e a maximização da eficiência.

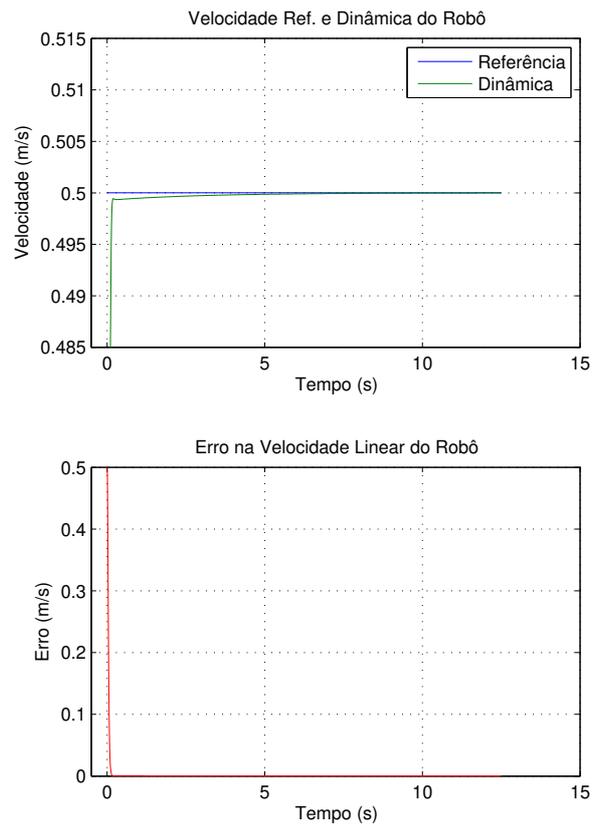


Figura 6.2: Gráficos da velocidade linear de referência e velocidade linear dinâmica do robô e do erro.

6.2.2 Deslocamento linear nos eixos cartesianos

A primeira análise que é feita é com relação ao deslocamento linear do robô nos eixos X e Y. Através da Figura 6.3, pode-se ver então, qual o comportamento dinâmico do robô com relação a estes parâmetros, assim como os erros apresentados.

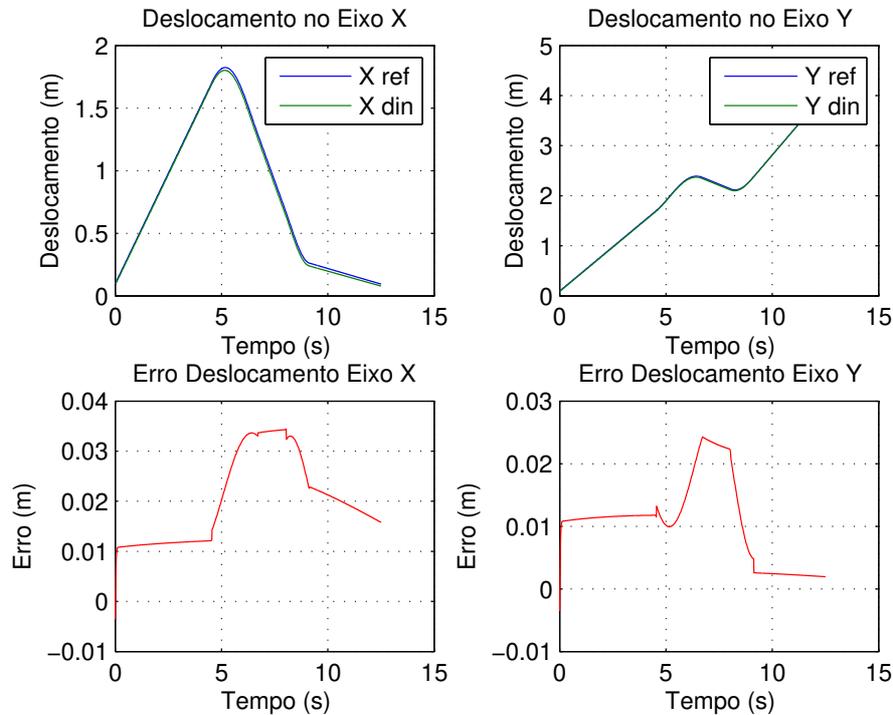
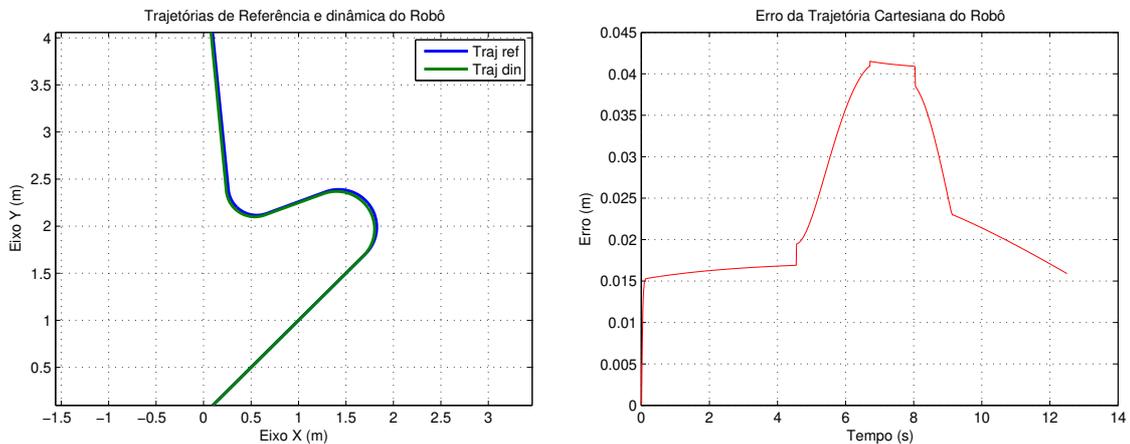


Figura 6.3: Gráficos do deslocamento linear de referência e dinâmico nos eixos cartesianos, com seus respectivos erros.

Percebe-se que um erro maior tanto no deslocamento do eixo X, como no eixo Y, ocorre no intervalo de tempo aproximado de 7 a 8 segundos da simulação. Neste intervalo de tempo, o robô termina sua primeira curva a esquerda e, depois de um pequeno trecho em linha reta, começa uma segunda curva a direita. Nota-se que estes erros têm seus valores menores do que 4 cm durante todo o trajeto, estando dentro de uma boa margem de aceitabilidade. Como a simulação leva em conta que o robô não utiliza seus dados sensoriais para correção da trajetória, as tendências dos erros passam a ser cumulativas. Mesmo neste sentido, o sistema responde bem dinamicamente e consegue executar a trajetória com êxito, minimizando os erros.

6.2.3 Trajetória cartesiana do robô

Outro importante gráfico gerado pelo sistema, no sub-menu *trajetória cartesiana*, é o gráfico da trajetória cartesiana cinemática e dinâmica do sistema robótico móvel no plano XY. A Figura 6.4(a), mostra o traçado de referência e dinâmico da trajetória do robô móvel. A Figura 6.4(b) ilustra o gráfico do erro da trajetória. Nota-se que o erro maior na trajetória acontece no intervalo de tempo de simulação que varia de 6 a 8 segundos, corroborando os resultados obtidos na seção 6.2.2. Percebe-se também que o erro máximo da trajetória fica abaixo de 4,5 cm, justamente num momento crítico da execução da trajetória, que é a primeira curva. Importante notar a recuperação do sistema, apontado pelo gráfico do erro, onde o erro da trajetória tende a diminuir de intensidade. Fica claro então o excelente desempenho do sistema, resultando num nível de erro aceitável para o sistema, uma vez que a execução da trajetória foi feita com êxito, conforme ilustrado na Figura 6.4(a).



(a) Gráfico da trajetória cartesiana cinemática e dinâmica do robô.

(b) Erro de trajetória.

Figura 6.4: Trajetória de referência e dinâmica do robô móvel com o erro apresentado.

6.2.4 Deslocamento angular do robô

A Figura 6.5, ilustra um exemplo do resultado gráfico do deslocamento angular do robô (θ). São apresentados os resultados cinemáticos e dinâmicos do sistema assim como o erro ao longo do tempo. Nota-se que o erro em θ tem valores em módulo um pouco acima de 0,05 rad nos momentos das execuções das curvas, onde o gráfico de θ descreve uma reta inclinada. Onde a $\frac{d\theta}{dt} \rightarrow 0$ o erro também tende a zero, estabilizando destarte a trajetória do robô.

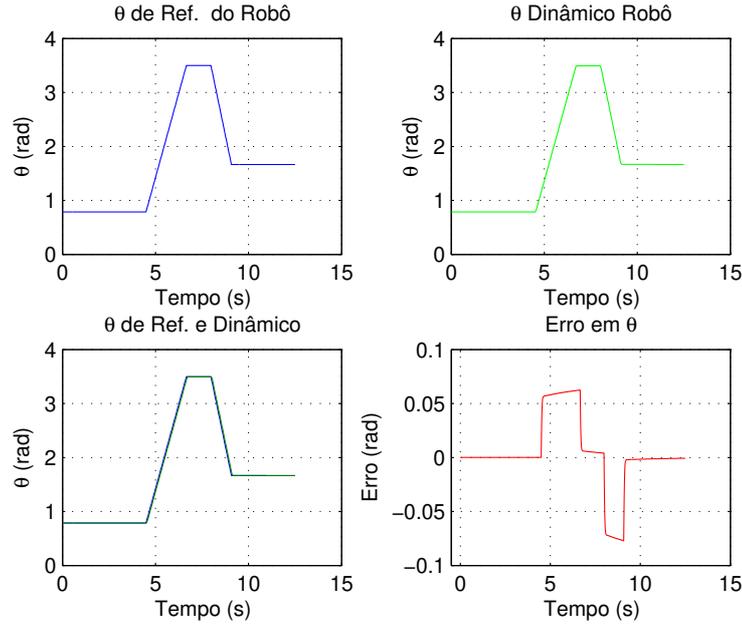


Figura 6.5: Gráficos do deslocamento angular de referência e dinâmico do robô com o respectivo erro.

6.2.5 Velocidade angular do robô

Na Figura 6.6, têm-se quatro gráficos relativos à velocidade angular executada pelo robô durante a trajetória dinâmica. Nos três primeiros gráficos observa-se a velocidade angular de referência e dinâmica do robô. No último gráfico, tem-se o erro. Nota-se que os erros encontrados são em forma de pulsos nos momentos exatos do começo e final de cada curva, resultante da atuação do controlador no sistema dinâmico. Percebe-se também, observando-se o gráfico do erro na velocidade angular do robô, que os valores tendem a zero mesmo quando o robô está executando uma trajetória curvilínea, comprovando assim a excelente estabilidade do sistema proporcionada pelo controlador, para este exemplo em questão.

6.2.6 Raio de curvatura do robô

A Figura 6.7, mostra os gráficos do raio de curvatura de referência e dinâmico do robô móvel com relação ao CCI. Vale observar que um raio de curvatura muito grande (e.g. $R_c > 1000$ m) resulta em uma trajetória praticamente retilínea em ambientes com extensões reduzidas (e.g. interiores de edifícios, nas ruas, etc.), que são os ambientes proposto para a utilização do sistema. Desta feita, pode-se notar grandes erros entre o sinal de referência e a atuação dinâmica do robô, embora o mesmo realize a trajetória a contento, conforme

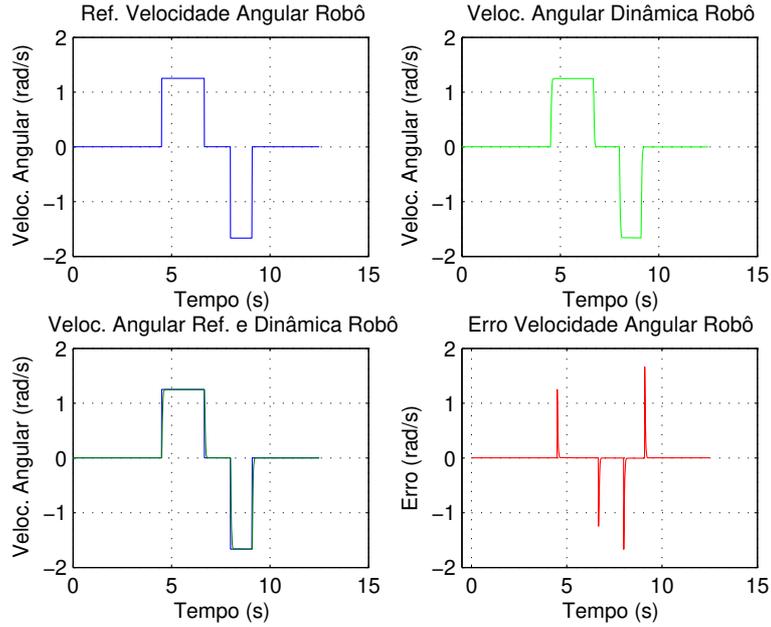


Figura 6.6: Gráficos da velocidade angular de referência e dinâmica do robô e o gráfico do respectivo erro.

ilustrado na Figura 6.4(a). Esta técnica de utilização de raio de curvatura com valores relativamente grandes é usada para fins de cálculos matemáticos uma vez que o valor de $R_c = \infty$ resultaria em indeterminações em algumas fórmulas cinemáticas do sistema.

6.2.7 Velocidade linear nas rodas

A Figura 6.8, apresenta um gráfico do sinal de referência da velocidade linear da roda esquerda (motor CC esquerdo) com sua respectiva evolução dinâmica. No segundo gráfico desta mesma figura, observa-se o sinal de erro deste eixo. Como foi comentado na seção 6.2.5, nota-se que os erros encontrados são em forma de pulsos nos momentos exatos do começo e final de cada curva, resultante da atuação precisa do controlador no sistema dinâmico. Vale observar a estabilidade do sistema, com velocidade linear dinâmica muito próxima da sua referência. Não foi apresentada aqui a análise do eixo direito pois apresenta o mesmo perfil em termos de controle do descrito no eixo esquerdo.

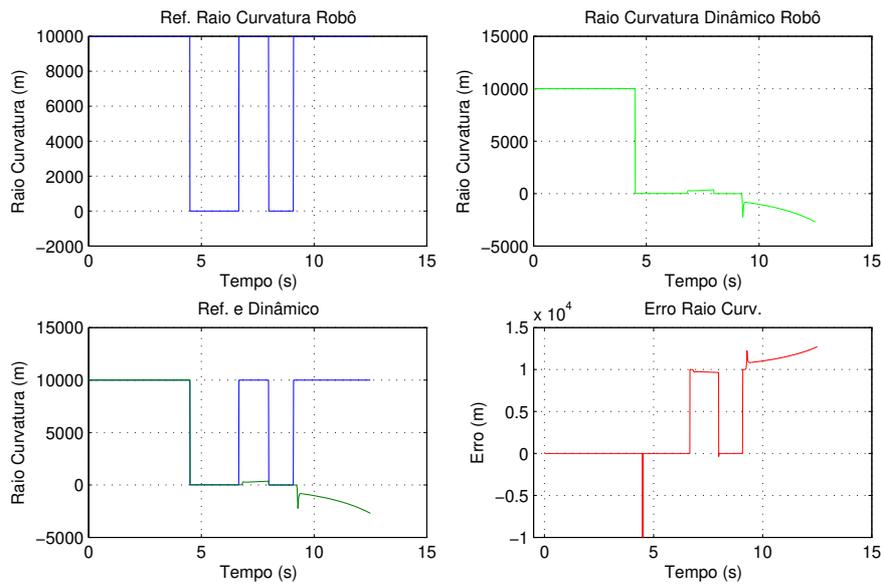


Figura 6.7: Gráficos do raio de curvatura de referência e dinâmica do robô e do respectivo erro.

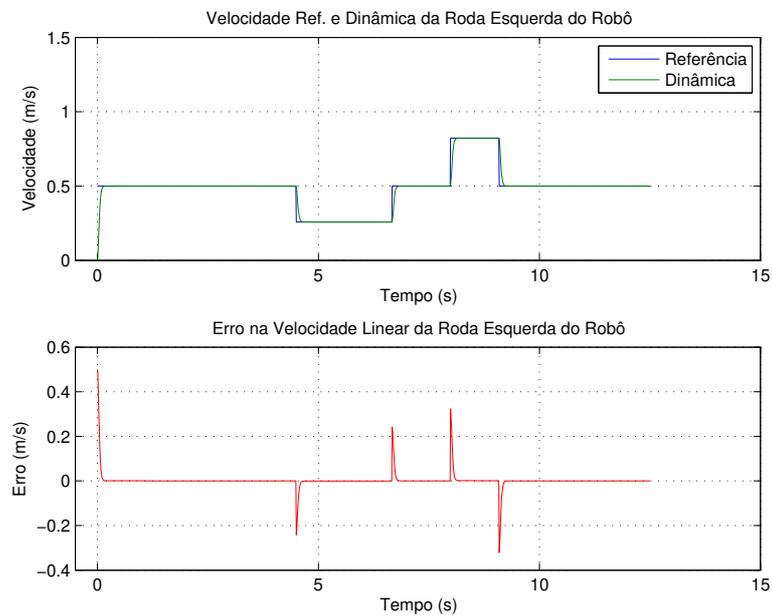


Figura 6.8: Gráfico da velocidade linear na roda esquerda e do erro.

6.2.8 Estudo dos erros

Uma das possibilidades de análise de erros oferecida pelo sistema analisador de resultados é mostrada na Figura 6.9, onde tem-se uma análise geral dos erros nas velocidades do robô. Esta abordagem analítica dos erros é particularmente interessante quando se deseja parametrizar o comportamento dinâmico dos subsistemas do robô móvel.

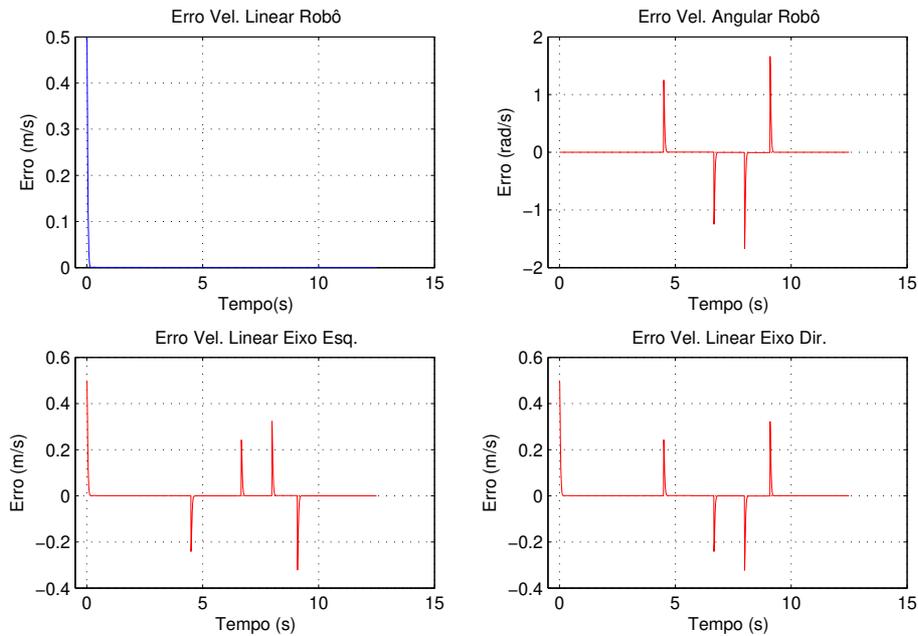


Figura 6.9: Gráficos da análise dos erros nas velocidades do robô.

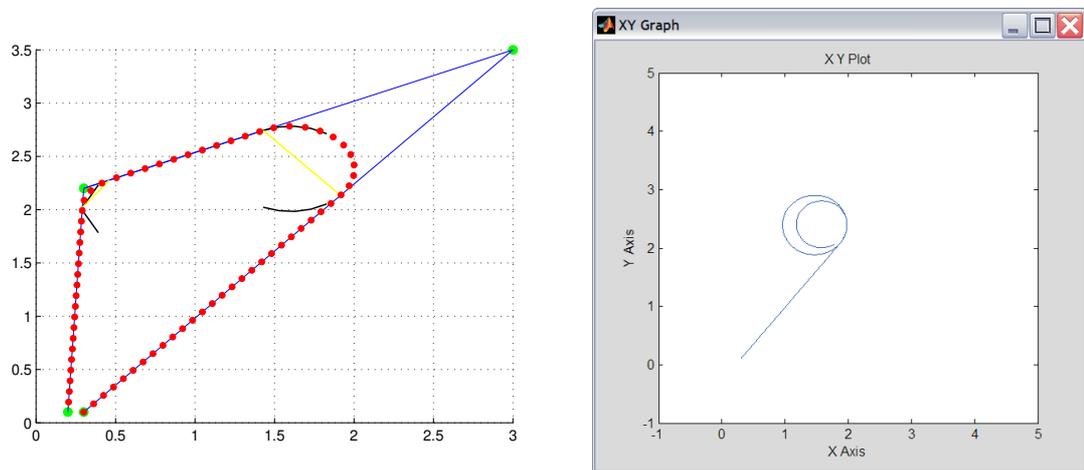
6.2.9 Conclusões

Neste estudo de caso comprovou-se, através das análises gráficas apresentadas, a eficácia do controlador adotado para o sistema robótico móvel proposto. Comprovou-se também, a utilidade do módulo analisador gráfico na análise dos parâmetros cinemáticos e dinâmicos adotados no sistema. Com uma velocidade linear constante aplicada ao sistema robô móvel, todos os resultados foram satisfatórios, atendendo perfeitamente as expectativas do projeto do controlador, descrito no capítulo 4. Assim sendo, o sistema simulador, descrito no capítulo 5, demonstrou sua aplicabilidade no sentido de prover todas as variáveis cinemáticas e dinâmicas do sistema em simulação em tempo real, deixando tudo pronto para a prototipagem rápida no sistema robótico móvel.

Na próxima seção é apresentado um segundo estudo de caso, com velocidade linear não constante.

6.3 Segundo estudo de caso

Neste segundo exemplo da utilização do sistema simulador, far-se-á uma análise do comportamento do sistema robótico móvel seguindo uma trajetória inicial de referência ilustrado na Figura 6.10(a). Neste estudo de caso, o robô não consegue seguir a trajetória pre-determinada e se perde em *loops* circulares. O motivo desta trajetória não concluída é o fato da velocidade linear de referência ser maior do que a máxima permitida pelo modelo, em alguns trechos do percurso, como é melhor explicado na seção 6.3.1. A Figura 6.10(b) ilustra a trajetória dinâmica obtida pelo simulador após transcorrido o tempo final de simulação.



(a) Trajetória inicial de referência.

(b) Trajetória dinâmica gerada pelo simulador.

Figura 6.10: Trajetória de referência e dinâmica do robô móvel. Segundo estudo de caso.

6.3.1 Velocidade linear do robô

A Figura 6.11 ilustra o sinal de referencia utilizado como entrada da velocidade linear do robô móvel.

A Figura 6.12 ilustra um dos resultados gráficos possíveis através do sub-menu *velocidade linear do robô*. Pode-se analisar o comportamento da velocidade linear dinâmica do robô

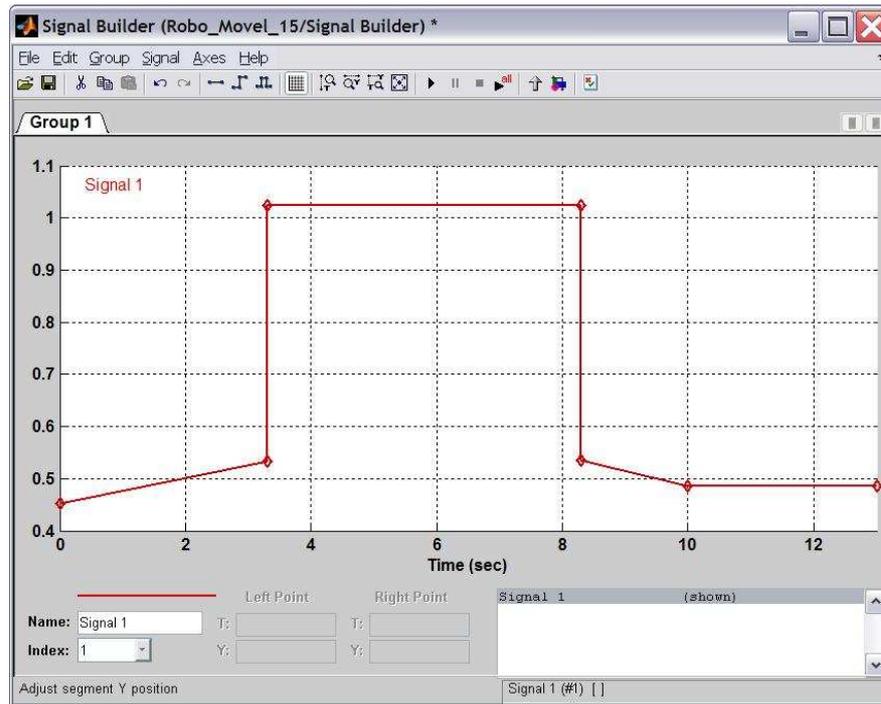


Figura 6.11: Sinal de referência da velocidade linear do robô.

móvel com relação ao sinal de referência, assim como o erro da velocidade linear ao longo do tempo. Uma conclusão interessante que se chega ao analisar estes gráficos da Figura 6.12, é que o primeiro e o segundo *spike* de erro, nos tempos aproximados de 0 s e 4 s respectivamente, acontecem devido a inércia dinâmica do sistema. O erro mais grosseiro, gerando um forma de onda retangular começando no tempo aproximado de 4 s, é devido ao sinal de referência da velocidade angular do eixo 1 (motor direito) ter ultrapassado o valor máximo suportado pelos motores CC, que é de 300 rad/s, como pode ser visto e analisado nos outros gráficos. Quando isto ocorre, a trajetória do robô pode ter um erro fatal e o mesmo ficar em *loop* até o final da simulação, conforme ilustrado na Figura 6.13.

Outra possibilidade de apresentação de resultados é mostrado na Figura 6.14, onde as formas de onda da velocidade linear dinâmica do robô móvel, o sinal de referência da velocidade linear, assim como o erro, estão dispostos em quatro gráficos distintos na mesma figura para uma melhor análise do conjunto.

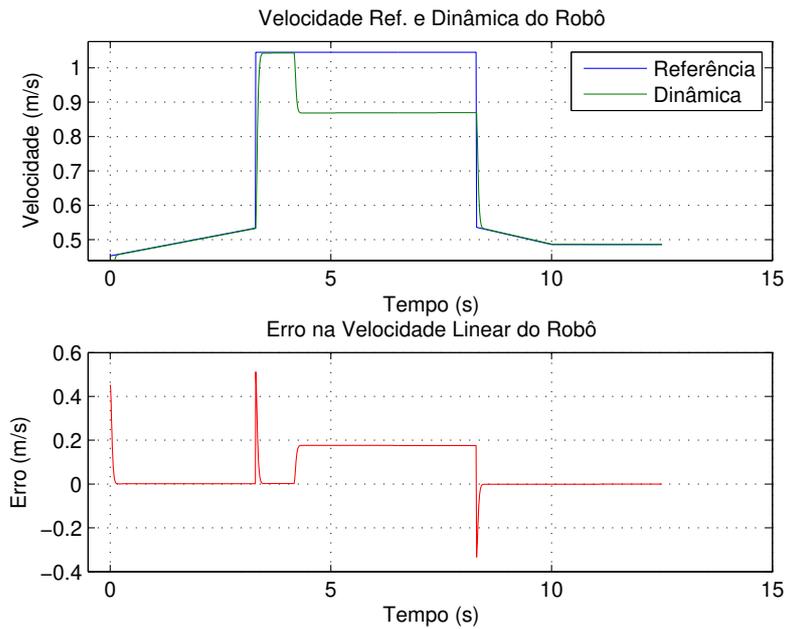
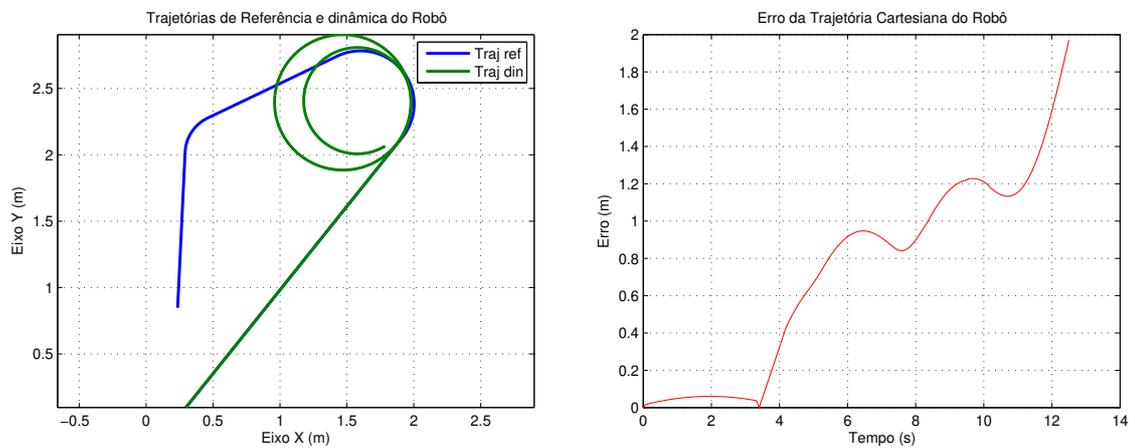


Figura 6.12: Gráficos da velocidade linear do robô e do erro respectivo.



(a) Trajetória dinâmica com erro.

(b) Gráfico do erro na trajetória do robô.

Figura 6.13: Erro de trajetória causada por uma simulação de velocidade linear acima da máxima permitida.

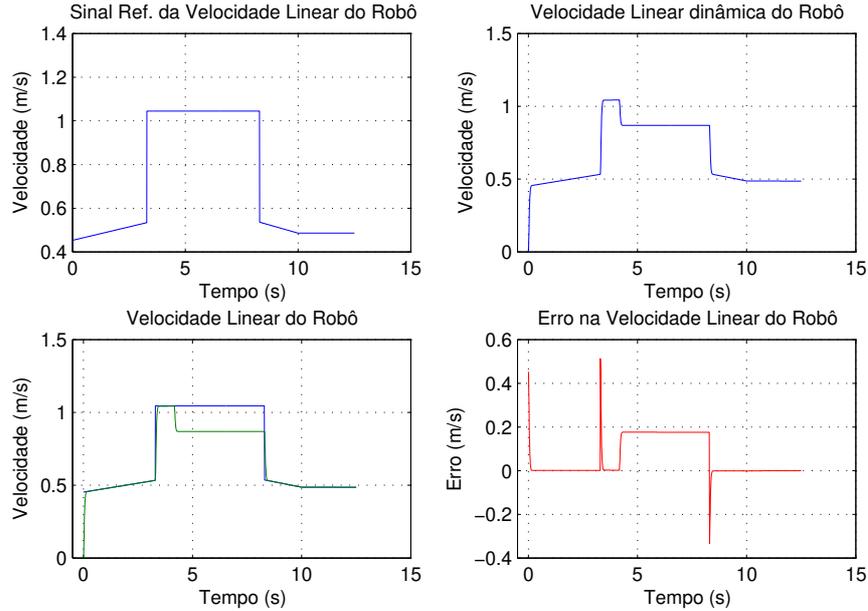


Figura 6.14: Gráficos da velocidade linear do robô e do erro correspondente.

6.3.2 Deslocamento angular do robô

A Figura 6.15, ilustra um exemplo do resultado gráfico do deslocamento angular do robô (θ). São apresentados os resultados cinemáticos e dinâmicos do sistema assim como o erro ao longo do tempo. Percebe-se que, a partir do tempo igual a 4,17 s, o robô começa a acumular um erro gradativo, uma vez que começa a realizar a primeira curva com velocidade em uma das rodas inferior ao sinal de referência.

6.3.3 Velocidade angular do robô

Na Figura 6.16 têm-se quatro gráficos relativos a velocidade angular executada pelo robô durante a trajetória dinâmica. Nos três primeiros gráficos observa-se a velocidade angular de referência e dinâmica do robô. No último gráfico, tem-se o erro. Tem-se que, a velocidade angular acima da máxima permitida pelos parâmetros de modelamento do robô móvel é cortada, o que pode ser observado na Figura 6.16. Neste exemplo de modelo de robô móvel, a velocidade angular máxima é de aproximadamente 1,7 rad/s.

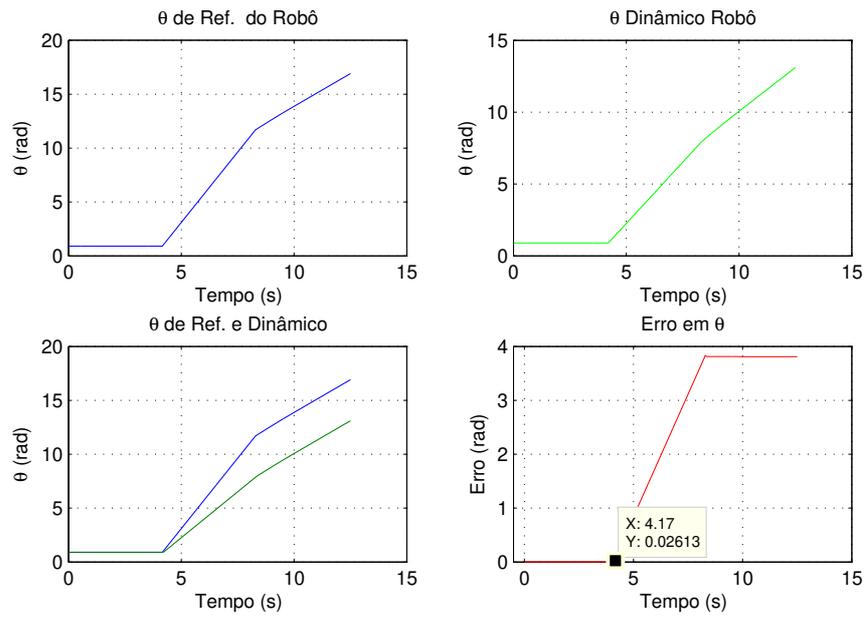


Figura 6.15: Gráficos do deslocamento angular de referência e dinâmico do robô com o respectivo erro.

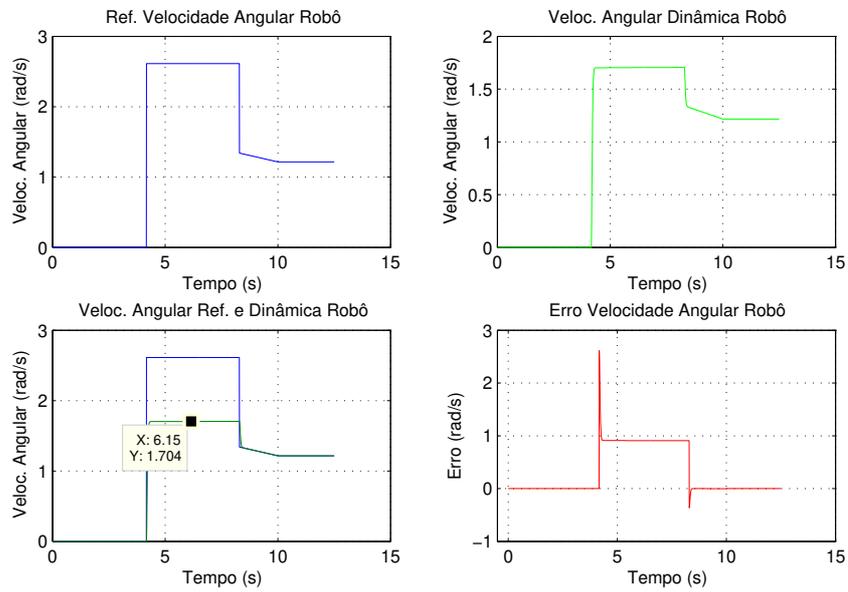


Figura 6.16: Gráficos da velocidade angular de referência e dinâmica do robô e do respectivo erro.

6.3.4 Raio de curvatura do robô

A Figura 6.17, mostra os gráficos do raio de curvatura de referência e dinâmico do robô móvel com relação ao CCI.

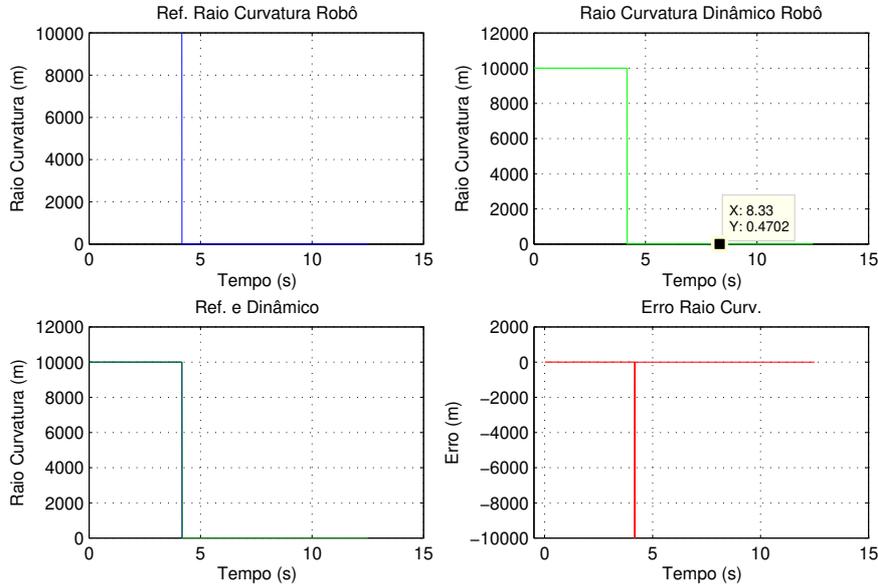
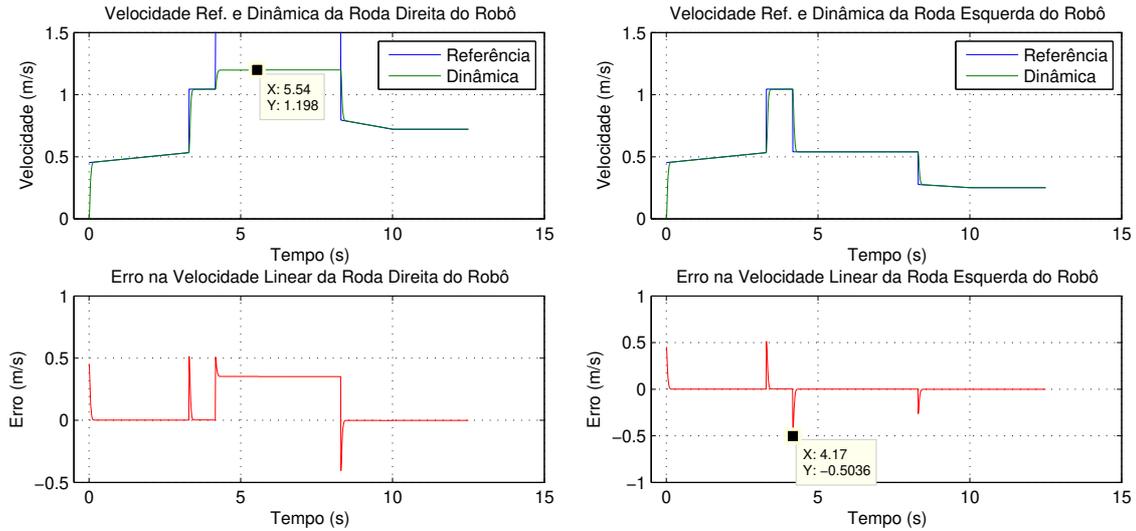


Figura 6.17: Gráficos do raio de curvatura de referência e dinâmica do robô e do respectivo erro.

6.3.5 Velocidade linear nas rodas

A Figura 6.18(a), apresenta um gráfico do sinal de referência da velocidade linear da roda direita (motor CC direito) com sua respectiva evolução dinâmica. No segundo gráfico desta mesma figura, observa-se o sinal de erro deste eixo. A Figura 6.18(b), apresenta um gráfico do sinal de referência da velocidade linear da roda esquerda (motor CC esquerdo) com o respectivo erro. Como foi comentado na seção 6.3.1, percebe-se claramente um sinal maior de erro na roda direita que acontece a partir do tempo de 4,16 s. Isto é devido a velocidade linear de referência da roda direita ter excedido o seu máximo, tendo sido cortada na sua resposta dinâmica. Não obstante, o eixo da roda esquerda responde perfeitamente ao sinal, uma vez que a curva se opera à esquerda e, desta monta, esta roda fica com a menor parcela da velocidade linear final.



(a) Velocidade linear na roda direita e seu erro. (b) Velocidade linear na roda esquerda e seu erro.

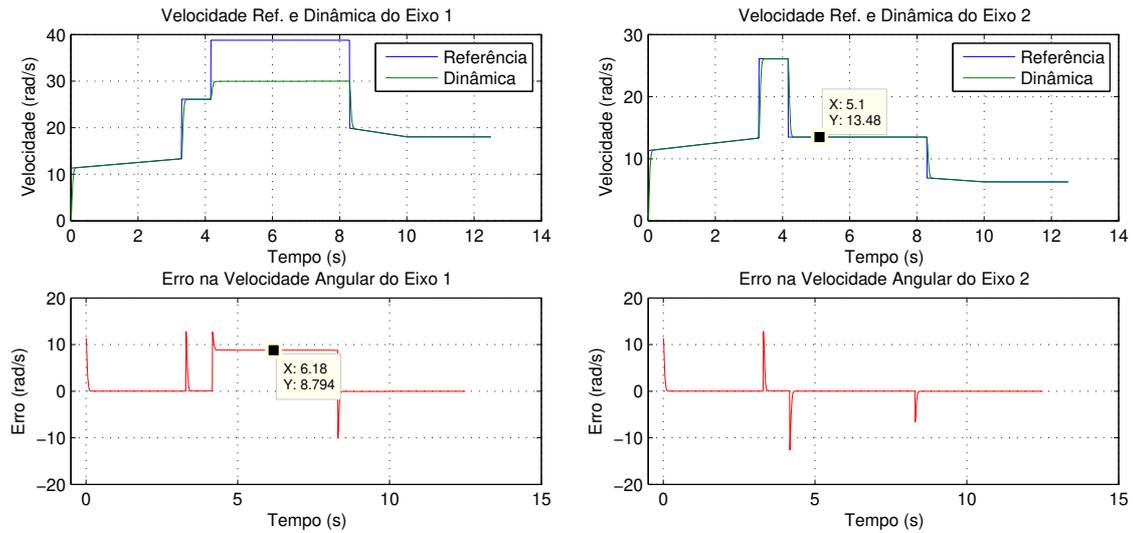
Figura 6.18: Velocidade linear nas rodas direita e esquerda com os respectivos erros dinâmicos.

6.3.6 Velocidade angular nos eixos

Os gráficos da velocidade angular nos eixos 1 e 2, rodas direita e esquerda respectivamente, podem ser observados na Figura 6.19. O gráfico superior destas figuras ilustra a velocidade angular de referência e dinâmica e o gráfico inferior o erro. Observa-se, mais uma vez na Figura 6.19(a), o corte na velocidade angular dinâmica na roda direita em 30 rad/s. Isto acontece pois a velocidade angular máxima permitida nas rodas é de 30 rad/s, uma vez que a velocidade angular máxima nos eixos dos motores é de 300 rad/s com um fator de redução de 10.

6.3.7 Análise dos erros

Estando-se interessado somente nas análises e estudos dos erros apresentados pelo sistema, pode-se acessar o submenu *Estudos dos erros*. A Figura 6.20, ilustra a apresentação do submenu de análise dos erros do robô em conjunto com um dos gráficos gerados por este sistema (gráfico dos erros das velocidades do robô). Esta análise de erros é particularmente interessante quando se está procurando otimizar o sistema obtendo-se um *feedback* pormenorizado dos erros apresentados em todos os pontos cinemáticos e dinâmicos do sistema



(a) Velocidade angular na roda direita e seu erro. (b) Velocidade angular na roda esquerda e seu erro.

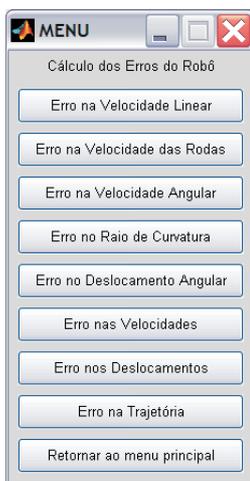
Figura 6.19: Velocidade angular nas rodas direita e esquerda com os respectivos erros dinâmicos.

robô móvel.

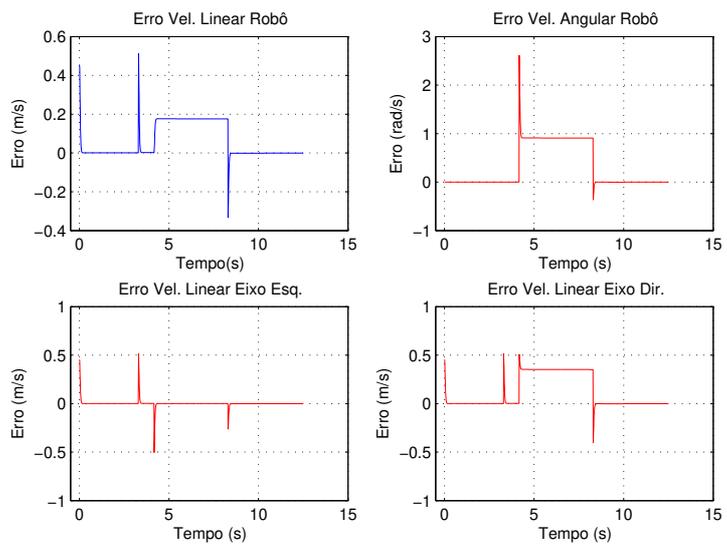
A Figura 6.21, exemplifica o uso do submenu do módulo de estudos de erros para análise dos erros nos eixos do robô móvel. É apresentado um dos gráficos gerado pelo sistema onde tem-se os erros da velocidade angular (ω) dos eixos esquerdo e direito do robô móvel.

6.3.8 Conclusões

Neste estudo de caso comprovou-se, através das análises gráficas apresentadas, que a atuação dinâmica do sistema responde perfeitamente aos parâmetros do modelo cinemático e dinâmico do robô móvel em estudo. Mostrou-se a eficácia do controlador adotado para o sistema robótico móvel proposto na tentativa de amenizar os erros dinâmicos do sistema. Comprovou-se também, a utilidade do módulo analisador gráfico na análise dos parâmetros cinemáticos e dinâmicos adotados no sistema. Com uma velocidade linear alternada aplicada ao sistema robô móvel, ultrapassando propositalmente, para efeito de estudos do sistema, a velocidade máxima permitida pelo modelo, todos os resultados foram satisfatórios, uma vez que a simulação traduz a realidade do sistema. Assim sendo, o sistema simulador demonstrou sua aplicabilidade no sentido de prover todos as variáveis cinemáticas e dinâmicas do modelo

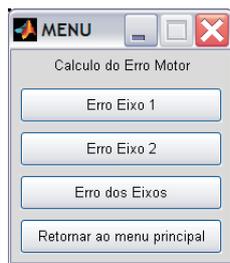


(a) Submenu de análise dos erros do robô.

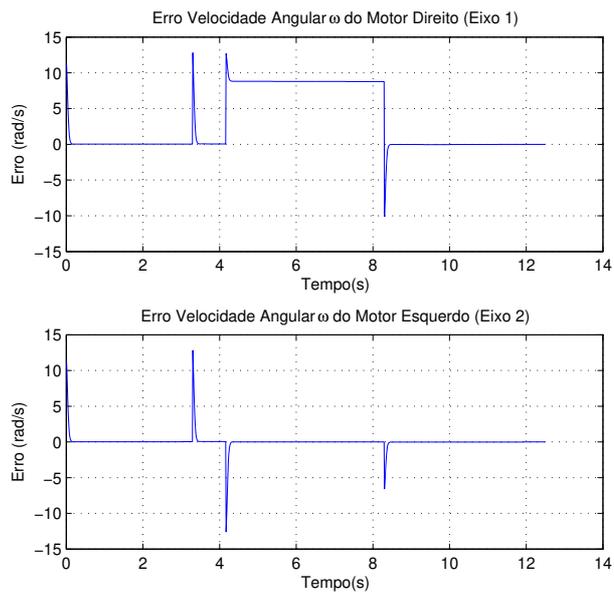


(b) Gráfico dos erros das velocidades do robô.

Figura 6.20: Submenu do módulo de estudos de erros do robô com um exemplo de gráfico gerado.



(a) Submenu de análise dos erros dos eixos.



(b) Gráfico dos erros nos eixos do robô.

Figura 6.21: Submenu do módulo de estudos de erros dos eixos com um exemplo de gráfico gerado.

em simulação em tempo real, onde são detectadas as falhas e refeitos os testes antes de ser implementado em *hardware* no robô móvel, obtendo-se e validando-se o sistema virtualmente para depois implementá-lo através das técnicas de prototipagem rápida, vistas na seção 5.6.

6.4 Validação experimental cinemática

Neste tópico são apresentados resultados experimentais através da implementação de um exemplo de trajetória cinemática de um robô móvel, obtido através do simulador apresentado nesse trabalho (capítulo 5). Esta validação de um exemplo de trajetória cinemática foi executada em robôs móveis do Laboratório de Robôs Móveis Autônomos da Faculdade de Ciências da Universidade Estadual Paulista “Júlio de Mesquita Filho” (UNESP-Bauru).

Embora esses protótipos tenham sido desenvolvidos para fins educacionais, eles foram utilizados nesse trabalho de pesquisa pela sua simplicidade, facilidade de utilização e arquitetura aberta. Nesta seção, dar-se-a ênfase somente aos aspectos cinemáticos e referentes ao simulador de movimentos, não sendo enfatizado nem a implementação para validação de um controlador PID e nem a análise dinâmica do sistema, tendo em vista as limitações de capacidade de processamento do microprocessador embarcado utilizado neste protótipo. Em trabalhos futuros de pesquisa, deverão ser estudados e validados toda a arquitetura de supervisão e controle de movimentos, proposta nesse projeto de pesquisa.

6.4.1 O ambiente “Aedromo”

O Aedromo (Ambiente Experimental Didático de Robôs Móveis) é um ambiente de pequenas dimensões constituído de dois robôs móveis autônomos, com sistema de acionamento e controle local independentes. A supervisão e coordenação de movimentos destes robôs são realizadas através de uma arquitetura em malha fechada baseada numa câmera satélite sobre o ambiente. As informações fornecidas por uma câmera de vídeo modelo *WebCam* são enviadas para um ou dois computadores para processamento. As informações obtidas deste processamento são usadas para gerar uma seqüência de instruções que são enviadas para os robôs. Os robôs recebem as instruções e realizam as ações obedecendo a tarefas predeterminadas. As instruções são resultados de estratégias desenvolvidas pelo(s) programador(es) para efetuar as tarefas e realizar a navegação do robô pelo ambiente.

Este ambiente (Figura 6.22) apresenta inúmeras opções para aplicações didáticas, como para pesquisa e entretenimento. As aplicações implementadas podem ser, por exemplo, do

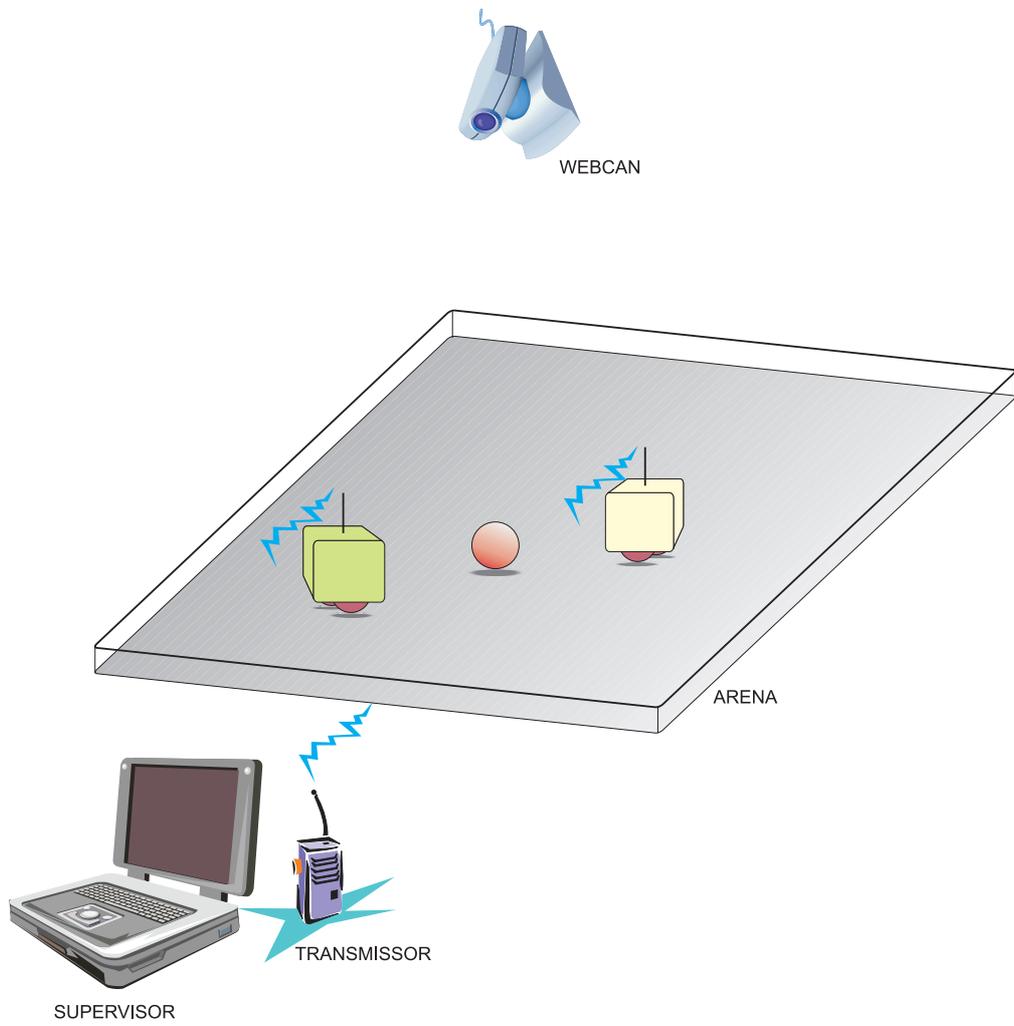


Figura 6.22: O ambiente Aedromo.

tipo jogo entre os robôs, similar ao tênis. Outra aplicação é um jogo parecido com o “pega-pega”. Uma tarefa de deslocar objetos de pesos diferentes de uma área para outra, onde a cooperação pode ser a motivação da pesquisa, também pode ser implementada. Tem-se também circuitos onde os robôs devem caminhar e com isso, criar situações de velocidade, aceleração, etc., permitindo a implementação de experimentos didáticos para a verificação de conceitos físicos e tratamento matemático. Aplicações práticas em áreas relacionadas diretamente à computação e a engenharia, como o processamento de imagens, inteligência artificial, sistemas de controle, planejamento de trajetória, navegação autônoma, enfim, toda sistemática envolvendo sistemas robóticos móveis, também podem ser desenvolvidas nestes sistemas.

A arquitetura de controle é caracterizada através de uma malha de controle fechada (**Sentir** \Rightarrow **Planejar** \Rightarrow **Atuar**). As imagens capturadas pela câmera satélite são enviadas por um cabo ao computador. No computador as imagens são adquiridas e processadas (**Sentir**). O processamento determina as posições individuais dos objetos (robôs, bolas, caixas, marcações, etc.) inseridos na arena. Estas informações são utilizadas pelo módulo de estratégia, de acordo com as regras da tarefa em execução (**Planejar**). Então, comandos são enviados aos robôs para agirem no ambiente segundo estas regras (**Atuar**). Os movimentos ocorridos devido às ações são captados pela câmera e o processo se repete.

6.4.2 Elementos constituintes do Aedromo

O sistema Aedromo é constituído por um sistema de capturação de imagens, através de uma câmara de vídeo, pelos robôs móveis, por uma arena onde se locomovem os robôs móveis, por um sistema de transmissão de dados via RF e por um ou mais computadores supervisórios.

6.4.2.1 Sistema de visão

Considerando o custo, o sistema de visão inicialmente proposto foi baseado na utilização de uma câmera do tipo *WebCam*. Estas câmeras, facilmente encontradas no mercado, possuem um campo de visão que permite capturar uma área aproximada de 60x80 cm² quando distanciada de aproximadamente 1 m do alvo. Estas medidas são interessantes para as dimensões operadas pela arena e do ambiente.

As *WebCams* são normalmente encontradas no padrão USB (*Universal Serial Bus*). Isto significa que não há a necessidade de placas de aquisição. A taxa de aquisição de imagens é

de 15 fps¹. A velocidade máxima para estes robôs é de 60 cm/s, isto significa que os robôs navegam cegos, em velocidade máxima, distâncias de, no máximo 4 cm. Portanto, o erro máximo de deslocamento provocado pela visão é de 4 cm.

6.4.2.2 Robôs móveis

Os dois robôs móveis inseridos no ambiente seguem fundamentalmente o mesmo projeto dos robôs que são utilizados em times de futebol robóticos. Têm dimensões de 7,5 cm de altura, por 7,5 cm de largura e 7,5 cm de profundidade, padronizados tanto pela FIRA (*Federation of International Robot-soccer Association*) quanto pelo Robocup². O projeto destes robôs móveis utiliza componentes do mercado nacional, permitindo a facilidade de manuseio e, pelo pouco peso dos robôs, uma menor potência de baterias com, conseqüentemente, baixo custo de implementação.

O sistema de tração usado nos robôs móveis é do tipo *diferencial drive*, isto é, um motor para cada roda de tração. O fato do eixo das rodas estar situado na metade da base com controle independente dos motores, permite uma grande mobilidade dos mesmos no ambiente. A mobilidade é uma característica fundamental para se aumentar as opções de uso e aplicações no ambiente.

Estes motores, de baixo custo têm velocidade de rotação de 3000 RPM com potência aproximada de 2 W, cada um, quando submetidos a uma tensão nominal de 5 VDC. Como as rodas possuem um diâmetro de 40 mm e a velocidade máxima estipulada em projeto, para o ambiente, de 0,6 m/s foi necessária a implementação de redutor de engrenagens na razão aproximada de 10:1. O arranque destes robôs, mesmo com o aumento de torque provocado pela redução, é modesto comparado aos utilizados nas competições citadas. Entretanto, não compromete as aplicações almejadas. Os robôs móveis recebem informações e agem no ambiente controlado através de um circuito embarcado, conforme é apresentado na Figura 6.23.

Os dados recebidos pelos módulos receptores de RF ou de IR³(infravermelho) são tratados pelo microprocessador. O tratamento dos dados se resume ao protocolo de comunicação. A informação resultante é de ações que serão realizadas pelos motores. Os motores são

¹Quadros por segundo, do inglês: *frames per second*.

²O Robocup é uma competição a nível mundial que se desenrola todos os anos. Visa o estudo e desenvolvimento da Inteligência Artificial (IA) e da Robótica, fornecendo desafios e problemas onde várias tecnologias e metodologias se podem combinar para obter os melhores resultados.

³Do inglês: *Infra-Red*.

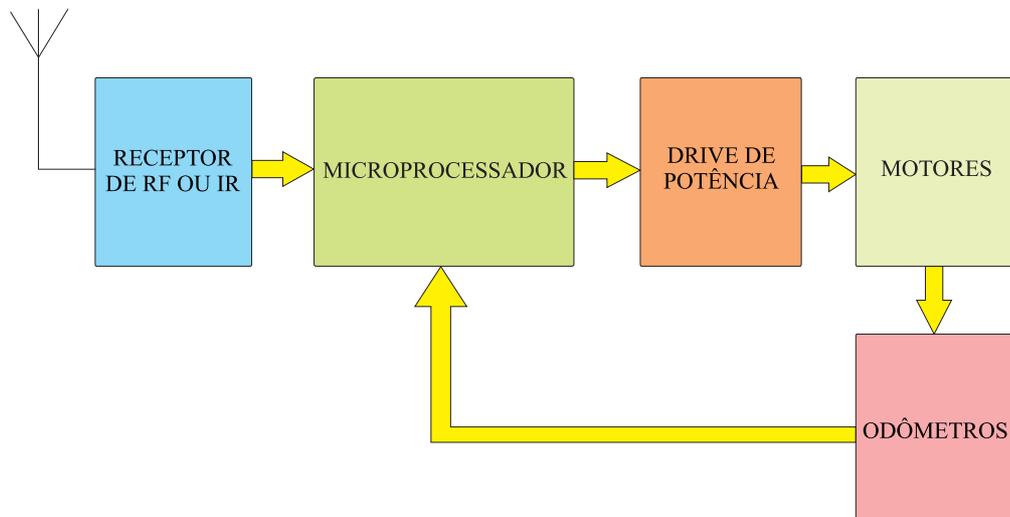


Figura 6.23: Diagrama em blocos da eletrônica embarcada dos robôs.

controlados por uma ponte de chaveamento eletrônico (*H-Bridge*) que determina o sentido de rotação e a velocidade dos motores. A técnica utilizada para controle da velocidade é a modulação por largura de pulso, PWM (*Pulse Width Modulation*). A rotação das rodas é monitorada por um sistema de odometria que corrige os erros sistemáticos mecânicos do robô móvel.

6.4.2.3 A arena

A arena é uma área de 60 x 80 cm plana. A superfície é preta e a murada lateral, de 3 cm de altura, é branca. Estas dimensões foram especificadas devido ao enquadramento obtido por uma câmera *WebCam* fixada a uma altura, entre 80 cm e 1m, do centro da arena. Estas dimensões são suficientes para permitir inúmeras situações aos dois robôs que ocupam esta área. Além disso, seu custo é menor e a facilidade de transporte maior.

Obviamente, a superfície preta foi escolhida para facilitar o sistema de visão. A murada lateral em branco foi escolhida para melhorar a reflexão de raios IR quando utilizado como alternativa de menor custo para o sistema de comunicação entre o computador e os robôs.

6.4.2.4 O sistema de comunicação

O sistema de comunicação utilizado é constituído por um módulo de Rádio frequência (RF) que utiliza transmissor integrado, por um módulo de infravermelho (IR) e por um módulo controlador, onde um sistema microprocessado, utilizando-se microcontroladores da

família PIC⁴, é usado para compatibilizar os sinais de saída serial do computador e o sistema utilizado. Além disso, este circuito pode receber e concentrar o sinal de dois computadores diferentes para um mesmo módulo de transmissão, conforme mostra a Figura 6.24. Isto ocorre quando a aplicação implica em diferentes estratégias de controle de movimentos em cada robô.

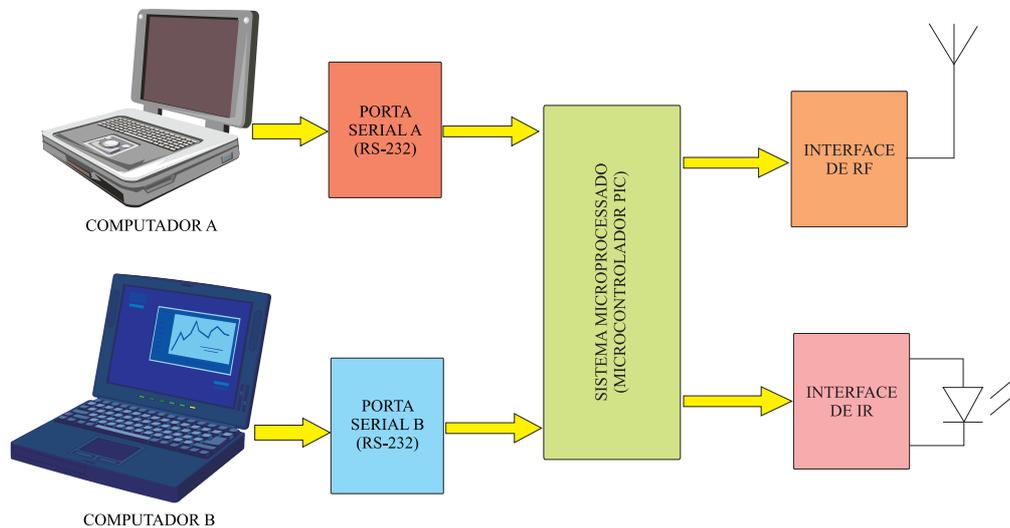


Figura 6.24: Diagrama de blocos do sistema de comunicação e transmissão de sinal.

Um sistema microprocessado recebe os sinais seriais (padrão RS-232), de um ou dois computadores, e converte para o formato exigido pelo módulo transmissor RF utilizado. O módulo transmissor de RF e os receptores em cada robô são de mesma frequência. O sinal convertido pelo microprocessador é endereçado individualmente para cada um deles. Assim, mesmo um robô recebendo todo o sinal obtido, ele identificará e utilizará somente os que forem dele.

6.4.3 Descrição material dos elementos do Aedromo

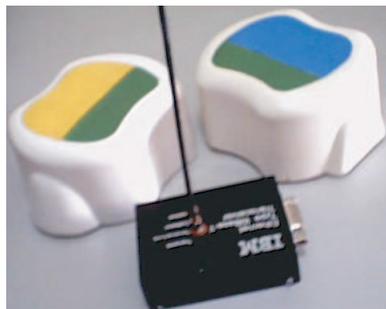
O ambiente de atuação, a arena (Figura 6.25) e dispositivos robóticos móveis utilizados constituem o Aedromo. Os robôs móveis possuem duas rodas principais com tração e motorização independente em cada uma das rodas, conforme ilustrado na Figura 6.26(b).

Como pode ser observado na Figura 6.26(a), estes robôs possuem, na parte superior, uma combinação de duas cores primárias distintas. Estas servem para dois propósitos básicos: primeiro, localizar o robô no plano cartesiano (x, y) e, segundo, obter a direção (θ) do robô

⁴Fabricante: Microchip



Figura 6.25: Arena - Plano cartesiano utilizado para navegação do robô.



(a) Robô móvel.



(b) Visão do sistema motriz.

Figura 6.26: Sistemas robóticos do Aedromo.

com relação ao plano cartesiano (x, y) . A posição e a direção do robô é obtida através de uma câmara satélite que capta as imagens instantâneas da área do campo em que os robôs atuam.

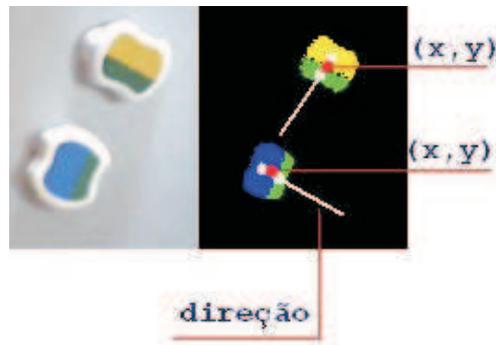
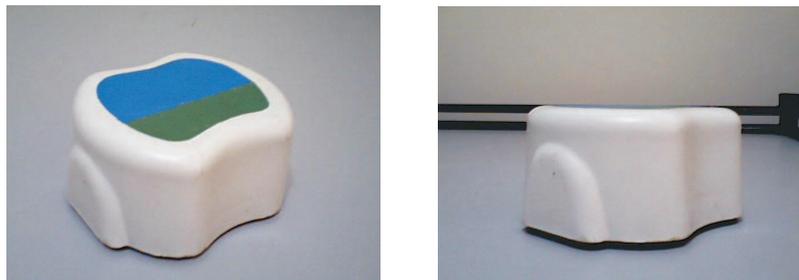


Figura 6.27: A posição e a direção dos robôs no plano cartesiano.

A Figura 6.27 apresenta a captura aérea da câmara satélite e o processamento da posição e da direção de cada um dos robôs do exemplo, e na Figura 6.28 são apresentados maiores detalhes da estrutura dos robôs utilizados.



(a) Visão em perspectiva.

(b) Visão lateral.

Figura 6.28: Detalhes do robô utilizado.

6.4.4 Implementação do supervisor de trajetórias

A trajetória a ser executada pelo robô a partir do software de geração de trajetórias, apresentado na seção 5.2, é ilustrada na Figura 6.29. Para efeitos de validação experimental, foi utilizado um mesmo exemplo de trajetória apresentada na validação do simulador, permitindo assim a obtenção de parâmetros de comparação entre o sistema real com implementação cinemática deste exemplo e o sistema virtual com tração diferencial com implementação de

controlador cinemático e dinâmico, conforme visto no capítulo 5.

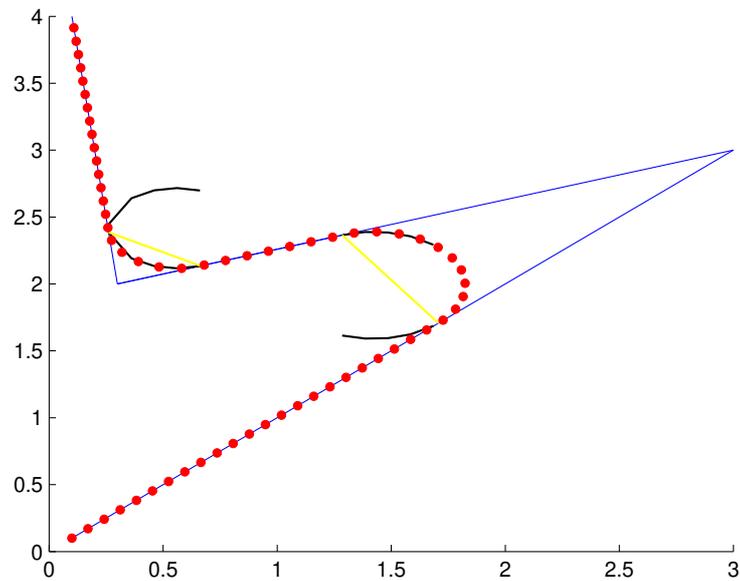


Figura 6.29: Trajetória a ser executada pelo robô.

O resultado experimental obtido através da execução desta trajetória num dos Robôs moveis constituintes do aedromo pode ser visualizado na representação gráfica mostrada na Figura 6.30. Pode-se observar que a trajetória foi executada, a primeira vista, a contento, mesmo não sendo considerado aspectos relacionados a dinâmica e controle do robô móvel utilizado, e estando fora da escala originalmente proposta. Isto foi necessário para se adaptar a trajetória às dimensões físicas do ambiente de navegação.

O gráfico apresentado na Figura 6.31, ilustra a diferença da trajetória de referência, mostrada pela linha fina com marcações em quadrados e a trajetória cinemática executada pelo robô (mostrada por linha grossa). Embora o robô consiga executar a trajetória proposta desejada, examinando a Figura 6.31, pode-se observar a existência de erros de posicionamento de trajetória significativos, principalmente nas curvas em relação a execução de retas. Isto pode ser justificado, considerando-se que o sistema de controle utilizado no sistema robótico em estudo é um sistema proporcional simples. Com a implementação de um controlador que considere aspectos dinâmicos, conforme descrito no capítulo 5, este erro de trajetória poderá ser minimizado, sendo até desprezível, se comparado ao exemplo em questão. Comprovam-se, desta maneira, a necessidade e eficácia do sistema virtual proposto neste trabalho.

A título de apresentar-se outro exemplo de execução de trajetória cinemática num dos

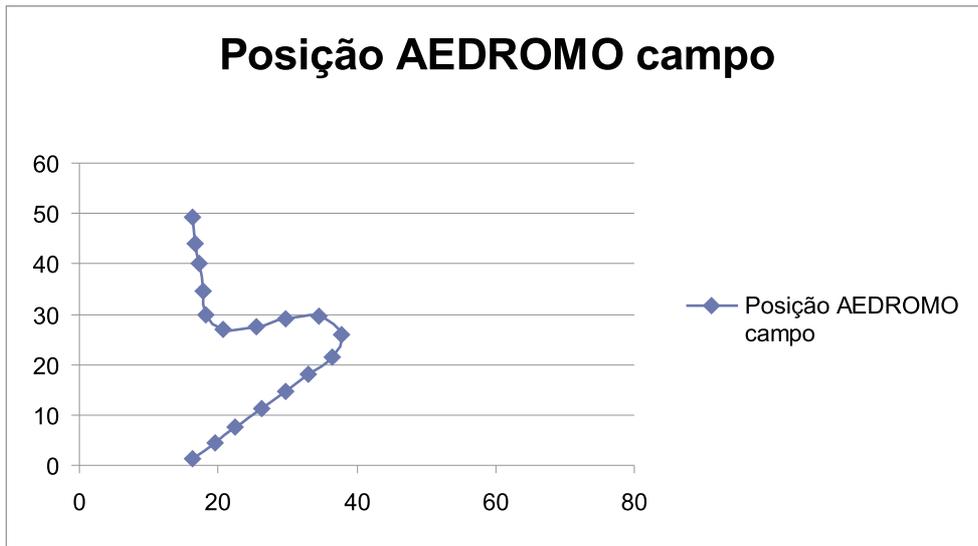


Figura 6.30: Execução da trajetória proposta pelo robô móvel.

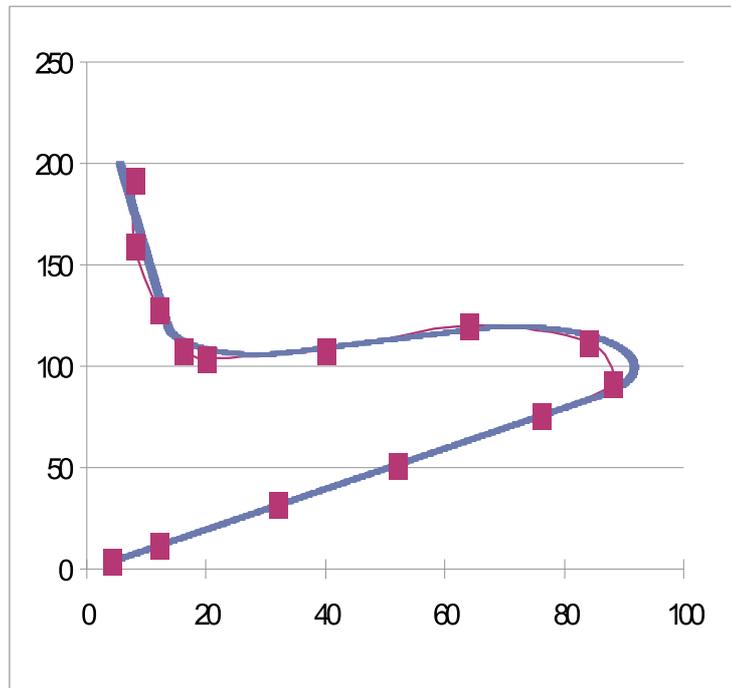


Figura 6.31: Trajetórias de referência e executada pelo robô.

robôs constituintes do Aedromo, tem-se ilustrado na Figura 6.32 uma trajetória de referência em formato de retângulo, com sua trajetória executada (traçado irregular). Percebe-se que a trajetória real pode apresentar erros significativos, principalmente nas curvas, demonstrando a ineficiência e baixa qualidade do sistema de controle implementado atualmente, e necessidade de implementação de um sistema de controle que considere a dinâmica do robô móvel em estudo.

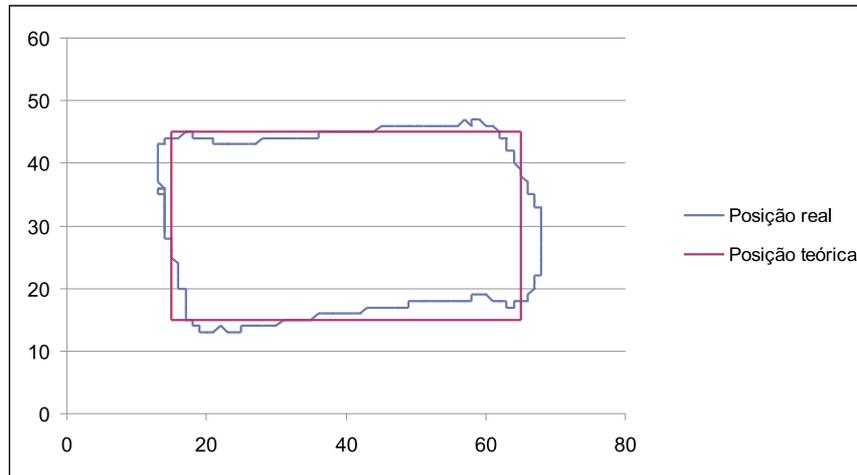


Figura 6.32: Execução de uma trajetória retangular pelo robô móvel em estudo.

6.4.5 Conclusões

Fica patente então, com as ilustrações dos exemplos apresentados, que, para que um sistema robótico móvel apresente dirigibilidade, mobilidade e confiabilidade nas execuções de trajetórias dinâmicas, ele deve ser implementado com um controlador robusto, conforme sugerido neste trabalho nos capítulos 4 e 5, para que possa, uma vez carregado em sua memória embarcada a trajetória cartesiana a ser executada, executa-la a contento, com os mínimos erros possíveis e com a otimização dos parâmetros necessários.

6.5 Considerações finais

Conforme descrito neste capítulo, nos estudos de casos (seções 6.2 e 6.3), fica patente então a importância do sistema simulador no desenvolvimento de protótipos de sistemas robóticos móveis. Isto porque, o modelo do robô móvel pode ser testado em todas as diferentes configurações e composições possíveis, validando-se a cinemática e a dinâmica do sistema,

antes de sua implementação física final, que pode ser feita através de técnicas de prototipagem rápida (seção 5.6). Desta maneira, para que o robô móvel tenha o desempenho desejado, pode-se ajustar os parâmetros do controlador, as variáveis dos atuadores, os parâmetros do modelo cinemático, e todas as demais condições, até se conseguir um modelo ótimo, ou pelo menos, dentro das especificações necessárias. Isto é possível, graças ao módulo analisador gráfico que, através de gráficos gerados a partir da simulação dinâmica da trajetória do robô móvel, fica muito mais fácil a obtenção de parâmetros comparativos para uma otimização ou correção de erros.

Para o estudo do comportamento dinâmico dos robôs móveis do ambiente “Aedromo” apresentado na seção 6.4.1, torna-se necessário alterações no sistema de acionamento e odometria destes robôs, assim como o conhecimento de todos os parâmetros do modelo. Fica como sugestão para trabalhos futuros este levantamento de dados.

Capítulo 7

Conclusões e Sugestões para Próximos Trabalhos

O desenvolvimento de sistemas de controle para robôs móveis autônomos tem se mostrado um grande desafio para os pesquisadores até os dias atuais. Diferentes abordagens para o projeto de sistema de controle para robôs móveis vêm sendo utilizadas em diversas áreas de pesquisa. Existem inúmeros sistemas de controle para serem utilizados nos sistemas robóticos móveis, mas geralmente estes são limitados e não apresentam uma arquitetura aberta o suficiente que permitam adaptações, melhorias, otimizações, ampliações, etc.

A eletrônica embarcada apresenta restrições quanto ao consumo de energia e quanto ao volume físico ocupado. Considerando ainda o grande número de tarefas e operações que dispositivos embarcados móveis podem realizar, bem como a variedade de combinações de sensores e atuadores usados na realização destas tarefas, é necessário prever em sua eletrônica, aspectos de flexibilidade e facilidade de alterações no projeto.

Desta monta, é apresentada neste trabalho, uma proposta de arquitetura aberta de supervisão e controle embarcada a ser utilizada em sistemas robóticos móveis, onde é descrito o projeto estruturado em pequenos módulos independentes, com interfaces de comunicação, inseridos dentro de uma arquitetura aberta orientada à estrutura. A aplicação final visa o desenvolvimento de um sistema de controle embarcado para dispositivos robóticos móveis que terá por finalidade a navegação, dentro de um ambiente interno ou externo, possibilitando a aplicação do conceito de prototipagem rápida.

O planejamento da locomoção, sob vários tipos de restrições, é um campo muito vasto de pesquisas na área da robótica móvel. O planejamento básico de trajetória para os robôs móveis refere-se à determinação de um caminho entre uma configuração inicial do robô e uma configuração final, de tal forma que o robô não colida com nenhum obstáculo no ambiente, e que a movimentação planejada seja consistente com as restrições cinemáticas do veículo.

Neste sentido, é apresentado neste trabalho, um planejador de trajetórias que tem como propósito a geração de uma melhor trajetória possível a ser executada pelo robô móvel, respeitando-se as restrições cinemáticas e físicas do robô assim como os obstáculos fixos ou móveis do ambiente.

Ferramentas que venham a permitir a rápida implementação prática de um projeto de *hardware* e de *software* não são apenas uma comodidade técnica, mas são também essenciais para a viabilidade econômica do projeto. Isto é válido para empresas, universidades e centros de pesquisa, cada vez mais sujeitos às competições do mercado. Espera-se, com este trabalho, contribuir para o esclarecimento dos potenciais destes sistemas, bem como, através de exemplos práticos, demonstrar sua aplicação na área de controle de sistemas embarcados. A utilização de uma arquitetura aberta no projeto do controlador permite descrever um projeto típico em sistemas modulares, onde diferentes blocos são projetados separadamente para posterior integração. O ambiente de simulação permite a visualização de possíveis erros de projeto, antes mesmo da execução física do mesmo. O ambiente de desenvolvimento contribui ainda para o baixo custo final do projeto, sua facilidade de implementação e de correção de erros. Outra vantagem do ambiente de desenvolvimento é a característica de hierarquização, a qual facilita o trabalho em equipe.

Uma das principais motivações deste trabalho é propiciar um ambiente virtual que facilite o desenvolvimento de protótipos de sistemas embarcados, enfatizando a implementação de ferramentas que permitam a simulação das condições cinemáticas, dinâmicas e de controle, com monitoração em tempo real de todos pontos importantes do sistema. Desta monta, a proposta de um simulador virtual de sistemas robóticos móveis é apresentada juntamente com técnicas de prototipagem rápida. Foi dado ênfase no projeto do simulador virtual cinemático e dinâmico de robôs móveis, onde são testadas a eficácia do projeto do controlador.

Neste contexto, a prototipagem rápida é a tecnologia que permite, em conjunto com um ambiente virtual de simulação, o desenvolvimento de um projeto de um controlador para robôs móveis. Depois de testado e validado no simulador, o sistema de controle é programado na memória da placa de controle do robô móvel. Desta feita, uma economia de tempo e material é conseguida, validando primeiramente todo o modelo virtualmente para depois se operar a implementação física do sistema.

O ambiente proposto pode também ser utilizado como uma ferramenta de apóio didático em laboratórios de graduação e pós-graduação de engenharia. O simulador de trajetória cinemática e dinâmica de sistemas robóticos móveis também é uma contribuição para o meio tecnológico de estudos robóticos. Finalmente, pode-se citar como contribuição ao meio

científico, a geração de artigos publicados em revistas e eventos científicos da área, de âmbito nacional e internacional.

São aspectos promissores do ambiente proposto:

- Flexibilidade - há uma grande variedade de configurações possíveis na implementação de soluções para diversos problemas associados com desenvolvimento de protótipos de sistemas embarcados.
- Adaptação a tarefa - apenas os recursos necessários a execução de uma tarefa são utilizados. Sistemas mais simples são tratados com versões mais simples do ambiente proposto, evitando-se perda de recursos.
- Ambiente aberto - permite que seus blocos componentes sejam totalmente acessados e eventualmente modificados para agregar novas soluções
- Facilidade de expansão - novos recursos podem ser facilmente adicionados a um projeto, permitindo a adição de novas funcionalidades.
- Possibilidade de controle supervisorio de vários protótipos simultaneamente.
- Custo relativamente baixo.

O ambiente proposto não está completo e é provável que nunca venha estar. O objetivo é que este ambiente evolua, seus componentes sendo aperfeiçoados com o tempo, permitindo que soluções mais simples, rápidas e baratas sejam usadas no desenvolvimento de protótipos de sistemas embarcados.

Um dos objetivos finais deste trabalho seria a implementação física do ambiente de arquitetura aberta proposta, no protótipo do robô móvel apresentado. Este sistema seria utilizado para validação final do controlador desenvolvido no simulador, através das técnicas de prototipagem rápida. Isto não foi possível devido a dificuldade na aquisição da placa do controlador embarcado com processador DSP, conforme descrito na seção 3.3.3, uma vez se tratar de material de importação. Fica então esta implementação como sugestão para trabalho futuro.

Como continuação deste trabalho são também relacionados os seguintes objetivos:

- Melhoria no sistema gerador de trajetórias, com maiores alternativas de resultados de trajetórias otimizadas.

- Implementação de controladores preditivos, ou de outras técnicas, para melhoria do desempenho do sistema.
- Desenvolvimento de novos blocos no simulador que permita a simulação de navegação autônoma dos robôs móveis.
- Implementação de um sistema de navegação autônoma por marcadores ativos, para robôs móveis, conforme descrito de maneira introdutória no apêndice D.
- Estudo e implementação de técnicas de fusão de sensores, com a implementação do filtro de Kalman no simulador, já modificado para navegação autônoma, para otimização da localização do robô móvel no plano cartesiano.
- Desenvolver interfaces para operação de sistemas embarcados cooperativos, como o caso de vários robôs móveis operando em conjunto para um determinado objetivo.

Referências

- ABEL, D.; BOLLIG, A. *Rapid Control Prototyping: Methoden und Anwendungen*. 1. ed. German: Springer, 2006.
- ABUT, H. *DSP FOR IN-VEHICLE AND MOBILE SYSTEMS*. [S.l.]: SPRINGER VERLAG NY, 2004.
- AGUINAGA, I.; BORRO, D.; MATEY, L. Path-planning techniques for the simulation of disassembly tasks. *Assembly Automation*, Emerald Group Publishing Limited, USA, v. 27, n. 3, p. 207–214, 2007.
- ALAMI, R. et al. An architecture for autonomy. *The International Journal of Robotics Research*, SAGE Publications, v. 17, n. 4, p. 315–337, 1998.
- ALTERA. *AHDL*. USA, 2007. Disponível em: <<http://www.altera.com>>. Acesso em: mar. 2007.
- AMBRÓSIO, J. A. C. *Advances in Computational Multibody Systems*. 1. ed. Netherlands: Springer, 2005.
- BAR-SHALOM, Y.; FORTMANN, T. E.; CABLE, P. G. Tracking and Data Association. *Acoustical Society of America Journal*, v. 87, p. 918–919, fev. 1990.
- BETKE, M.; GURVITS, L. Mobile robot localization using landmarks. *IEEE Trans. Robotics and Automation*, v. 13, n. 2, p. 251–263, 1997.
- BHASKER, J. A. *VHDL Primer*. 3. ed. New Jersey, USA: Prentice Hall PTR, 1999.
- BORENSTEIN, J. et al. Mobile robot positioning: Sensors and techniques. *Journal of Robotic Systems*, John Wiley & Sons, Inc, v. 14, n. 4, p. 231–249, dec 1998.
- BOUCHER, P.; DUMUR, D. Survey on predictive control. In: BALTHAZAR, P. G. J.; CLAYSSSEN, J. (Ed.). *Nonlinear Dynamics, Chaos, Control and their Applications to Engineering Sciences*. [S.l.: s.n.], 1998. v. 2, p. 1–40.
- BOZIC, S. *Digital and Kalman Filtering*. 2. ed. [S.l.]: Butterworth-Heinemann, 1994.
- BRIOT, M. Le système de perception du robot hilare. In: *2o. Congress AFCET/IRIA*. Toulouse, France: [s.n.], 1979.

- BURGARD, W. et al. Estimating the absolute position of a mobile robot using position probability grids. In: *The 1996 13th National Conference on Artificial Intelligence AAAI/IAAI*. Portland, OR, USA: [s.n.], 1996. v. 2, p. 896–901.
- CHANG, K. C. *Digital Design and Modeling with VHDL and Synthesis*. 1. ed. CA, USA: IEEE Computer Society Press, 1997.
- CHEN, R.-X.; CHEN, L.-G.; CHEN, L. System design consideration for digital wheelchair controller. In: *IEEE Transactions on Industrial Electronics*. [S.l.: s.n.], 2000. v. 47, n. 4, p. 898–907.
- CHEN, Y. D.; NI, J.; WU, S. M. Dynamic calibration and compensation of a 3d laser radar scanning system. In: *IEEE International Conference on Robotics and Automation*. Atlanta, GA: [s.n.], 1993. v. 3, p. 652–664.
- CHOSSET, H. et al. *Principles of Robot Motion : Theory, Algorithms, and Implementations*. [S.l.]: The MIT Press, 2005.
- CHOW, W. W. et al. The ring laser gyro. *Reviews of Modern Physics*, v. 1, n. 57, p. 61–104, 1985.
- CHUNG, Y.; PARK, C.; HARASHIMA, F. A position control differential drive wheeled mobile robot. *IEEE Transactions on Industrial Electronics*, v. 48, n. 4, p. 853–863, aug 2001.
- COMPTON, K. Reconfigurable computing: A survey of systems and software. *ACM Computing Surveys*, v. 34, n. 2, p. 171–210, 2002.
- CONRAD, D. J.; SAMPSON, R. E. 3d range imaging sensors. In: HENDERSON, T. (Ed.). *Traditional and Non-Traditional Robotic Sensors*. [S.l.]: Springer-Verlag, 1990. (NATO ASI, F63), p. 35–47.
- CORIC, S.; LEESER, M.; MILLER, E. Parallel-beam backprojection: An FPGA implementation optimized for medical imaging. In: *FPGA '02*. Monterey, California, USA: [s.n.], 2002. p. 217–226.
- COX, I. J. Blanche - an experiment in guidance and navigation of an autonomous robot vehicle. In: IEEE (Ed.). *IEEE Transactions on Robotics and Automation 7*. [S.l.: s.n.], 1991. p. 193–204.
- CRAIG, J. J. *Introduction to Robotics: Mechanics and Control*. 2. ed. Boston: Addison-Wesley, 1989.
- D'AMORE, R. *VHDL: descrição e síntese de circuitos digitais*. 1. ed. Rio de Janeiro, Brasil: LTC, 2005.
- D'AZZO, J. J.; HOUPIS, C. J.; SHELDON, S. N. *Linear controle system analysis and design with matlab*. 4. ed. New York, USA: Mc Graw-Hill, 2003.

- DENG, X.; MIRZAIAN, A. Competitive robot mapping with homogeneous markers. *IEEE Journal of Robotics and Automation*, v. 12, n. 4, p. 532–542, aug 1996.
- DIDO, J. et al. A flexible floating-point format for optimizing data-paths and operators in fpga based dsps. In: *FPGA'02*. Monterey, California, USA: [s.n.], 2002. p. 50–55.
- DIMITROV, D. et al. Rapid prototyping driven design and realisation of large components. *Rapid Prototyping Journal*, Emerald Group Publishing Limited, USA, v. 13, n. 2, p. 85–91, 2007.
- DINIZ, P. C.; PARK, J. Data reorganization engines for the next generation of system-on-a-chip FPGAs. In: *FPGA's 2002 Proceedings*. Monterey, California, USA: [s.n.], 2002. p. 237–244.
- DRIZO, A.; PEGNA, J. Environmental impacts of rapid prototyping: an overview of research to date. *Rapid Prototyping Journal*, Emerald Group Publishing Limited, USA, v. 12, n. 2, p. 64–71, 2006.
- DUBRAWSKI, A.; SIEMIATKOWSKA, B. A method for tracking pose of a mobile robot equipped with a scanning laser range finder. *Proc. IEEE Int. Conference Robotics and Automation*, Leuven, Belgium, p. 2519–2523, may 1998.
- DUDEK, G.; JENKIN, M. *Computational Principles of Mobile Robotics*. Cambridge CB2 2RU, UK: Press Syndicate of the University of Cambridge, 2000.
- DUDEK, G. et al. Reflections on modelling a sonar range sensor. McGill University, Montreal, Quebec, Canada, 1996.
- DUDEK, G. L. Environment representation using multiple abstraction levels. In: *Proceedings of the IEEE*. [S.l.]: IEEE, 1996. v. 8, n. 11, p. 1684–1704.
- ELLEPOLA, R.; KOVESI, P. Mobile robot navigation using recursive motion control. In: *Proceedings., Second EUROMICRO workshop on Advanced Mobile Robots*. [S.l.]: IEEE, 1997. p. 168–174.
- EVANS, W. R. Graphical analysis of control systems. In: *AIEE Trans.* [S.l.: s.n.], 1948. v. 69, n. 2, p. 547–551.
- EVERETT, H. R. *Sensors for Mobile Robots: Theory and Application*. [S.l.]: AK Peters, Ltd., 1995. ISBN 1568810482.
- FIERRO, R.; LEWIS, F. L. Control of a nonholonomic mobile robot using neural networks. *IEEE Transactions on Neural Networks*, v. 9, n. 4, jul 1998.
- FIGUEROA, J. F.; BARBIERI, E. Increased measurement range via frequency division in ultrasonic phase detection methods. *Acustica*, v. 73, p. 47–49, 1991.

- FOX, D. et al. Position estimation for mobile robots in dynamic environments. In: *Proc. of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*. Madison, WI: [s.n.], 1998. p. 983–988.
- GAO, W.; HINDERS, M. Mobile robot sonar backscatter algorithm for automatically distinguishing walls, fences, and hedges. *The International Journal of Robotics Research*, v. 25, n. 2, p. 135–145, 2006.
- GE, S. S.; LEWIS, F. L. *Autonomous Mobile Robots: Sensing, Control, Decision-making, and Applications*. USA: CRC Press, 2006.
- GELB, A. *Applied Optimal Estimation*. Cambridge, MA: The MIT Press, 1974.
- GIRALT, G.; CHANTILA, R.; VAISSET, M. An integrated navigation and motion control system for autonomous multisensory mobile robots. *Autonomous robot vehicles*, Springer-Verlag New York, Inc., New York, NY, USA, p. 420–443, 1990.
- GONZALEZ-BANOS, H.; LATOMBE, J.-C. Robot navigation for automatic model construction using safe regions. *Lecture Notes in Control and Information Sciences, Experimental Robotics VII*, Springer Berlin / Heidelberg, v. 271, p. 405–416, 2001.
- GORNI, A. A. Introdução à prototipagem rápida e seus processos. *Plástico Industrial*, v. 1, n. 23, p. 230–239, 2001.
- GRAF, B.; WANDOSELL, J. M. H.; SCHAEFFER, C. Flexible path planning for nonholonomic mobile robots. *Angenommen zur Eurobot'01.*, Stuttgart, Germany, 2001.
- HAMBLEN, J. O.; FURMAN, M. D.; HALL, T. S. *Rapid prototyping of digital systems - a tutorial approach*. 2. ed. Atlanta, Georgia: Georgia Institute of Technology, 2006.
- HAYES, M. H. *Schaum's Outline of Digital Signal Processing*. 1. ed. [S.l.]: McGraw-Hill, 1998.
- HOFFMANN-WELLENHOF, B. H. L.; COLLINS, J. *GPS: Theory and Practice*. 3. ed. New York: Springer-Verlag, 1994.
- ITO, S. A.; CARRO, L. A comparison of microcontrollers targeted to FPGA-based embedded applications. In: *IEEE 13th Symposium on Integrated Circuits and Systems Design Proceedings*. Manaus, Brasil: [s.n.], 2000. p. 397–402.
- JAZWINSKI, A. H. *Stochastic Processes and Filtering Theory*. San Diego, California: Academic Press, 1970.
- KALMAN, R. E. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, v. 82, p. 35–45, 1960.
- KALRA, P. Reconfigurable FPGAs help build upgradable systems. *Embedde Developers Journal*, vol.1, n. 1, p. 16–21, September 2001.

- KAMBHAMPATI, S.; DAVIS, L. Multiresolution path planning for mobile robots. *IEEE Journal of Robotics and Automation*, v. 2, n. 3, p. 135–145, sep 1986.
- KEAN, T. Cryptographic rights management of FPGA intellectual property cores. In: *FPGA's 2000 Proceedings*. [S.l.: s.n.], 2000. p. 113–118.
- KELLY, A. *Essential Kinematics for Autonomous Vehicles*. Pittsburgh, PA, May 1994.
- KERR, J. R. Real time imaging rangefinder for autonomous land vehicles. In: SPIE. *Mobile Robots III*. Cambridge, MA, 1988. v. 1007, p. 349–356.
- KIM, C.-J.; NELSON, C. R. *State-Space Models with Regime Switching: Classical and Gibbs-Sampling Approaches with Applications*. [S.l.]: The MIT Press, 1999.
- KLEEMAN, L. Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead-reckoning. *IEEE International Conference on Robotics and Automation*, v. 3, p. 2582–2587, may 1992.
- KLEEMAN, L.; KUC, R. Mobile robot sonar for target localization and classification. *The International Journal of Robotics Research*, v. 14, n. 4, p. 295–318, aug 1995. Disponível em: <<http://citeseer.ist.psu.edu/kleeman95mobile.html>>.
- KORTENKAMP, D. *AI-based Mobile Robots: Case Studies of Successful Robot Systems*. 1. ed. Cambridge, MA: MIT Press, 1998.
- KUC, R.; SIEGEL, M. W. Physically based simulation model for acoustic sensor robot navigation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, v. 9, n. 6, p. 766–768, 1987.
- KUIPERS, B.; BYUN, Y.-T. *A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations*. [S.l.], 1, 1990. Disponível em: <citeseer.ist.psu.edu/kuipers91robot.html>.
- LACROIX, S.; DUDEK, G. On the identification of sonar features. *Proc. IEEE/RSJ IROS*, Grenoble, France, sep 1997.
- LANG, S. Y. T.; KORBA, L. W.; WONG, A. K. C. Characterizing and modelling a sonar ring. *SPIE Conference on Mobile Robots*, Philadelphia, PA (USA), p. 291–304, nov 1989.
- LATOMBE, J. C. *Robot motion planning*. Norwell, MA: Kluwer, 1991.
- LATOMBE, J. C.; BARRAQUAND, J. Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles. *Algorithmica*, Springer New York, vol.10, n. 2-4, p. 16–21, August 1993.
- LAVALLE, S. M. *Planning Algorithms*. Cambridge University Press, 2006. Disponível em: <<http://misl.cs.uiuc.edu/planning/>>.

- LEDIN, J. *Simulation Engineering: Build Better Embedded Systems Faster*. 1. ed. USA: CMP Books, 2001.
- LEONARD, J.; DURRANT-WHYTE, H.; COX, I. J. Dynamic map building for an autonomous mobile robot. In: *IEEE int. conf. on Robotics and Automation*. [S.l.: s.n.], 1990.
- LEONARD, J. J.; DURRANT-WHYTE, H. F. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, v. 7, n. 3, p. 376–382, jun 1991.
- LEONARD, J. J.; DURRANT-WHYTE, H. F. *Directed Sonar Sensing for Mobile Robot Navigation*. 1. ed. [S.l.]: Springer, 1992.
- LEVITT, T. S.; KUIPERS, B. Navigation and mapping in large scale space. *AI Magazine*, v. 9, n. 2, p. 25–43, 1988.
- LI, S. et al. Making 2D map of environments based upon routes scenes. *Autonomous Robots*, Springer Netherlands, v. 8, n. 2, p. 117–128, apr 2000. ISSN 0929-5593.
- LOZANO-PEREZ, T.; WESLEY, M. A. An algorithm for planning collision-free paths among polyhedral objects. In: *Communications ACM*. [S.l.: s.n.], 1979. v. 22, n. 10.
- LU, F.; MILIOS, E. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, v. 4, p. 333–349, 1997. Disponível em: <citeseer.ist.psu.edu/lu97globally.html>.
- LU, F.; MILIOS, E. Robot pose estimation in unknown environments by matching 2d range scans. *Journal of Intelligent and Robotic Systems*, Springer Netherlands, v. 18, n. 3, p. 249–275, mar 1997.
- LUMELSKY, V. J.; STEPANOV, A. A. Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, Springer New York, v. 2, n. 1, p. 403–430, mar 1987.
- MACHADO, R. L. et al. Um ambiente de prototipagem rápida para o desenvolvimento de algoritmos de navegação de robôs móveis baseada em visão. In: *Anais do XIV Congresso Brasileiro de Automática - CBA 2002*. Natal, RN, BRASIL: [s.n.], 2002.
- MADSEN, C. B.; ANDERSEN, C. S. Optimal landmark selection for triangulation of robot position. *Robotics and Autonomous Systems*, Elsevier Science, v. 23, n. 4, p. 277–292, jul 1998.
- MALIK, R.; POLKOWSKI, E. T. Robot location densities. *Proceedings of SPIE Mobile Robots V*, p. 280–290, mar 1991.
- MAYBECK, P. S. *Stochastic models, estimation, and control*. [S.l.]: Academic Press, Inc, 1979. (Mathematics in Science and Engineering, v. 1).

- MELO, L. F. de; ROSÁRIO, J. M. Hardware reconfigurable control with opened architecture for mobile robots. In: COBEM 2005. *18th International Congress of Mechanical Engineering*. Ouro Preto, MG, Brazil, 2005. v. 1, p. 1–8.
- MELO, L. F. de; ROSÁRIO, J. M. A proposal for a hybrid opened architecture with hardware reconfigurable control applied in mobile robots. In: ROBIO 2006. *IEEE International Conference on Robotic and Bionemetics*. Kunming, China: IEEE, 2006. v. 1, p. 1101–1106.
- MELO, L. F. de; ROSÁRIO, J. M. A hybrid opened architecture with hardware reconfigurable control applied in nonholonomic mobile robots. In: COBEM 2007. *19th International Congress of Mechanical Engineering*. Brasilia, DF, Brazil, 2007. v. 1, p. 100–108.
- MELO, L. F. de; ROSÁRIO, J. M.; LIMA, C. R. Uma utilização de sistemas de arquitetura aberta aplicada em sistemas robóticos. In: ISA SHOW 2003. *Congresso Internacional de Automação*. São Paulo, SP: Revista InTech, 2003. v. 1, n. 66, p. 42–48. ISSN 1518-6024.
- MELO, L. F. de; ROSÁRIO, J. M.; LIMA, C. R. Robô móvel com sistema de arquitetura aberta utilizado na realização de tarefas de inspeção das paredes internas e externas de dutos. In: ISA SHOW 2004. *Congresso Internacional de Automação*. São Paulo, SP: Revista InTech, 2004. v. 1, n. 67, p. 77–78. ISSN 1518-6024.
- MELO, L. F. de; ROSÁRIO, J. M.; LIMA, C. R. Uma aplicação da robótica móvel utilizando sistema de arquitetura aberta. In: DINCON'2004. *3o. Congresso Temático de Dinâmica e Controle da SBMAC*. UNESP, Ilha Souteira, SP: SBMAC-Sociedade Brasileira de Matemática Aplicada e Computacional, 2004. v. 1, p. 1369–1373.
- MELO, L. F. de et al. A reconfigurable control architecture for mobile robots. In: MUSME 2005. *IEEE 2nd. International Symposium on Mutibody and Mechatronics*. Uberlândia, MG, Brazil: IEEE, 2005. v. 1, p. 1–8.
- MIYAZAKI, T. Reconfigurable systems. In: *IEEE Proceedings of the Design Automation Conference, ASP-DAC '98*. [S.l.: s.n.], 1998. p. 447–452.
- MONTEMERLO, M.; THRUN, S.; WHITTAKER, W. Conditional particle filters for simultaneous mobile robot localization and people-tracking. *Proceedings of ICRA '02, IEEE International Conference on Robotics and Automation*, v. 1, p. 695– 701, 2002.
- MORAVEC, H. The stanford cart and the cmu rover. *Autonomous Robot Vehicles*, Springer-Verlag, 1990.
- NASHASHIBI, F.; DEVY, M. 3D incremental modeling and robot localization in structured environment using a laser range finder. *IEEE International Conference on Robotics and Automation*, Atlanta, USA, v. 2, n. 6, p. 20–27, may 1996.
- NAVABI, Z.; DAY, S. Tutorial on use of vhdl for description of digital systems. In: *Fourth Annual IEEE International ASIC Conference and Exhibit Proceedings*. [S.l.: s.n.], 1991. p. 1–12.

- NICHERSON, S. B. et al. Design of ARK, a sensor-based mobile robot for industrial environments. *In Proc. Intelligent Vehicles*, Tokyo, p. 252–257, 1993.
- NILSSON, N. J. A mobile automation: An application of artificial intelligence techniques. In: *IJCAI: Control, Planning and Architecture*. [S.l.: s.n.], 1969. v. 2, n. 1, p. 233–239.
- NISE, N. S. *Engenharia de sistemas de controle*. 3. ed. Rio de Janeiro: John Wiley and Sons, 2002.
- OGATA, K. *Engenharia de Controle Moderno*. 4. ed. Sao Paulo: Prentice Hall, 2003.
- OSHANA, R. *DSP Software Development Techniques for Embedded and Real-Time Systems*. [S.l.]: Newnes, 2005.
- O’SULLIVAN, F. A statistical perspective on ill-posed inverse problems. *Statistical Science*, v. 1, n. 4, p. 502–518, nov 1986.
- PAGE, I. Reconfigurable processor architectures. *Microprocessors and Microsystems 20 - Elsevier Science Ltd.*, p. 185–196, 1996.
- PAGNOTTELLI, S.; VALIGI, P. SARA: a flexible framework for rapid prototyping of mobile robotics applications. In: *14th Mediterranean Conference on Control and Automation, 2006*. USA: [s.n.], 2006. p. 1–5.
- PARK, I.-P.; KENDER, J. R. Topological direction-giving and visual navigation in large environments. *Artificial Intelligence*, Elsevier Science Publishers Ltd., Essex, UK, v. 78, n. 1-2, p. 355–395, 1995.
- PARKINSON, B. W.; SPILKER, J. J. *Global Positioning System: Theory and Practice*. Washington, DC: American Institute of Aeronautics and Astronautics, 1996.
- PIMENTEL, J. C.; LE-HUY, H. A vhdl-basead methodology to develop high performance servo drivers. In: IEEE (Ed.). *Industry Applications Conference. Conference Record of the 2000 IEEE*. [S.l.: s.n.], 2000. v. 3, p. 1505–1512.
- POGGIO, T.; TORRE, V.; KOCH, C. Computational vision regularization theory. *Nature*, p. 314–319, 1985.
- RADIOMETRIX. *Radiometrix Bim433-40*. UK, 2007. Disponível em: <<http://www.radiometrix.co.uk/>>. Acesso em: mar. 2007.
- RENNER, F. M. et al. Desing methodology of application specific integrated circuit for mechatronic systems. *Microprocessors and Microsystems*, Elsevier Science Ltd, p. 95–102, 2002.
- ROUMELIOTIS, S. I.; BEKEY, G. A. Distributed multirobot localization. In: IEEE (Ed.). *IEEE Transactions on Robotics and Automation*. [S.l.: s.n.], 2002. v. 18, n. 2, p. 781–795.

- SCHEUER, A.; FRAICHARD, T. Continuous-curvature path planning for car-like vehicles. *In IEEE/RSJ IROS*, Grenoble, France, v. 2, p. 997–1003, 1997.
- SCHWARZ, R.; FRIEDLAND, E. *Linear System Theory*. [S.l.]: McGraw-Hill, 1964.
- SCIAVICCO, L.; SICILIANO, B. *Modeling and Control of Robot Manipulators*. New York: McGraw-Hill, 1996.
- SHIM, H.-S.; KIM, J.-H.; KOH, K. Variable structure control of nonholonomic wheeled mobile robot. *IEEE International Conference on Robotics and Automation*, Nagoya, Japan, v. 2, p. 1694–1699, may 1995.
- SIEGWART, R.; NOURBAKHSI, I. R. *Introduction to Autonomous Mobile Robots (Intelligent Robotics and Autonomous Agents)*. [S.l.]: The MIT Press, 2004.
- SIM, R.; DUDEK, G. Learning visual landmarks for pose estimation. *IEEE International Conference on Robotics and Automation*, Detroit, MI, USA, v. 3, p. 1972–1978, 1999.
- SMITH, S. *Digital Signal Processing: A Practical Guide for Engineers and Scientists*. [S.l.]: Newnes, 2003.
- SORENSEN, H. W. Least-squares estimation: from gauss to kalman. *IEEE Spectrum*, v. 7, p. 63–68, 1970.
- SPIEGEL, M. R. *Estatística*. 3. ed. São Paulo, SP: Makron Books do Brasil Editora Ltda., 1999.
- SPROUSE, J. *The NOMAD 200: Merging mind and motion*. 2004. Disponível em: <<http://nomadic.sourceforge.net/>>.
- STANLEY, B.; MCKERROW, L. Measuring range and bearing with a binaural ultrasonic sensor. *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Grenoble, France, v. 2, p. 565–571, sep 1997.
- SUGIHARA, K. Some location properties for robot navigation using a single camera. *Computer Vision, Graphics and Image Processing*, v. 42, p. 112–129, 1988.
- SUTHERLAND, K. T.; THOMPSON, W. B. Inexact navigation. *In Proc. IEEE Inter. Conf. on Robotics and Automation*, v. 1, p. 1–7, 1993.
- TAN, J.; XI, N. Integrated sensing and control of mobile manipulators. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, v. 2, n. 2, p. 865–870, 2001.
- TARDOS, J. D. et al. Robust mapping and localization in indoor environments using sonar. *The International Journal of Robotics Research*, v. 4, n. 21, p. 311–330, 2002.
- THIELSCHER, M. *Reasoning Robots. The art and science of programming robotic agents*. Netherlands: Springer, 2005. (Applied logic series, v. 33).

- THRUN, N. Finding landmarks for mobile robot navigation. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Leuven, Belgium: IEEE, 1998. p. 958–963.
- THRUN, S. A probabilistic on-line mapping algorithm for teams of mobile robots. *The International Journal of Robotics Research*, v. 20, n. 5, p. 335–363, may 2001.
- THRUN, S.; BURGARD, W.; FOX, D. A probabilistic approach to concurrent mapping and localization for mobile robots. *Autonomous Robots*, Springer Netherlands, v. 5, n. 3-4, p. 253–271, jul 1998.
- THRUN, S. et al. Integrating topological and metric maps for mobile robot navigation: a statistical approach. In: *AAAI '98/IAAI '98: Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1998. p. 989–995.
- TI. *DSP*. USA, 2007. Disponível em: <<http://www.dsp.ti.com>>. Acesso em: mar. 2007.
- TIKHONOV, A. N.; ARSEININ, V. Y. *Solutions of Ill-Posed Problems*. Washington, DC: Wiston, 1977.
- UNITED STATES COAST GUARD NAVIGATION CENTER. *Global Positioning System: Standard Positioning Service Specification*. 2. ed. [S.l.], jun 1995. Disponível em: <<http://www.navcen.uscg.mil/>>.
- VUYLSTEKE, P.; PRICE, C. B.; OOSTERLINCK, A. Image sensors for real-time 3d acquisition, part 1. In: _____. [S.l.]: Springer-Verlag, 1990. (NATO ASI, F63), cap. Traditional and Non-Traditional Robotic Sensors, p. 187–210.
- WALTER, W. G. An imitation of life. *Scientific American*, v. 182, n. 5, p. 42–45, 1950.
- WANG, J.; WILSON, W. J. 3D relative position and orientation estimation using Kalman filter for robot control. *Proceedings of IEEE International Conference on Robotics and Automation*, Nice, France, v. 3, p. 2638–2645, may 1992.
- WILKES, D. et al. Modelling sonar range sensors. *Advances in Machine Vision: Strategies and Applications*, World Scientific Press, Singapore, p. 361–370, 1992.
- WISSPEINTENER, T. The VolksBot concept - rapid prototyping for real-life applications in mobile robotics. *it-Information Technology*, Oldenbourg Verlag, German, v. 47, n. 5, p. 274–281, 2005.
- WOHLERS, T. T. *Rapid Prototyping and Tooling State of the Industry*. Colorado, USA, 2001.
- WOODBURY, N.; BRUBACHER, M.; WOODBURY, J. R. Noninvasive tank gauging with frequency-modulated laser ranging. *Sensors*, p. 27–31, sept 1993.
- WRIGHT, P. K. *21st Century manufacturing*. New Jersey: Prentice-Hall Inc, 2001.

YAMAUCHI, B. Mobile robot localization in dynamic environments using dead-reckoning and evidence grids. *IEEE International Conference on Robotics and Automation*, Minneapolis, MN, USA, v. 2, p. 1401–1406, apr 1996.

YAMAUCHI, B.; SCHULTZ, A.; ADAMS, W. Mobile robot exploration and map-building with continuous localization. *IEEE International Conference on Robotics and Automation*, Leuven, Belgium, v. 4, p. 3715–3720, may 1998.

ZHAO, Y.; BEMENT, S. L. Kinematics, dynamics and control of wheeled mobile robots. *IEEE International Conference on Robotics and Automation*, v. 1, p. 91–96, may 1992.

ZIEGLER, J. G.; NICHOLS, N. B. Optimum settings for automatic controllers. In: *ASME Trans.* [S.l.: s.n.], 1942. v. 64, p. 759–768.

ZILBERSTEIN, S. Resource-bounded sensing and planning in autonomous systems. *Autonomous Robots*, v. 3, n. 4, p. 31–48, 1996.

Apêndice A – Sensores nos Sistemas Robóticos Móveis

O sensoriamento é uma condição essencial para o controle dos sistemas robóticos móveis, até os mais simples. Os sensores mais fundamentalmente associados com a mobilidade são aqueles que medem a distância que as rodas se movimentaram ao longo do solo, sensores que medem mudanças inerciais e aqueles que medem as estruturas externas do ambiente. Para um robô móvel saber onde se encontra ou onde deve ir, ou ser capaz de “raciocinar” sobre os locais que tem estado, sensores e algoritmos de sensoriamento são necessários (LAVALLE, 2006).

Sensores e algoritmos para interpretação dos parâmetros podem ser muito complexos. Embora existam uma variedade muito grande de sensores, duas classes diferentes classes de sensores tem sido mais utilizadas em robôs móveis (DUDEK; JENKIN, 2000):

- sensores visuais, que utilizam a reflexão da luz pelos objetos do ambiente em que se localiza o robô, traduzindo-se num modelamento da estrutura física ao redor do mesmo;
- sensores não visuais, que utilizam o áudio, a inércia ou outra modalidade física para “sentir” o ambiente.

Neste capítulo é apresentada uma introdução aos algoritmos de sensores não visuais que serão de grande utilidade para o projeto dos sistemas de controle para os robôs móveis. É também abordado tecnologias de sensoriamento.

A.1 Conceitos básicos sobre sensores

Talvez a classificação mais utilizada sobre os tipos de sensores é a de *sensores de estado interno* e *sensores de estado externo* (BORENSTEIN et al., 1998):

- Sensores de estado interno provem a realimentação dos parâmetros do sistema robótico móvel, como nível da bateria, posicionamento das rodas, velocidade angular das rodas, ângulo das juntas (para robôs móveis com pernas), etc. O termo geral para sensores internos em organismos biológicos é *sensores proprioceptivos*, cujo termo também é utilizado por alguns autores em robótica.
- Sensores de estado externos tratam da observação dos aspectos do mundo exterior ao robô: umidade, a cor dos objetos, localização dos objetos, etc. Sensores externos que utilizam o toque são conhecidos como sensores de contato, enquanto todas as outras modalidades podem ser classificadas como de sensores de não contato.

Um sensor específico pode ser *ativo* ou *passivo* (EVERETT, 1995).

- Sensores ativos são aqueles que realizam suas medidas emitindo certo tipo de energia no ambiente ou modificando de certa forma o ambiente.
- Sensores passivos são aqueles que recebem passivamente a energia do meio para realizar suas medições.

Por exemplo, a visão humana ou o olfato envolvem a utilização de sensores passivos. Por outro lado, o toque pode ser considerado um sistema de sensor ativo.

Sensores passivos são mais freqüentemente preferidos porque são não intrusivos e mais eficientes em termos de consumo de energia. Pelo fato de não emitirem energia, são especialmente utilizados quando o robô tem que estar inconspícuo, como é necessário em vários contextos militares.

Esta diferença entre sensores ativos e passivos pode ser ilustrada considerando-se duas abordagens alternativas para se fazer uma medição de uma certa distância por um robô móvel: utilizando-se sensores de visão estéreo passiva e utilizando-se um sistema baseado em raio laser. A visão estéreo passiva funciona observando o mesmo ponto (e.g., a posição de alguma marca ou característica) num objeto através de duas câmaras diferentes e, então, medindo a paralaxe (i.e., a mudança na posição observada). A condição essencial é que a marca observada com uma câmara tem que ser captada pela outra câmara para que ocorra a triangulação. Se o objeto tem padrões repetitivos, se possuir marcas insuficiente ou se as marcas forem especulares (um ponto brilhante) que se movem com relação a posição do observador, esta distância será difícil de ser medida. Por outro lado, se a marca observada for um ponto laser projetado pelo robô, a alta energia a uma particular freqüência do laser irá, tipicamente, tornar esta marca única e sem ambigüidade.

Algumas observações gerais, porem muitas vezes críticas, podem ser feitas com respeito aos sensores no contexto da robótica móvel (BORENSTEIN et al., 1998):

- sensores reais são ruidosos;
- sensores reais retornam uma descrição incompleta do ambiente;
- sensores reais não podem ser, normalmente, completamente modelados.

Entendendo-se a extensão que estas três generalizações significam pode-se avaliar as dificuldades encontradas quando algoritmos “ideais” são utilizados no mundo real dos robôs móveis. Entender as propriedades reais dos sensores e as complexidades a serem medidas nos ambientes de operação dos robôs móveis pode levar projetos promissores a serem encerrados antes do término.

O uso de sensores para tirar conclusões a respeito do ambiente é freqüentemente descrito como um problema de recuperação ou reconstrução do ambiente.

Uma variedade muito grande de alternativas de tecnologias de sensores está disponível comercialmente. Esta plethora de opções pode ser decomposta em quatro classes principais baseadas no tipo de dados retornados pelos sensores (DUDEK; JENKIN, 2000):

- **Sensores de alcance.** Retornam medidas das distâncias entre o sensor e objetos no ambiente.
- **Sensores de posição absoluta.** Retornam a posição do robô (ou elementos do vetor posição) em termos absolutos (i.e., latitude e longitude).
- **Sensores ambientais.** Retornam as propriedades do meio ambiente em que está operando o robô. Podem ser propriedade ambientais tais como temperatura, pressão, altitude, etc., ou propriedades pontuais, tais como a cor de um ponto em frente ao robô, etc.
- **Sensores inerciais.** Retornam propriedades diferenciais da posição do robô (e.g., aceleração).

Uma variedade de características adicionais podem ser usadas para se classificar diferentes tecnologias de sensores. Embora não seja do objetivo deste trabalho estar entrando em detalhes a esse respeito, alguns exemplos de importantes propriedades dos sensores são listados abaixo.

- **Velocidade de operação.** Esta propriedade dos sensores diz respeito a taxa em que as medidas retornam quando os mesmos estão em funcionamento contínuo ou o tempo de atraso até que uma medição seja concluída quando é solicitada intermitentemente.
- **Custo.** Um par de sensores emissor-detector infravermelho pode custar em torno de alguns reais enquanto que giroscópios de precisão podem custar algumas centenas de dólar.
- **Taxa de erro.** Um dos fatores relevantes na escolha do sensor para um determinado projeto de robótica móvel é taxa média de erro que pode ser traduzida com sendo o número de medições incorretas e/ou o número de medições perdidas por um montante total de medições feitas.
- **Robustez.** Diz respeito a tolerância do sensor aos desvios do ambiente das condições ideais de operação. Fatos relevantes podem ser distúrbios físicos, ruídos do meio tais como ruídos elétricos, magnéticos, etc.
- **Requisitos computacionais.** Uma simples chave de contato quase que não requer esforço computacional para interpretação de seu resultado, enquanto que alguns algoritmos de visão robótica requerem significativos recursos computacionais e muitas vezes processadores especiais (i.e., DSPs) para obtenção de resultados satisfatórios.
- **Potência, peso e tamanho.** Alguns sensores necessitam potência contínua para se manter ativo de modo que as leituras possam ser feitas, enquanto outros podem ser desligados e só consomem energia quando estão efetuando medidas.

A.2 Modelamento de sensores

Pode-se modelar a relação entre as propriedades físicas e de interesse no ambiente e a leitura r feita pelo sensor utilizando um modelo de sensor, como

$$r = f(e). \tag{A.1}$$

Em princípio, o modelo de sensor deve incluir um modelo para ruídos internos ao sensor assim como ruídos oriundos de outros sensores. Alguns modelos básicos de sensores para diferentes aplicações serão demonstrados neste capítulo.

Um modelo típico de sensor inclui não somente o componente estocástico devido ao ruído mas também, usualmente, tem um alcance com menor número de dimensões do que o alcance

ambiental que se está medindo. Isto significa que o modelo para o sensor reduz um ambiente multidimensional num ambiente com dimensões menores. O problema da reconstrução do ambiente a partir dos dados gerados pelo sensor pode ser descrito como um *problema inverso*: restabelecer o(s) argumento(s) de uma função dada a saída da função. Este problema é tradicionalmente difícil de ser resolvido sem informações adicionais ou suposições, uma vez que a redução das dimensões freqüentemente trata da ambigüidade em estabelecer quais das várias possibilidades de cenários ambientais (i.e., valores de domínios) originou uma leitura particular. Estes tipos de problemas são reportados como sendo de *mau-posicionamento* (*ill-posed*). Mais geralmente, problemas *ill-posed* são problemas com:

- uma solução que é indefinida,
- uma solução que não é unicamente definida, ou
- uma solução que não é estável.

Como um exemplo concreto de problema *ill-posed*, considerando-se a tarefa de diferenciação de um sinal $f_0(x)$ adulterado por uma pequena quantidade de ruído de alta freqüência:

$$f(x) = f_0(x) + k \text{sen}(\omega x). \quad (\text{A.2})$$

O sinal diferencial $f'(x)$ é dado por

$$f'(x) = f'_0(x) + k\omega \text{cos}(\omega x). \quad (\text{A.3})$$

Se $\omega \gg 0$ (o ruído tem alta freqüência), a amplitude da parcela da equação correspondente ao ruído torna-se muito grande, adulterando o resultado a ser medido $f'(x)$ significativamente, ou, muitas das vezes, inviabilizando-o. Tentar extrair o conteúdo de $f'_0(x)$ do sinal corrompido $f'(x)$ é um exemplo clássico de um problema *ill-posed*, pois a derivada do sinal $f(x)$ é instável na presença de ruídos de altas freqüências.

Sensores também podem ser instáveis com respeito às suas entradas. Um sensor instável é aquele cuja saída pode sofrer grandes variações com uma pequena variação na entrada. Para um sensor com uma saída definida por $f(e)$, a instabilidade pode ser definida como (CHOSET et al., 2005)

$$\lim_{\Delta e \rightarrow 0} \frac{f(e) - f(e + \Delta e)}{\Delta e} \rightarrow \infty \quad (\text{A.4})$$

onde e é a propriedade física a ser medida e Δe é uma pequena variação desta grandeza. Na prática, se o valor de $\frac{f(e) - f(e + \Delta e)}{\Delta e}$ é grande para pequenos valores de Δe , o sensor pode ser

considerado como sendo instável. Isto implica que o valor da medição retornado pelo sensor pode variar imprevisivelmente.

Uma opção para se superar a instabilidade e outros atributos de problemas *ill-posed* é a aplicação de técnicas para se restringir o sistema e utilizar uma abordagem tal como a dos mínimos quadrados para uma maior exatidão nas medições. Uma das dificuldades dos sistemas que utilizam a técnica dos mínimos quadrados é que a otimização não pode representar facilmente suavidades ou restrições do ambiente que estiverem implícitas nos dados. Hipóteses de sistemas ambientais podem ser utilizadas para a resolução destes problemas o que permite selecionar qual das várias soluções possíveis é preferível. Estas hipóteses são apresentadas em várias formas. A mais geral delas é conhecida como hipótese de regularização de Tikhonov (TIKHONOV; ARSENIN, 1977). Esta abordagem refere-se a formulação do problema inverso com um termo de custo de estabilização adicional que implicitamente seleciona algumas soluções sobre outras baseadas em seus custos preferidos (a função custo deve ser construída para ser uma seminorma (i.e., é não ambígua)). No caso de regularização, um problema variacional, o termo custo é muitas vezes conhecido como *estabilizador*. A chave para se resolver a ambiguidade inerente ao problema *ill-posed* é que a ambiguidade deve ser resolvida pela seleção da solução correta, não apenas qualquer solução (O'SULLIVAN, 1986). Sob a abordagem de Tikhonov, a solução \mathbf{x} para um conjunto de restrições $\mathbf{M}\mathbf{x} = \mathbf{y}$ é dado pela minimização da quantidade apresentada a seguir, muitas vezes referida como uma energia, em respeito a \mathbf{x}

$$\min(\|\mathbf{M}\mathbf{x} = \mathbf{y}\|^2 + \lambda\|\mathbf{C}\mathbf{x}\|^2), \quad (\text{A.5})$$

onde

$$\|\mathbf{C}\mathbf{x}\|^2 = \sum_n \int C_n(z) |x^{(n)}(z)|^2 dz. \quad (\text{A.6})$$

O termo $x^{(n)}(z)$ é o enésimo termo derivativo de $x(z)$ e n é a ordem do derivativo máximo a ser computado. \mathbf{M} é a matriz que relaciona do dado desejado \mathbf{x} para observações e $\mathbf{C}\mathbf{x}$ é usado para impor uma tendência de preferência na solução.

Essencialmente, $\mathbf{C}\mathbf{x}$ é o termo que penaliza soluções não suaves. Restrições conhecidas na solução são utilizadas para a escolha das funções $C_n(z)$ e λ . Os estabilizadores de Tikhonov encontraram diversas aplicações nas interpretações dos dados de sensores (POGGIO; TORRE; KOCH, 1985).

A.3 Sensores de contato

Sensores tácteis são aqueles utilizados para se criar “um senso de toque”. São quase sempre sensores de contato, embora alguns aspectos do toque pode ser capturado com sensores de não contato, que passam muito perto da superfície do objeto e exploram fenômenos tais como capacitância ou indutância.

Sensores de toque são comumente utilizados em para-choques de robôs móveis. Os dispositivos mais simples constituem-se micro-chaves que retornam um valor binário: aberto ou fechado. Quando utilizados em para-choques, uma seqüência de micro-chaves ao redor do robô é geralmente montada em material complacente de tal forma que um pressão exercida em alguma parte do para-choque irá pressionar uma ou mais chaves. Sensores tácteis são também bastante utilizados nos elementos terminais, como, por exemplo, em garras de robôs.

Abordagens mais sofisticadas de sensoriamentos tácteis exploram dispositivos que retornam um sinal analógico proporcional à força aplicada nos sensores. Alguns destes sensores têm tecnologia de fabricação em materiais resistivos ou capacitivos complacentes que sofrem uma modificação na resistência ou capacitância em resposta a compressão.

A maneira com que os dados dos sensores instalados nos para-choques são manuseados, depende da arquitetura do robô móvel. Sensores de para-choque são os últimos recursos para se evitar colisões, mesmo porque, quando estes sensores detectam algo já é tarde demais. Algumas vezes este sistema é tratado pelo controle do robô como sensores, mas em muitos casos, eles são ligados diretamente no nível mais baixo do controle, assim como os nervos dos nossos joelhos são acoplados ao sistema muscular com respostas autônomas, sem a intervenção do cérebro. Por exemplo, no robô móvel Nomad 200 (SPROUSE, 2004), ilustrado na Figura 2.11, o para-choque (*bumper*) é um sistema sensor que retorna ao controle a posição angular e a altura do contato.

Embora o propósito dos sistemas de para-choques ativos com sensores embarcados seja o da prevenção de danos no caso de uma colisão, se a velocidade de impacto for alta, a detecção da colisão pode não ser o suficiente para evitar avarias. Muitos robôs móveis que são projetados para ambientes interiores não são equipados com sistemas de freios, e, mesmo que sejam, a quantidade de movimento (inércia) associada ao peso do robô pode significar uma considerável distância a ser percorrida até a parada total do robô. Em alguns casos, como os automóveis, os para-choques são instalados visando nem tanto as habilidades de sensoriamento, mas sim uma redução dos danos do choque, isto devido ao material de que são feitos proverem uma absorção da força de impacto.

A.4 Sensores inerciais

Sensores inerciais são uma classe de sensores externos que retornam as últimas referências do ambiente exterior. São sensores que medem as derivadas das variáveis da posição do robô. Normalmente o termo sensor inercial é usado para se referir a acelerômetros e giroscópios, que medem a segunda derivada da posição (i.e., a aceleração e a aceleração angular do robô) (BORENSTEIN et al., 1998). A utilização de múltiplos sensores inerciais e sensores externos para a compensação da limitação de algum sensor em particular, é uma abordagem comum em sistemas robóticos autônomos. Esta técnica é utilizada para se prover sensoriamento suficiente para a realização de toda a tarefa do sistema autônomo.

A.4.1 Acelerômetros

Acelerômetros são essencialmente massas montadas em dispositivos elásticos cujos deslocamentos sob aceleração pode ser medido. Estes sistemas exploram a lei de Newton $F = ma$ e a relação massa-mola $F = kx_2$ (em um sistema ideal). Resolvendo as duas equações para a aceleração a , obtém-se

$$a = \frac{kx^2}{m}, \quad (\text{A.7})$$

onde m é a massa, k é a constante elástica da mola (ou do dispositivo elástico) e x é o deslocamento linear da massa m sob a aceleração a . Na prática, outras alternativas para os sistemas mola-massa podem ser utilizados. Cada acelerômetro mede a aceleração ao longo de uma única direção. Montando-se três acelerômetros ortogonalmente, um em relação ao outro, um sensor de aceleração omnidirecional pode ser construído. Estes dispositivos podem ser encontrados comercialmente e são utilizados, por exemplo, em veículos de alta performance; para o monitoramento de condições de funcionamento de turbinas; na indústria aeronáutica, entre outros.

Acelerômetros são as vezes vistos como sensores que não necessitam de referências com o mundo exterior. Embora isto seja em parte verdadeiro, para se computar corretamente a aceleração o vetor da gravidade local deve ser conhecido em ambos os sentidos: direção e magnitude, para que, desta forma, a aceleração possa ser fatorada da leitura do sensor. Estes sensores possuem a característica versátil de poderem ser utilizados em sistemas robóticos sub-aquáticos, de superfície ou aéreos.

A.4.2 Giroscópios

Como no caso dos acelerômetros, os giroscópios medem a aceleração angular explorando a mecânica básica newtoneana. A forma mais familiar do dispositivo é o giroscópio mecânico que consiste de um rotor suspenso por um suporte formado por dois círculos articulados, com juntas tipo cardan. Seu funcionamento baseia-se no princípio da inércia. O eixo em rotação guarda direção fixa em relação ao espaço. A Figura A.1 ilustra um exemplo de giroscópio.

O giroscópio é um dispositivo usado para orientação de navios, aviões e espaçonaves, inventado por Léon Foucault¹ em 1852. O giroscópio veio a substituir a bússola na navegação marítima. Na aviação, serve de girocompasso e piloto automático, permitindo o voo em condições de visibilidade zero. Nos vôos espaciais o dispositivo é fundamental para a orientação das espaçonaves.



Figura A.1: Exemplo de giroscópio mecânico.

O giroscópio pode ser descrito como um compasso diferencial, ou seja, um dispositivo para medição da orientação diferencial ou relativa. A principal dificuldade com a estimação da orientação feita pelo giroscópio é que pequenos erros de medida acontecem com frequência, o que causa a perda gradual da exatidão por acumulação de erros, que é referida como *deriva*

¹Jean Bernard Léon Foucault foi um físico e astrônomo francês, nascido em Paris em 18 de setembro de 1819 e falecido em Paris em 11 de fevereiro de 1868. Foucault era filho de um editor de Paris. Estudou medicina, que ele abandonou rapidamente para se dedicar à Física. Interessou-se aos métodos fotográficos de Daguerre. Foucault é particularmente conhecido pela sua experiência demonstrando a rotação da Terra em torno de seu eixo (pêndulo de Foucault), tendo também determinado a velocidade da luz e inventado o giroscópio.

(*drift*). Em sistemas mecânicos, a deriva vem da fricção e da inevitável imperfeição dos eixos cardan. A taxa de crescimento destes erros pode variar de décimos de graus até dezenas de graus por segundo e depende da precisão da máquina, que, por sua vez, é determinada pelo seu preço.

Uma abordagem alternativa para a medição da orientação absoluta é a exploração do fenômeno óptico descoberto por Geoges Sagnac, que veio a ser denominado *efeito Sagnac*². Devido ao fato do efeito ser um fenômeno puramente óptico, isto permite a construção de giroscópios ópticos sem partes móveis.

Uma descrição alternativa do efeito Sagnac é visualizar uma mudança dl no comprimento da cavidade como resultado de um movimento rotacional com velocidade angular Ω (CHOW et al., 1985):

$$dl = 4\pi \frac{r_2 \Omega}{c}, \quad (\text{A.8})$$

onde r é o raio da cavidade e c é a velocidade da luz no meio na cavidade.

Para uma onda estável ser produzida, o comprimento da onda da luz deve dividir o comprimento da cavidade em números inteiros de vezes. Como resultado, variações no curso do comprimento levam a mudanças no comprimento de onda da luz. A variação na frequência induzida pela variação no comprimento é dado por

$$df = \frac{2r\Omega}{\lambda}, \quad (\text{A.9})$$

onde λ é o comprimento de onda da luz medido.

O efeito original observado por Sagnac envolveu o deslocamento de fase $\delta\phi$ da luz não coerente em um interferômetro em anel definido por

$$\delta\phi = \frac{8\pi\Omega A}{\lambda c}, \quad (\text{A.10})$$

onde A é a área do interferômetro.

Uma metodologia alternativa permite que o efeito Sagnac seja observado sem a utilização de raios laser, utilizando-se interferometria, o que torna esta abordagem mais conveniente para os sistemas robóticos móveis. Tecnologias empregando fibras ópticas tem conduzido,

²O efeito de Sagnac vem da experiência que foi inicialmente planejada para medir a velocidade angular da Terra em seu movimento de rotação no espaço. Utiliza o princípio da interferometria em anel. Um raio de luz é inicialmente dividido em dois. Os dois raios resultantes percorrem uma trajetória fechada em sentidos opostos. Os dois raios se recombinaem ao sair pelo mesmo ponto do anel em que entraram. Ao fazê-lo, produzem um padrão de interferência em que a distância entre a posição original das franjas e suas novas posições dependem diretamente da velocidade angular do sistema. Foi levada a cabo em 1913 por G. Sagnac.

recentemente, a uma redução significativa do preço dos giroscópios ópticos com aperfeiçoamentos em sua robustez e disponibilidade.

Giroscópios ópticos têm derivado estes resultados de imperfeições ópticas, elétricas e ruídos. Sem compensação especial, alguns sistemas são incapazes de fornecerem medidas em baixas velocidades angulares.

Unidades típicas de baixo custo podem ter taxas de derivação na ordem de vários graus por hora, enquanto que sistemas mais robustos utilizados em navegação aeroespacial podem ter precisão de até 0,001 grau por hora (DUDEK; JENKIN, 2000).

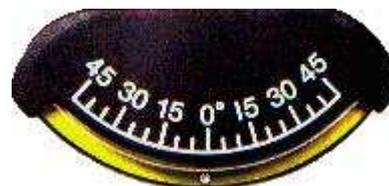
A.4.3 Inclinômetros e bússolas

A.4.3.1 Inclinômetros

Um inclinômetro é um dispositivo simples que mede a orientação do vetor gravitacional. Inclinômetros digitais são os correspondentes modernos aos prumos e às régua de nível utilizadas por carpinteiros e pedreiros. A Figura A.2 ilustra exemplos de inclinômetros comerciais utilizados no auxílio à navegação.



(a) Inclinômetro com bússola



(b) Inclinômetro de esfera

Figura A.2: Exemplos de inclinômetros comerciais.

Embora isto possa parecer trivial, a importância de se integrar um inclinômetro em sistema robóticos móveis não pode ser desprezada. Um sistema robótico em que ocorram falhas nos sensores internos de inclinação, perde a capacidade de determinar com precisão sua inclinação e conseqüentemente, seu controle não irá responder a inesperadas rampas, solavancos ou depressões no ambiente, o que pode levar a efeitos catastróficos.

A.4.3.2 Bússolas magnéticas

Os inclinômetros medem desvios do quadro gravitacional, enquanto que as bússolas executam medidas da orientação com respeito ao campo magnético da terra. Bússolas magnéticas mecânicas utilizam o campo magnético da terra para orientação, onde a agulha magnética móvel é posicionada em torno de um eixo que passa pelo seu centro de gravidade, montada, geralmente, em caixa com limbo graduado.

A.4.3.3 Giro-bússolas

Giro-bússolas (*Gyrocompasses*) são giroscópios modificados de tal forma que apontem no sentido do pólo norte magnético da terra. A Figura A.3 ilustra exemplos de giro-bússolas.



(a) Recorte de uma giro-bússola



(b) Giro-bússola

Figura A.3: Exemplos de giro-bússolas.

Uma giro-bússola é essencialmente um giroscópio, uma roda girando de forma que seu eixo é livre para se orientar em qualquer direção. Pela lei da conservação do momento angular, esta roda irá se manter na orientação original. Com a rotação da terra, um observador estacionário na terra tem a impressão de que o eixo do giroscópio está rodando uma vez a cada 24 horas; o eixo sempre aponta para mesma estrela. O ingrediente adicional crucial necessário a giro-bússola é a fricção: o giroscópio não está, de fato, totalmente livre para se reorientar. Se o dispositivo conectado ao eixo está imerso num fluido viscoso, então este

fluido irá resistir a reorientação do eixo. Esta força de ficção causada pelo fluido resulta num torque atuando no eixo, forçando o eixo a se virar numa direção ortogonal à força. Com o tempo, isto tem o efeito de que o eixo apontará para o norte verdadeiro, que se permanecerá estacionário. Isto acontece porque o norte verdadeiro é a única direção para onde a giro-bússola pode permanecer apontada na superfície da terra que não necessita de mudança. É considerado um ponto de energia potencial mínima.

A giro-bússola têm duas principais vantagens em relação às bússolas magnéticas:

- Apontam para o norte verdadeiro da terra (norte geográfico, na direção do eixo de rotação da terra);
- Não são afetadas pelo casco de metal das embarcações ou dos robôs móveis.

A.4.3.4 Bússolas de fluxo de porta

Uma bússola de fluxo de porta (*flux-gate compass*) incorpora um sistema eletromagnético de corrente alternada (AC) cujo qual é desbalanceado na presença de um campo magnético externo direcionado. Este desbalanceamento eletromagnético induz tensões nas bobinas do sistemas. A amplitude e a fase das tensões indicam a relativa orientação do sistema e do campo. A Figura A.4 ilustra um esquema elétrico desta tecnologia, onde α é o desvio angular da linha de referência norte-sul da terra, u_{ex} é a tensão de excitação do sistema, u_o é a tensão de saída, os pontos 1, 2 e 3 indicam núcleos com bobinas do sistema eletromagnético.

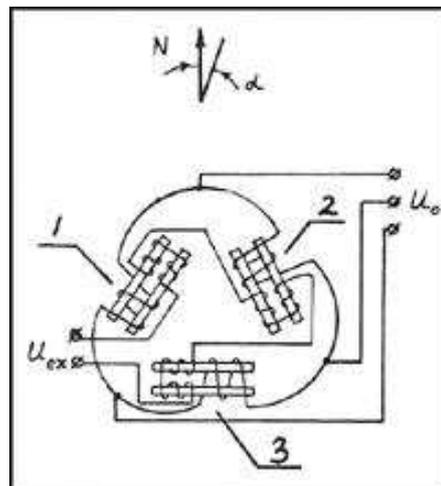


Figura A.4: Esquema elétrico de uma bússola de fluxo de porta.

A vantagem desta tecnologia sobre a bússola magnética é que a leitura é eletrônica e pode ser digitalizada (através de um conversor analógico-digital, por exemplo) e transmitida facilmente, exibida remotamente, e utilizada pelo controle do robô para correções de curso.

Para se evitar inexatidão criadas pelo componente vertical do campo, o alinhamento da bússola de fluxo de porta deve ser mantido o mais raso possível montando-a em eixos cardan ou utilizando-se de sistemas de suspensão fluídica. A Figura A.5³ ilustra um exemplo comercial de uma bússola de fluxo de porta em suspensão fluídica. Este amortecimento viscoso causado pelo fluido aumenta a relação sinal-ruído do sistema.



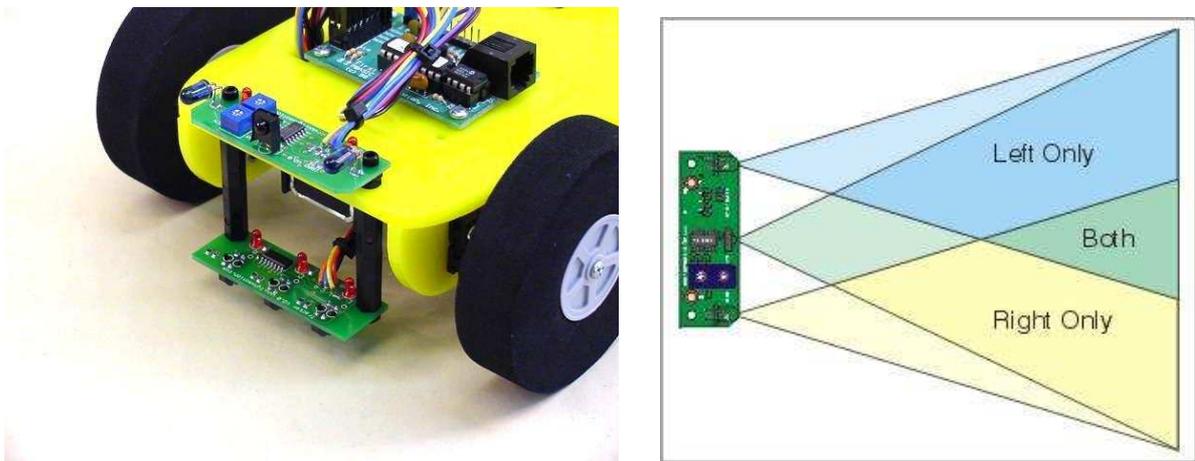
Figura A.5: Bússola de fluxo de porta em suspensão fluídica.

A.5 Sensores infravermelho

Detectores infravermelho de proximidade normalmente são sensores rápidos e baratos, e, por esses motivos, dentre outros, são bastante utilizados em robótica móvel. O conceito básico é a emissão de um pulso infravermelho e a detecção do sinal refletido. A distância do alvo é estimada utilizando-se do sinal reconstruído e suposições a respeito da reflectibilidade das superfícies do ambiente. Para se evitar potenciais confusões com outras fontes de infravermelho no ambiente, o sinal emitido é codificado, e sinais detectados são ignorados a

³Cortesia de The Autonav Coop

menos que a forma de onda apropriada seja detectada. A Figura A.6 ilustra um robô móvel com um sistema de sensor infravermelho instalado em sua parte posterior.



(a) Instalação na parte traseira

(b) Campo de atuação dos sensores

Figura A.6: Exemplo de sensores infravermelho nos robôs móveis.

Sinais infravermelhos não codificados são susceptíveis a interferência (e.g., da luz do sol). Mesmo um sinal mais complexo pode ser encoberto se a iluminação do ambiente for muito forte. Contudo, devido ao baixo custo, sistemas transmissores/detectores de infravermelho são frequentemente utilizados em robôs móveis. Estes sistemas normalmente utilizam um formato de anel em redor do robô, cobrindo todos os ângulos de ação.

O processo de se deduzir a distância do robô até o obstáculo, através das informações retornadas pelos sensores infravermelhos não é trivial porque, a priori, é praticamente impossível de se estimar as propriedades reflexivas das superfícies destes objetos. A intensidade do sinal retornado também é função da distância e das propriedades reflexivas das superfícies (TAN; XI, 2001). Por outro lado, o mecanismo tem um custo baixo, é compacto e relativamente ortogonal em relação a outros dispositivos sensores que possam ser montados em sistemas robóticos embarcados. Desta maneira, sensores infravermelho de proximidade, como o exemplo da Figura A.6, tendem a ser utilizados como sensores de não-contato, sendo uma extensão aos sensores de contato dos para-choques, do que ser utilizados como dispositivos para se medir distâncias com precisão.

A.6 Sonar

Sonar (Sound Navigation and Ranging) refere-se a sensores de distância que utilizam sinais acústicos. Sonar é uma tecnologia de sensoriamento ativo por meio do qual um sinal ou pulso sonoro é emitido e sua reflexão, quando encontra um anteparo, é subsequentemente recebida. O tempo de resposta, o deslocamento de fase e a atenuação do sinal como uma função da frequência, são aspectos do sinal refletido que são explorados pelos diferentes tipos de sistemas sonar.

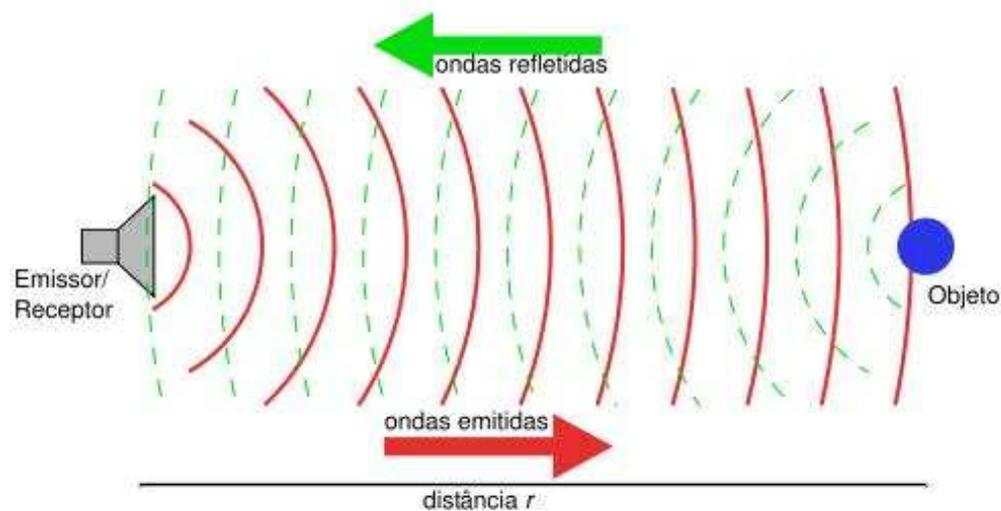


Figura A.7: Ilustração do princípio de funcionamento do *sonar*.

O sonar foi construído pelo físico francês Paul Langevin em 1917, para localizar submarinos alemães. Apesar de já estar pronto, não foi usado antes do final da Primeira Guerra Mundial.

Em 1918 ultra-sons eram utilizados para determinar posição, velocidade e orientação de objetos subaquáticos. A maioria dos sonares terrestres são baseados em sinais acústicos na faixa de frequência dos ultra-sons.

O sonar é muito utilizado para orientar a navegação, obter o perfil da placa marítima, revelar a presença de cardumes, etc. Na água, consegue-se uma precisão muito maior do que no ar, uma vez que a velocidade de propagação do som na água é muito maior, podendo chegar a até 1498m/s.

Nos robôs móveis, os sistemas sonar são instalados tipicamente montando-se unidades

emissoras-receptoras em posições fixas na base do veículo. Uma estratégia comum é posicionar unidades de sonar em intervalos angulares uniformes ao redor do robô, perfazendo-se um sistema sonar omnidirecional. Isto é possível porque os dispositivos sonares têm, relativamente, baixos custos.

Sistemas sonares subaquáticos são utilizados por golfinhos, baleias, e outras espécies de cetáceos. Cada uma destas espécies emitem chilreio característicos que são reconhecidos pelos outros integrantes do grupo. No caso dos morcegos, que utilizam sistemas sonares aéreos, a reflexão diferencial de um determinado objeto em função do comprimento de onda, permite ao morcego identificar várias características deste objeto. Esta informação é utilizada para encontrar e identificar insetos que são cassados pelo morcego, geralmente a noite. A multiplexação de frequências, as estratégias de modulação por pulso e os mecanismos de interpolação utilizados pelo morcegos são extremamente complexos, variam de espécie para espécie e ainda não são totalmente compreendidos.

Sensores sonares utilizados em robôs móveis normalmente usam estratégias mais simples. A mais simples e mais comum delas é a utilização do mesmo transdutor como transmissor e receptor, transmitindo um curto pulso de ultra-som e medindo o tempo de atraso de resposta do eco. Outras técnicas podem ser utilizadas; Stanley (STANLEY; MCKERROW, 1997), por exemplo, utiliza um transmissor e dois receptores para implementar o processo de medição. A metodologia padrão é enviar um curto sinal de alta tensão e alta frequência através do transdutor para produzir um pulso acústico. Após o envio do pulso, o transdutor alterna-se para o modo de recepção e passa a atuar como um microfone. Devido a oscilações residuais no transdutor imediatamente após o envio do pulso ultrasônico, este fica impedido de detectar ecos que estejam chegando, de uma maneira confiável, por um breve intervalo de tempo. Este intervalo de tempo é conhecido como *intervalo vazio* (*blanking interval*). Como consequência, estes transdutores de ultra-som são insensíveis a obstáculos que estiverem muito próximos ao robô. Tipicamente, a distância para uma medição confiável deve ser maior do que 6 cm, embora possa variar dependendo das características do sensor e dos detalhes da implementação.

Se a onda sonora viaja diretamente ao objeto, é refletida e retorna diretamente ao transdutor, a distância do mesmo ao objeto é metade da distância percorrida pelo pulso de ultra-som. Esta distância é dada por

$$d = \frac{1}{2}ct, \quad (\text{A.11})$$

onde d é a distância ao objeto, c é a velocidade do som no ar e t é o tempo decorrido entre a transmissão e a recepção do sinal de eco. Uma aproximação da velocidade do som no ar

pode ser equacionado por

$$c = c_0 + 0,6T \text{ m/s}, \quad (\text{A.12})$$

onde T é a temperatura em graus Celsius e $c_0 = 331 \text{ m/s}$.

A velocidade do som é também afetada pela densidade do ar e, conseqüentemente, existe o efeito da altitude, assim como outros fatores, tais como umidade ou concentração de poeira. A velocidade do som c varia na razão direta da raiz quadrada da temperatura (em Kelvin) e é dependente da umidade e outras propriedades do meio. Em ambientes normais de operação, é aceitável uma variação de 2% em c . A Figura A.8 ilustra um modelo de transdutor ultrassônico utilizado em robôs móveis⁴.



Figura A.8: Exemplo de um sistema sonar.

Dependendo da forma do transdutor sonar, o perfil da amplitude espacial M da frente de onda emitida pode ser formado desde o início até o limite da extensão de sua trajetória. Uma vez que a frente de onda acústica estiver longe do transdutor, ela assemelha-se com uma parte de uma esfera em expansão, com a amplitude do sinal sendo reduzida de acordo com o inverso da lei quadrática. Essa dispersão esférica geralmente toma forma a partir de 1 m da superfície do transdutor. A distribuição da energia nesta superfície esférica é não uniforme, com uma amplitude oscilante, que é função do ângulo perpendicular a superfície do transdutor. Quanto mais longe percorre a onda, mais atenuada ela se torna. Desta forma, o eco que tem um percurso longo de retorno, normalmente tem um sinal muito fraco. Isto é tipicamente compensado, em parte, aumentando-se o ganho do receptor como uma função do tempo de retorno do sinal. Um fator adicional que influencia a amplitude do sinal de eco é a refletância dos objetos, que pode variar bastante, dependendo das características reflexivas do material.

⁴Gentileza de Polaroid Coop.

A frequência acústica utilizada nos sonares comerciais está normalmente situada na faixa de 40-50 kHz. Altas frequências são atenuadas mais rapidamente, embora resultem numa melhor resolução espacial. O sinal emitido tem um perfil de amplitude complexa que é função do ângulo de distanciamento a normal do transdutor. A Figura A.9 ilustra a curva característica da amplitude da onda no plano, de um transdutor ultra-sônico, em função da orientação com relação a normal do transdutor. Nota-se que, a 15° do lobo principal (que está alinhado com a normal do transdutor) a atenuação do sinal está em torno de -35dB.

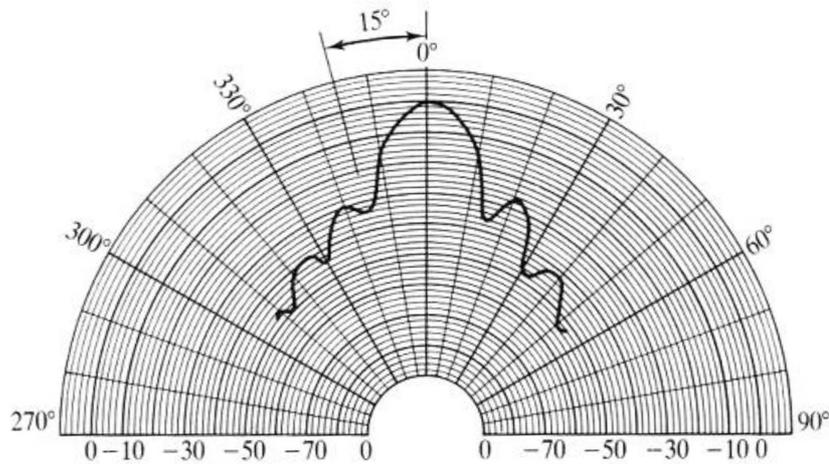


Figura A.9: Amplitude em dB do sinal sonar em função da orientação com relação a normal do transdutor.

Na realidade, a forma de onda 3D (tridimensional) do sonar é em forma de cone com distribuição de energia não uniforme em sua superfície. A máxima energia está localizada no centro, alinhada com a normal do sonar. Uma implicação deste formato característico da onda é que um eco retornado pode ser uma reflexão num ângulo oblíquo de uma porção de onda que foi transmitida. Então, um eco retornado não necessariamente resulta em uma localização precisa de um objeto que refletiu a onda, utilizando-se apenas a metodologia de cálculo por tempo de resposta.

Outra dificuldade para se deduzir a localização de uma superfície refletora é que, nas frequências ultra-sônicas, a maioria dos objetos comuns são refletores especulares. Isto significa que, nestas frequências, a refletância dos objetos é próxima de 100%, funcionando com um espelho ao ultra-som, de tal forma que uma onda sonora oblíqua (a superfície reflexiva não está perpendicular ao transdutor, como na maioria dos casos) será refletida para longe do emissor com uma parcela pequena da energia retornada. Se um eco retorna ao robô, não

existe certeza de que este eco não seja o resultado de uma soma complexa de ondas refletidas ao redor do ambiente em que se encontra do robô, principalmente quando se encontrar em locais fechados. Existe também o problema relacionado a objetos com espessuras relativamente pequenas, cujas ondas refletidas têm pouca energia, sendo consideradas, na maioria das vezes, como espúrios pelos controladores dos robôs móveis (KLEEMAN; KUC, 1995).

Embora as reflexões especulares e as dificuldades de localização das superfícies refletoras são, na maiorias da vezes, tratadas como ruído, elas são de fato, características repetitivas do sensor que podem ser modeladas com precisão (WILKES et al., 1992) (KUC; SIEGEL, 1987).

Sinais acústicos que retornam ao robô após uma reflexão, são freqüentemente conhecidos como reflexões de primeira ordem. Ecos que retornam de duas ou mais reflexões são chamados de reflexões de segunda ordem, terceira ordem, etc. É comum a implementação de filtros (digitais ou analógicos) para que sejam mantidas somente as reflexões de primeira ordem, descartando-se as demais. A Figura A.10 ilustra os casos mais comuns da interação do sonar com os obstáculos encontrados em ambientes comuns.

A.6.1 Modelamento de transdutores ultra-sônicos

Vários modelos de transdutores já foram propostos. Para os transdutores ultra-sônicos, Kuc e Siegel (KUC; SIEGEL, 1987) construíram um modelo de transdutor sonar com resposta impulsiva que é muito utilizado em simulações. No decorrer deste trabalho, este modelo é utilizado na implementação de um dos blocos do sistema de controle que é utilizado na simulação e validação do sistema proposto.

Utilizando um modelo de sistema linear, pode-se expressar a forma de onda detectada no receptor sonar em função do tempo $r(t)$, que é a convolução de duas funções no tempo:

$$\begin{aligned} r(t) &= \int_{-\infty}^{+\infty} p(\tau)h(t - \tau)d\tau \\ &= p(t) * h(t), \end{aligned} \tag{A.13}$$

onde $p(t)$ é a forma de onda do pulso e $h(t)$ é a resposta impulsiva, que representa a contribuição das reflexões nas estruturas. A maior vantagem desta formulação com resposta impulsiva é que provê uma separação parcial entre os efeitos do transdutor e do refletor (o eco). Esta separação permite com que se examine eficientemente o mesmo conjunto de reflexões com os transdutores possuindo diferentes características.

A forma de onda utilizada é um pulso correspondente ao eco de um refletor especular

ideal, como por exemplo um plano normalmente incidente, localizado a uma distância do transdutor $z \gg \frac{a^2}{\lambda}$, onde a é o raio do transdutor e λ é o comprimento de onda da radiação. Uma forma de onda típica deste pulso pode ser aproximada por uma senoidal modulada por uma gaussiana (conforme ilustra a Figura A.11(a)):

$$p(t) = e^{-\frac{t^2}{2\sigma^2}} \sin(2\pi f_R t), \quad (\text{A.14})$$

para $-3\sigma \leq t \leq 3\sigma$, onde σ é duração do pulso e f_R é a frequência de ressonância do transdutor.

Assumindo-se que um objeto reflete o pulso de um sonar e está orientado a um ângulo α em relação ao eixo ortogonal do transdutor, então, a resposta do impulso que chega ao transdutor a um ângulo α da normal do sonar é dado por

$$h_R(t, z, a, \alpha) = \frac{2c \cos \alpha}{\pi a \sin \alpha} \left[1 - \frac{c^2(t - \frac{2z}{c})^2}{a^2 \sin^2 \alpha} \right]^{\frac{1}{2}} \quad (\text{A.15})$$

para c , que é a velocidade do som no ambiente, dado pelo equação A.12, sendo que

$$\frac{2z - a \sin \alpha}{c} \leq t \leq \frac{2z + a \sin \alpha}{c}, \quad (\text{A.16})$$

e

$$0 < |\alpha| \leq \frac{\pi}{2}. \quad (\text{A.17})$$

A expressão A.15 é substituída pela função delta quando $\alpha = 0$. Neste caso, a frente de onda é transmitida e é recebida em todas as partes do transdutor simultaneamente. Assim tem-se que

$$h_R(t, z, a, \alpha) = \delta(t - \frac{2z}{c}), \quad (\text{A.18})$$

para $\alpha = 0$, onde $\delta(t)$ é a função delta de Dirac (SCHWARZ; FRIEDLAND, 1964). Quando tem-se a convolução de h_R com a forma de onda que descreve o sinal elétrico usado para a excitação do transdutor (tipicamente um sinal senoidal modulado por um pulso gaussiano), a intensidade com que o sinal do sonar é atenuado é determinado como uma função do ângulo α com que o sinal resultante deixa (ou chega) do transdutor. A Figura A.11(a) ilustra o sinal inicial do sonar. A Figura A.11(b) ilustra o formato da função gaussiana que será utilizada para a convolução com o pulso do sonar. A Figura A.12(a) ilustra o resultado da convolução da resposta do impulso do transdutor com a função de propagação pontual. A Figura A.12(b) ilustra a atenuação do sinal em função do ângulo do transdutor.

Em adição a reflexão especular, os sinais do sonar podem sofrer difração pelos cantos e

outras bordas que contenham ângulos retos. Devido ao fato de que a energia é transmitida em várias direções, os efeitos da difração são normalmente significativos a pequenas distâncias. Um modelo simplificado deste fenômeno pode ser conseguido assumindo-se que cantos com ângulos retos retornam padrões esféricos de refletância.

Se, somente a interação entre um único transdutor e um único obstáculo no ambiente for considerado, o erro da distância associado a uma dada resposta do sonar pode ser melhor modelado pela distribuição normal do sinal do sonar (LANG; KORBA; WONG, 1989). Uma simulação para este tipo de modelo de erro de sonar é mais simples de ser implementada. Pode-se também, utilizando-se outras abordagens, tais como os mínimos quadrados ou o filtro de Kalman, produzir algoritmos empregados com sucesso na integração de medições sonares, onde os ruídos associados com cada medição pode ser tratado independentemente, e cada um pode ser modelado por algum processo gaussiano (LEONARD; DURRANT-WHYTE, 1991) (MALIK; POLKOWSKI, 1991). Infelizmente, poucos ambientes consistem de apenas uma superfície. Para ambientes reais mais complexos, ainda não se conseguiu um bom modelo pela distribuição gaussiana independente para erros dos sonares, porque são altamente correlatos e muito complexos (MONTEMERLO; THRUN; WHITTAKER, 2002).

Para se modelar com mais precisão as interações complexas entre os pulso do sonar e ambientes com múltiplas superfícies, um algoritmo de rastreamento de raios pode ser usado para se seguir o caminho do pulso quando o mesmo interage com diferentes estruturas do ambiente (WILKES et al., 1992). Num algoritmo computacional gráfico de rastreamento de raios, o pulso emitido pelo sonar é rastreado desde a origem até interagir com as estruturas do ambiente e ser refletido para o transdutor. Se o sinal for perdido ou atenuado além da capacidade de detecção ele será rastreado até sua condição final.

A.6.2 Interpretação dos dados do sonar

Em adição ao problema das múltiplas reflexões, a interação do som emitido com um superfície simples é também muito complexo (DUDEK et al., 1996). Uma forma simplificada, porem inexata, para interpretação desta interação é modelar um objeto diretamente na frente do transdutor com a distância medida utilizando-se a equação A.11. Deve ser notado que o padrão de resposta que será observado neste caso não será distribuído normalmente, embora alguns pesquisadores tenham feito esta assunção conveniente. A repetibilidade do padrão das respostas do sensor leva a alguns fenômenos característicos, como por exemplo paredes ilusórias. A Figura A.13 ilustra as medidas do sonar tomadas pelo robô móvel, representado

pelo círculo, num ambiente de um laboratório. O caminho e o sentido percorrido pelo robô são mostrados na linha pontilhada. As medidas do sonar são agrupadas em estruturas lineares (paredes). As medidas foram tomadas de vários pontos no ambiente quando o robô se movia ao longo do caminho mostrado na Figura A.13. A planta baixa descreve algumas das características das saliências dos objetos neste ambiente.

As respostas do sonar são freqüentemente agrupadas em arcos circulares conhecidas como *regiões de distância constante* (RDC)(LEONARD; DURRANT-WHYTE, 1992). Estes grupos de medições refletidas acontecem porque uma série de frentes de ondas são todas refletidas a partir do mesmo ponto (ou muito próximas do mesmo ponto) de um objeto. A Figura A.14 ilustra as regiões de distância constante com a reflexão em formato de arco para uma superfície plana. Um grupo de transdutores no ponto **S** emitem sinal que atingem uma superfície plana no ponto **P**. O transdutor apontado diretamente ao ponto **P** retorna um tempo associado com a distância **r**. Outros transdutores que estejam apontados em outras direções também retornam um tempo associado a distância **r** devido à vasta extensão do sinal. Um plano não é observada como uma parede plana mas como uma região de profundidade constante ou região de distância constante (RDC).

A estrutura de arco circular que resulta das paredes e dos cantos côncavos e convexos é repetitivo. Em princípio, a extensão do arco refletido pode ser utilizado para se diferenciar entre uma reflexão de um canto e de uma superfície plana. Na prática, isto é difícil porque depende da superfície em questão ter propriedades de reflexão acústicas uniformes (GAO; HINDERS, 2006). Uma alternativa é explorar variações do comportamento dos arcos refletidos sob pequenas movimentações do robô para se diferenciar entre diferentes classes de obstáculos (paredes, cantos convexos e cantos côncavos)(LACROIX; DUDEK, 1997). Para o mesmo tipo de parede (mesmo material), as reflexões para os cantos côncavos deve, teoricamente, ser duas vezes mais extensos do que o arco refletido para paredes planas, cujos quais devem ser mais largos do que a RDC para os cantos convexos. Independente do uso da técnica de RDC para se classificar as estruturas dos obstáculos no ambiente, as RDCs têm sido usadas com sucesso como medidas primitivas para navegação e exploração do ambiente de trabalho do robô móvel (LEONARD; DURRANT-WHYTE, 1992)

A.7 Radar

O *Radar* (*Radio Detecting and Ranging*)(Detecção e Telemetria pelo Rádio) é parecido com o sonar em seu princípio de operação. São transmitidas ondas de rádio de alta freqüência,

e as reflexões são observadas para se obter medições de distâncias e outras propriedades.

O radar é composto por uma antena transmissora e receptora de sinais para Super Alta Freqüência (SHF), a transmissão é um pulso eletromagnético de alta potência, curto período e feixe muito estreito. Durante a propagação pelo espaço, o feixe se alarga em forma de cone, até atingir ao alvo que está sendo monitorado, sendo então refletido, e, retornando para a antena, que neste momento é receptora de sinais.

Como se sabe a velocidade de propagação do pulso, e pelo tempo de chegada do eco, pode-se facilmente calcular a distância do objeto. É possível também, saber se o alvo está se afastando, ou se aproximando da estação, isto se deve ao Efeito Doppler, isto é, pela defasagem de freqüência entre o sinal emitido e recebido.

Até recentemente esta tecnologia tinha um custo relativamente alto (comparativamente aos sistemas sonares, por exemplo) para os sistemas robóticos móveis. Recentemente, progressos tem permitido que os radares possam ser usados em veículos robóticos móveis. O radar é um sistema atrativo para robôs móveis porque é mais rápido e provê informações sobre propriedades da superfície, tal como por exemplo, a geometria, que são mais difíceis de serem detectadas por outras tecnologias (e.g. os sonares). Por exemplo, o sistema radar pode ser utilizado para a discriminação entre diferentes tipos de terrenos o que pode ser útil na estimação da atravessabilidade. O radar pode penetrar a camada da superfície de um objeto e pode fornecer informações de sua estrutura. Devido ao fato do radar ser baseado em ondas de rádio, ao invés de ondas sonoras, ele pode ser utilizado em ambientes com ausência da atmosfera, como por exemplo, em sistemas robóticos para exploração espacial.

A.8 Laser

O laser (*Light Amplification by Stimulated Emission of Radiation*) ou amplificação da luz por emissão estimulada de radiação, é um dispositivo que produz radiação eletromagnética com características muito especiais: ela é monocromática (possui freqüência muito bem definida) e coerente (possui relações de fase bem definidas), além de ser colimada (propaga-se como um feixe). Quando esta radiação tem freqüência visível, chama-se luz. O efeito físico por trás de seu funcionamento é a emissão estimulada, descoberta pelo físico Albert Einstein como condição necessária ao equilíbrio térmico da radiação com a matéria. Einstein descobriu, através de considerações teóricas, que não apenas um átomo absorve um fóton (a partícula de luz) incidente e o reemite ao acaso após certo tempo (emissão espontânea), mas que também este mesmo átomo deve reemitir seu fóton absorvido se um segundo fóton

interage com ele. O fóton reemitido tem a mesma frequência do fóton que o estimulou e, igualmente importante, tem a mesma fase.

Um laser funciona desde que se consiga excitar um número mínimo de átomos de determinado material para um nível de energia superior, de modo a se obter uma inversão de população (quando existem mais átomos excitados do que átomos no estado fundamental). Quando isso ocorre, a emissão espontânea de fótons, que acontece naturalmente a todo tempo, é amplificada pelos átomos vizinhos, que vão emitir fótons estimulados pelos primeiros.

Estes fótons, por sua vez, estimulam a emissão de outros, num efeito cascata. Para que tudo isso funcione, entretanto, é necessária uma realimentação, ou seja, sempre manter fótons emitidos estimuladamente interagindo com os átomos. Isso é obtido com uma cavidade ótica, uma região do espaço em que se confina luz por algum tempo com o uso de espelhos altamente refletores e convenientemente alinhados. Entretanto, para se compreender perfeitamente um laser, faz-se necessário o uso da mecânica quântica.

Por suas propriedades especiais, o laser é hoje utilizado nas mais diversas aplicações: médicas (cirurgias), industriais (cortar metais, medir distâncias), pesquisa científica (pinças óticas, física atômica, ótica quântica, resfriamento de nuvens atômicas, informação quântica), em sistemas robóticos móveis, comerciais (comunicação por fibras óticas, leitores de códigos de barras), e mesmo todos os dias em nossas casas (aparelhos leitores de CD e DVD, laser pointer usado em apresentações com projetores).

É produzido por materiais como o cristal de rubi dopado com safira, mistura de gases no caso do hélio e neônio, dispositivos de estado sólido como diodos laser, moléculas orgânicas como os laser de corante.

A.8.0.1 Telêmetro a laser

Um dos importantes objetivos dos sensores nos sistemas robóticos móveis são as estimativas das distâncias entre o robô e os objetos no ambiente. Uma das tecnologias preeminente para se alcançar este objetivo é o telêmetro a laser. É baseado em uma (ou mais) das seguintes metodologias alternativas (DUBRAWSKI; SIEMIATKOWSKA, 1998):

- **Triangulação.** A utilização de relações trigonométricas entre o feixe de luz que deixa o sensor, o raio de luz que retorna e sua posição no plano.
- **Tempo de percurso.** A medida do atraso de tempo entre a saída do raio de luz do sensor e seu retorno, após reflexão num anteparo.

- Diferença de fase.** Baseado na diferença de fase entre a fase do sinal emitido e retornado.

O princípio da triangulação dos telômetros a laser é essencialmente o mesmo para os sistemas baseados em fontes de luz normais. As fontes de laser, contudo, tem a vantagem de serem colimadas.

O tempo de percurso e a diferença de fase para o sistema de sensores a laser, por outro lado, exploram o tempo de atraso da propagação do sinal do que efeitos geométricos. O uso de um feixe de luz estreito e colimado pode permitir uma alta resolução e boa eficiência em termos de potência de transmissão do sinal. Isto é possível, com os sistemas laser em particular, porque são fontes coerentes que podem produzir pulsos muito breves que permanecem colimados através de distâncias substanciais. A tecnologia de sensoriamento é baseada nas medições do atraso ou diferença de fase entre o sinal de laser emitido e a reflexão retornada. O processo de medição pode ser interferométrico ou direto. Em ambos os casos requerem um significativo processamento do sinal. O alcance máximo do sensor é limitado pela potência do sinal emitido. Sistemas de baixa potência tem uma faixa de operação de alguns metros, enquanto sistemas de alta potência podem funcionar a distâncias de quilômetros. Esta é a mesma tecnologia que tem sido utilizada para medições da distância da terra para superfícies lunares com alta precisão. Como o feixe de laser diverge com a distância, a localização da medição da distância se torna menos precisa nos alvos mais remotos.

Sistemas laser são tipicamente fixados em bases inclináveis ou em montagens com espelhos, de tal forma que possam direcionar o feixe para diferentes partes do ambiente. Dada a precisão e a repetibilidade dos sistemas laser, eles são uma alternativa efetiva a outras tecnologias de sensoriamento voltada a estimativa de posicionamento de robôs móveis. A Figura A.15 mostra duas varreduras de sensores laser em num mesmo ambiente tomadas de duas posições diferentes. Nos sistemas descritos por Lu e Milos (LU; MILIOS, 1997a) (LU; MILIOS, 1997b), um processo de otimização não linear é usado para o alinhamento das duas varreduras e então determinar o movimento relativo do robô entre as varreduras.

Nas Figuras A.15 e A.16, o robô foi da posição A para a posição B , mas apresentou erro no odômetro e fez com que “pensasse” que estivesse na posição B' . Os pontos de varredura relativos ao posicionamento A e B são marcados com x 's e pequenos círculos, respectivamente. Os posicionamentos são indicados pelas setas. A Figura A.15 mostra o erro de alinhamento relativo a diferença entre o posicionamento B' e o verdadeiro posicionamento B . A Figura A.16 mostra o resultado do alinhamento das duas varreduras laser. O posicionamento B' é corrigido para a posição verdadeira B pela mesma transformação. Como pode ser visto

dos dados, as respostas dos sistemas de sensores laser a partir de superfícies lisas, como por exemplos as paredes, proporcionam medições consistentes das estruturas do ambiente.

Nos ambientes internos, a maior preocupação com os sistemas baseados na tecnologia laser é o risco de acidentes com os olhos dos observadores. Para tanto, os dispositivos utilizados em robôs móveis são tipicamente de baixa energia, não prejudicando a visão dos observadores, uma vez que a interação com os objetos acontecem em relativas curtas distâncias.

A.9 Sistema de posicionamento baseado em satélite

O Sistema de Posicionamento Global, vulgarmente conhecido por GPS (do acrônimo do inglês *Global Positioning System*), é um sistema de posicionamento por satélite, por vezes incorretamente designado de sistema de navegação, utilizado para determinação da posição de um receptor na superfície da Terra ou em órbita.

O sistema GPS foi criado e é controlado pelo Departamento de Defesa dos Estados Unidos da América (*Department of Defense* - DoD), e pode ser utilizado por qualquer pessoa, gratuitamente, necessitando apenas de um receptor que capte o sinal emitido pelos satélites. O DoD fornece dois tipos de serviços GPS: O serviço padrão (*Standard Positioning Service* - SPS) e o serviço de precisão (*Precise Positioning Service* - PPS). Contrariamente ao que inicialmente acontecia, atualmente os dois serviços estão disponíveis em regime aberto em qualquer parte do mundo. O sistema GPS está dividido em três partes: espacial, de controle e do usuário. O segmento espacial é composto pela constelação de satélites. O segmento de controle é formado pelas estações terrestres dispersas pelo mundo ao longo da Zona Equatorial, responsáveis pela monitorização das órbitas dos satélites, sincronização dos relógios atômicos de bordo dos satélites e atualização dos dados de almanaque que os satélites transmitem. O segmento do usuário consiste num receptor que capta os sinais emitidos pelos satélites. Um receptor GPS (GPSR) decodifica as transmissões do sinal (código e fase) de múltiplos satélites e calcula a sua posição com base nas distâncias a estes. A posição é dada por latitude, longitude e altitude, coordenadas geodésicas referentes ao sistema WGS84⁵. O Sistema de coordenadas WGS84 é um sistema de referência terrestre convencional (*Conventional Terrestrial Reference System* - CTRS), o que quer dizer que as observações efetuadas estão reduzidas a uma mesma época, eliminando-se o efeito do movimento do pólo nas coordenadas determinadas. O sistema de coordenadas WGS84 é 3D, comum e ortogonal

⁵WGS84 (*World Geodetic System*) é um *datum* de referência para a utilização de Sistemas de Posicionamento Global - (GPS)

(HOFFMANN-WELLENHOF; COLLINS, 1994).

A.9.1 Descrição técnica

O sistema consiste numa “constelação” de pelo menos 24 satélites em 6 planos orbitais. Os primeiros satélites GPS foram lançados entre Fevereiro de 1978 (Bloco I), e 6 de Novembro de 2004 (o 29^o). Cada um circunda a Terra duas vezes por dia a uma altitude de aproximadamente 20.200 quilômetros (12.600 milhas náuticas). Os satélites têm a bordo relógios atômicos e constantemente difundem o tempo preciso de acordo com o seu próprio relógio, junto com informações adicionais, tais como os elementos orbitais de movimento.

O receptor não necessita ter um relógio de tão grande precisão, mas sim que seja suficientemente estável. O receptor capta os sinais de quatro satélites para determinar as suas próprias coordenadas, e ainda o tempo. Então, o receptor calcula a distância a cada um dos quatro satélites pelo intervalo de tempo entre o instante local e o instante em que os sinais foram enviados (esta distância é chamada pseudodistância). Decodificando as localizações dos satélites a partir dos sinais de rádio e de uma base de dados interna, e sabendo a velocidade de propagação do sinal, o receptor pode situar-se na intersecção de quatro esferas, uma para cada satélite (PARKINSON; SPILKER, 1996).

A.9.1.1 Trilateração do Satélite

Coordenadas exatas podem ser calculadas para qualquer posição na Terra medindo a distância a partir de um grupo de satélites até a posição a ser definida. Os satélites agem como pontos de referência precisos. Supondo que a distância a partir de um satélite seja conhecida, a posição pode ser restringida à superfície de uma esfera, que circunda aquele satélite. Se a distância a partir de um segundo satélite também é conhecida, isto limita a posição à intersecção de duas esferas. Adicione um terceiro satélite e a posição limita-se a um dos dois pontos. Desconsidera-se uma dessas posições, porque é uma resposta pouco provável, esta posição está no espaço ou movimentando-se em alta velocidade. Conhecendo-se a distância entre a posição e os três satélites, as coordenadas podem ser calculadas. Na prática, é necessário um quarto satélite para resolver as quatro incógnitas, X, Y, Z e tempo.

A.9.1.2 Alcance do Satélite

A distância de um único satélite é estabelecida medindo o tempo de transmissão dos sinais de rádio dos satélites ao receptor. Para medir o tempo de transmissão de sinais de rádio, o receptor precisa saber quando o sinal deixou o satélite. Para estabelecer quando um sinal deixou um satélite, o mesmo código Pseudo-Aleatório é criado ao mesmo tempo em ambos. Cada satélite GPS transmite dois sinais de rádio, $L1$ a 1575,42 MHz e $L2$ a 1227,60 MHz. O sinal $L1$ é modulado por dois códigos variáveis de ruídos pseudo-aleatórios, códigos P e C/A . A *precisão*, ou código P (*Precise*), pode transformar-se num código secreto para uso militar. O código *geral*, ou código C/A (*Coarse Acquisition*), não se converte em código secreto. O sinal $L2$ é modulado somente pelo código P . A maioria dos receptores civis utilizam o código C/A para obter informações do sistema GPS. Alguns receptores de grau topográfico utilizam o código P . O receptor examina o código que chega do satélite e então verifica o tempo que levou para criar tal código. Esta diferença de tempo multiplicada pela velocidade da luz (≈ 300.000 km/s) fornece a distância.

A.9.1.3 Precisão de tempo

Os cálculos dependem de relógios altamente precisos. O código tem que ser criado em ambos, receptor e satélite, exatamente ao mesmo tempo. Os satélites têm relógios atômicos com precisão de nanosegundo, mas são muito caros para se colocar em cada receptor de terra. Os receptores usam uma medição a partir de um quarto satélite para remover qualquer erro de relógio. Essa medição pode ser usada para remover erros que ocorrem se os relógios do satélite e do receptor não estão perfeitamente sincronizados.

A.9.1.4 Posicionamento do Satélite

Existem 24 satélites NAVSTAR⁶ em funcionamento orbitando a Terra a cada doze horas, a uma altitude de cerca de 20.200 km. Os satélites ocupam quatro órbitas em cada um dos seis planos inclinados 55° ao Equador. Os mesmos estão tão altos que há pouca resistência atmosférica e suas órbitas são muito estáveis. São constantemente monitorados pelo Departamento de Defesa (DoD). Cada satélite contém vários relógios atômicos de alta precisão e constantemente transmite sinais de rádio usando seu próprio código de identificação. O DoD tem quatro estações monitoras com base na Terra, três estações de transferência, e

⁶O sistema GPS foi concebido pelo Departamento de Defesa dos EUA no início da década de 1960, sob o nome de *projeto NAVSTAR*.

uma estação de controle central. As estações monitoras rastreiam os satélites continuamente e fornecem dados para a estação de controle central. A estação de controle central calcula os caminhos dos satélites e coeficientes de correção do relógio e envia-os para uma estação de transferência. As estações de transferência transmitem os dados para cada satélite pelo menos uma vez por dia.

A.9.1.5 Correção de erros

Algumas fontes de erros no GPS são difíceis de se eliminar. Os cálculos pressupõe que os sinais GPS são emitidos em uma velocidade constante, a velocidade da luz. Infelizmente, a velocidade da luz só é constante no vácuo. Uma vez que o sinal GPS entra na ionosfera (região altamente ionizada da atmosfera terrestre, e que tem aproximadamente de 40 km a 700 km de altitude) e a troposfera (camada atmosférica que vai da superfície até uma altitude média de 10 km), o sinal diminui de velocidade, resultando em cálculos de distância incorretos. Outra fonte significativa de erro é a disponibilidade seletiva (D/S). disponibilidade seletiva é uma degradação artificial do sinal do satélite pelo Departamento de Defesa dos Estados Unidos. Ela causa erro em uma posição GPS de mais de 100 metros. a D/S pode ser removida usando uma técnica chamada correção diferencial, que aumenta sobremaneira a precisão dos dados GPS coletados. Esta técnica envolve o uso de um receptor e um determinado lugar conhecido, a estação base e a coleta de posições GPS em lugares desconhecidos com outros receptores, itinerantes (*rovers*) e remotos. Os dados coletados no lugar conhecido são usados para determinar que erros estão contidos nos dados do satélite. A informação da Estação Base é então aplicada aos dados coletados pelos *rovers* e as diferenças de desvio são usadas para remover erros das posições dos receptores itinerantes. Deve-se conhecer a estação base muito bem pois a precisão da posição de correção diferencial depende da precisão das coordenadas da estação base.

A.9.1.6 Precisão dos serviços GPS

No serviço padrão (SPS) a precisão dos serviços GPS é (UNITED STATES COAST GUARD NAVIGATION CENTER, 1995):

- de ≈ 100 m na longitude;
- de ≈ 156 m na latitude;
- de ≈ 340 ns de relógio.

No serviço de precisão (PPS), a precisão do fornecimento dos serviços GPS é:

- de ≈ 22 m na longitude;
- de $\approx 27,5$ m na latitude;
- de ≈ 200 ns de relógio.

A.9.2 Aplicação do GPS na robótica móvel

Embora o GPS seja um sistema de posicionamento eficiente e atrativo, infelizmente não funciona a contento em vários ambientes de interesse para a robótica móvel. Na realidade, os sistemas de posicionamentos baseados na tecnologia GPS têm que estar em contato permanente com os satélites, para que seja assegurada uma leitura o mais precisa possível por parte do receptor. Nos ambientes fechados onde os sinais de rádio emitidos pelos satélites têm alcance limitado (ou nulo), o posicionamento baseado em GPS fica comprometido. Em ambientes tais como, densas florestas, entre grandes construções como nos centros urbanos, no subsolo, em ambientes subaquáticos, dentre outros, a precisão do sistema GPS fica aquém da necessária.

No contexto dos sistemas robóticos móveis, uma combinação do GPS para localização de aspecto amplo, porém mais grosseira e um método alternativo baseado em sensores para um posicionamento preciso, porém de alcance local, pode ser atrativa. O GPS então, pode servir para se determinar, ou verificar, se o robô está em determinada região do espaço, e então, métodos específicos, como posicionamento utilizando-se pontos de referências podem ser ativados.

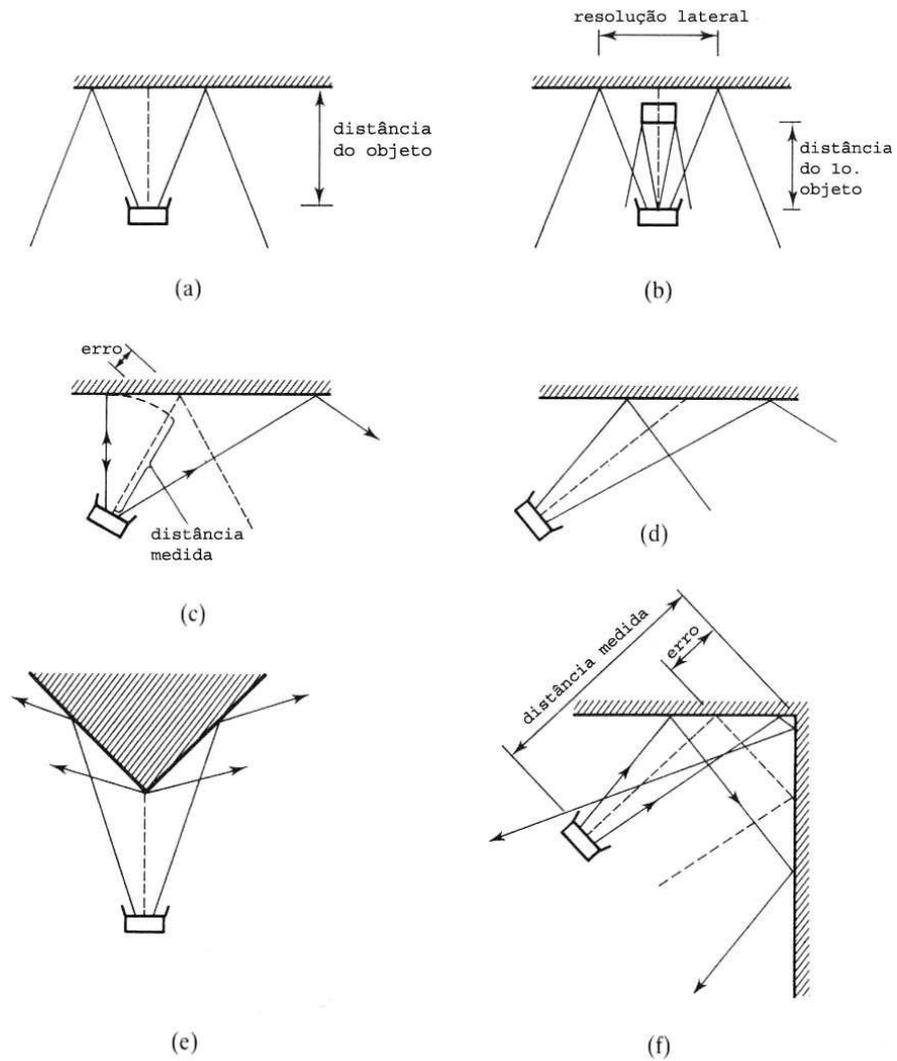
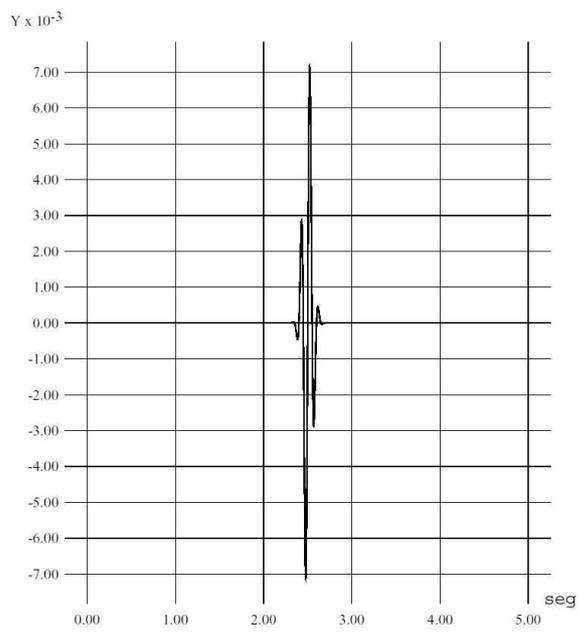
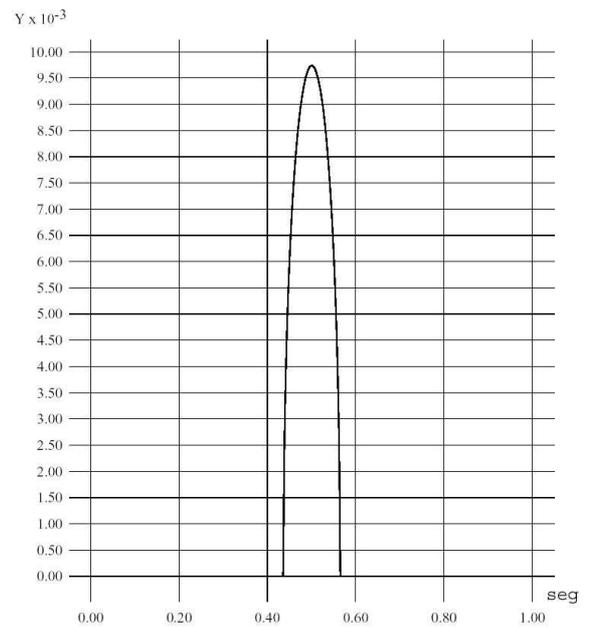


Figura A.10: (a) Sonar obtendo medições precisas. (b) Eco de um pequeno objeto próximo ao sonar. (c) A resolução lateral não é muito precisa. (d) Reflexões especulares causa o desaparecimento da parede. (e) Cantos abertos produzem fracas frentes de ondas esféricas. (f) Cantos fechados tem medições do próprio canto devido as múltiplas reflexões.

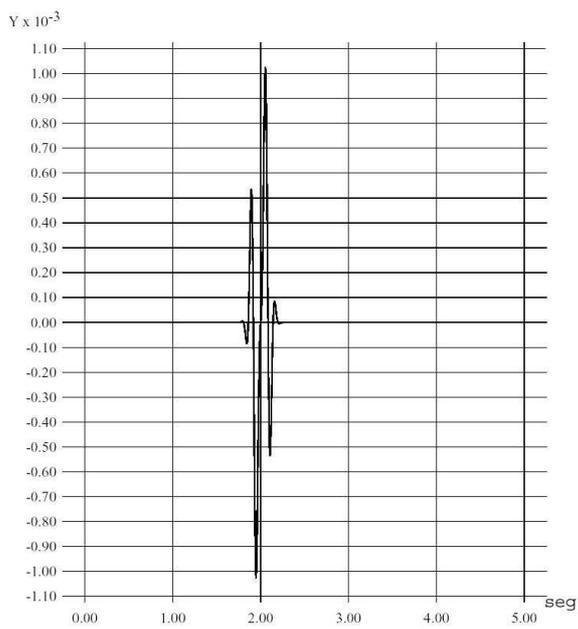


(a) Pulso do sonar

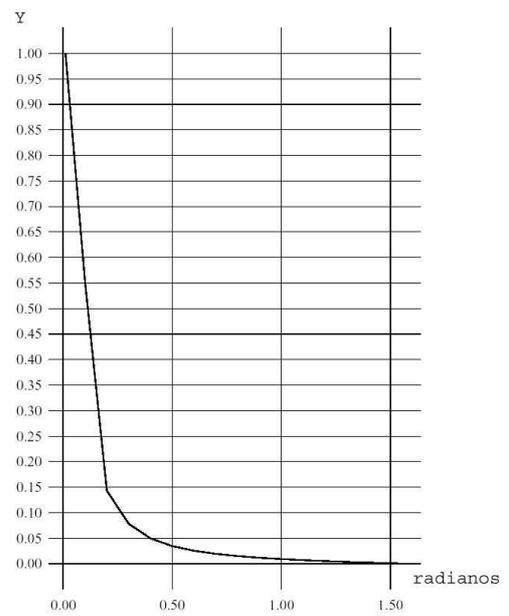


(b) Função de propagação pontual

Figura A.11: Pulso do sonar e a função de propagação pontual.



(a) Resultado da convolução do pulso do sonar



(b) Atenuação do sinal do sonar

Figura A.12: Convolução da resposta de impulso do transdutor e a atenuação do sinal em função do ângulo do transdutor.

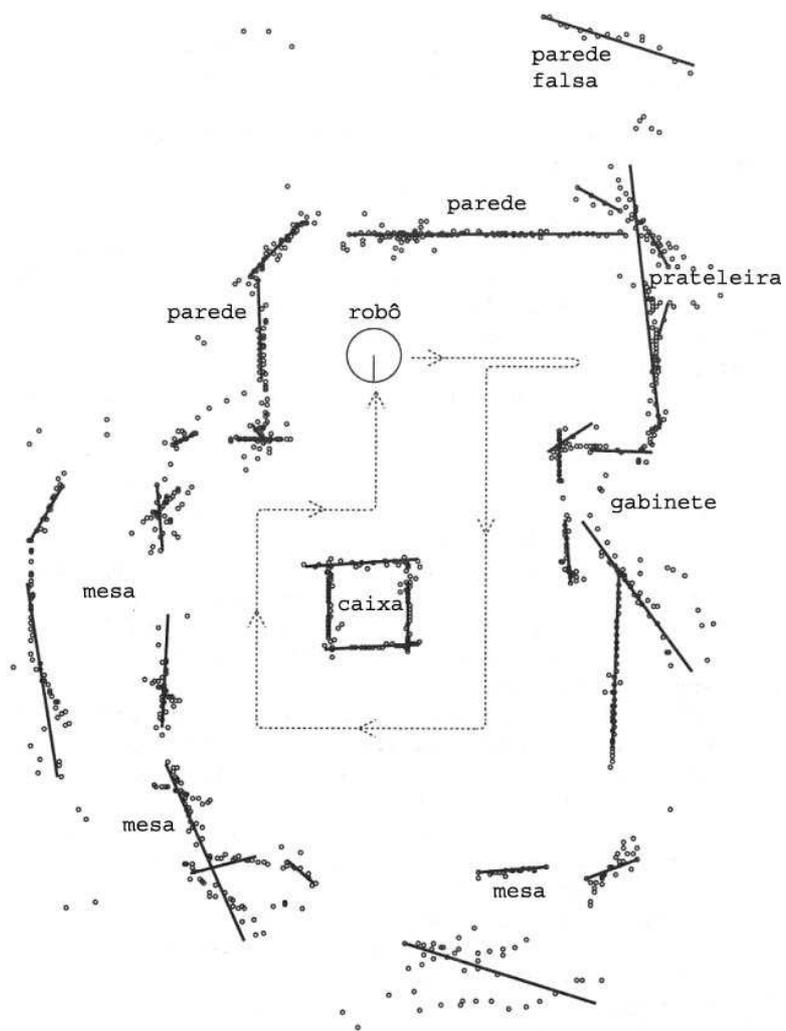


Figura A.13: Exemplo das medidas do sonar feitas pelo robô móvel.

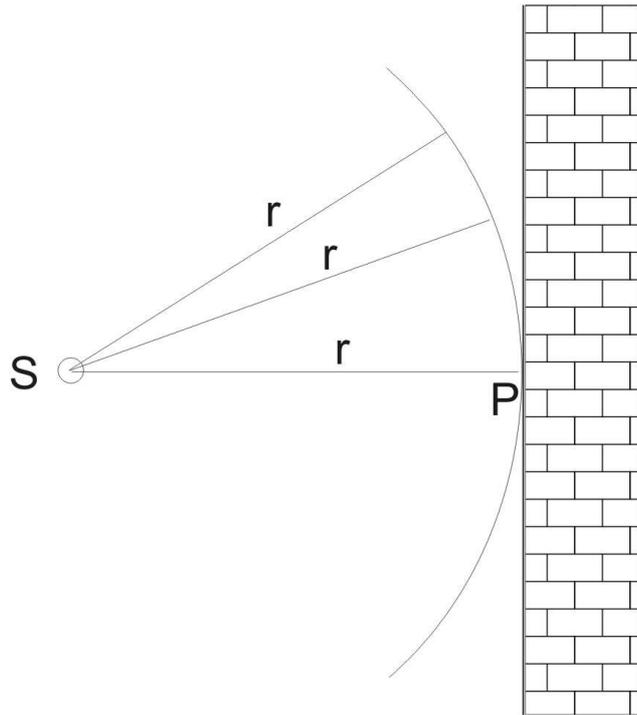


Figura A.14: Exemplo da região de distância constante (RDC) vista pelo sonar.

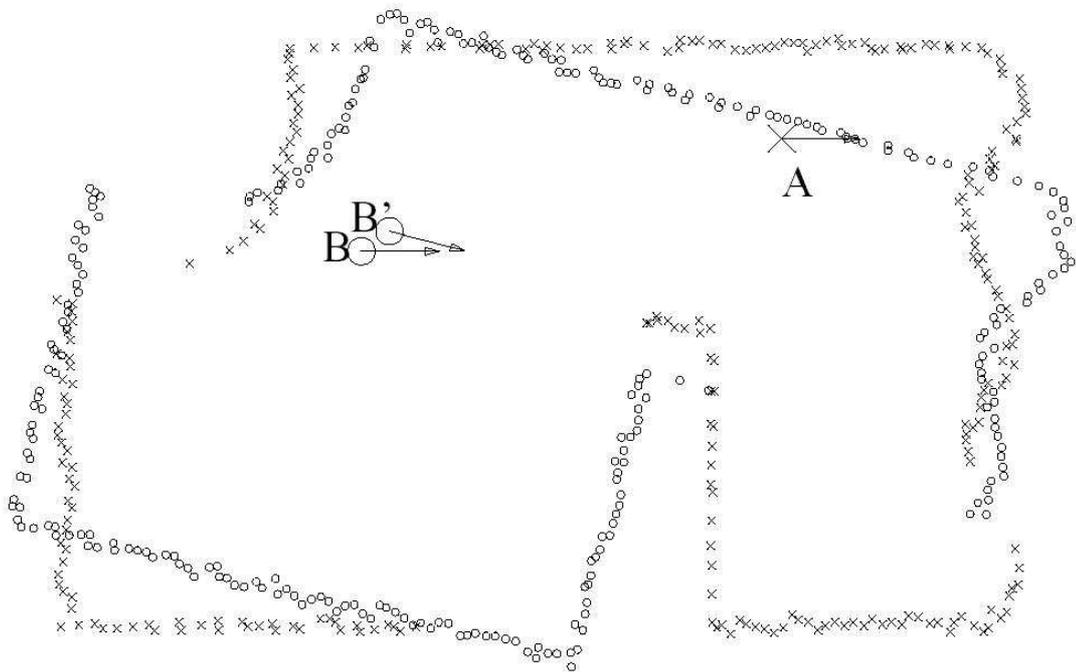


Figura A.15: Erro de alinhamento nas varreduras a laser feitas em B' e em B .

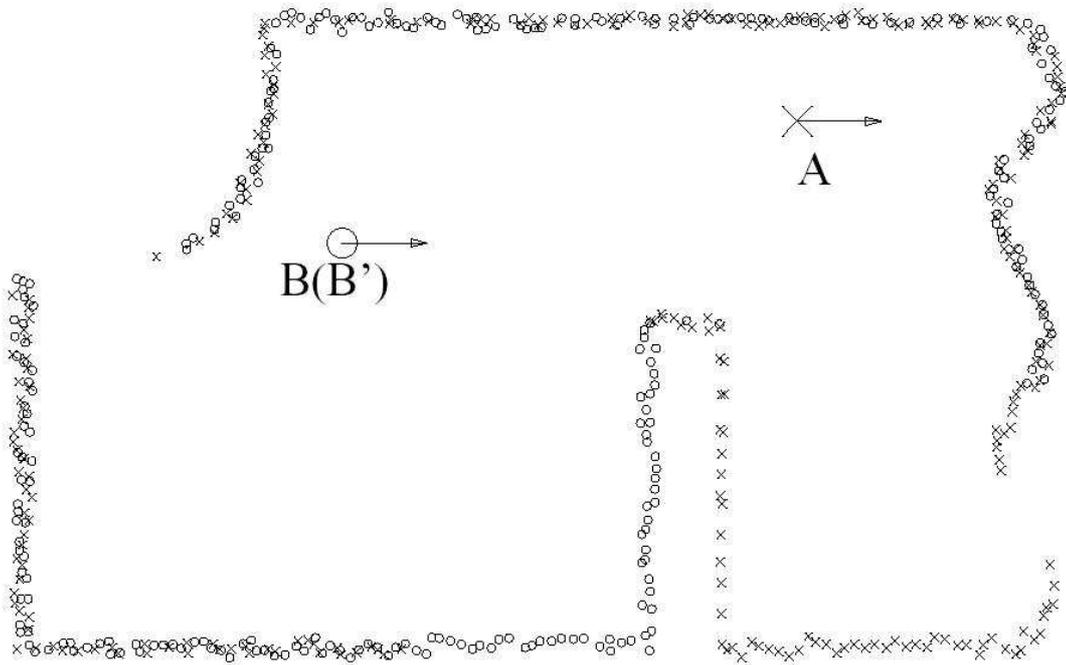


Figura A.16: Resultado do alinhamento das duas varreduras dos sensores laser.

Apêndice B – Sistemas de navegação

A navegação de robôs móveis autônomos está relacionada às técnicas que fornecem os meios para que um robô autônomo se mova de forma segura de um local a outro do ambiente. Encontrar um caminho de uma determinada posição até um destino é um dos problemas fundamentais da robótica móvel autônoma. Um algoritmo de planejamento de trajetória deve garantir um caminho até o destino, ou indicar se o destino é inacessível.

Os métodos tradicionais de planejamento de trajetória (LOZANO-PEREZ; WESLEY, 1979) assumem que o conhecimento sobre o ambiente é completo e perfeito. Utilizando modelos de planta, um caminho livre de colisões é planejado e uma trajetória é traçada. O sistema de controle motor é então acionado para que o robô siga a trajetória da melhor maneira possível.

Numa situação mais realística o ambiente pode ser alterado com o passar do tempo. Por exemplo, um robô móvel autônomo que atua em um ambiente onde pessoas circulam deve ser capaz de lidar com a troca de posição de certos objetos (cadeiras, caixas, etc.). O robô deve se basear em seus sensores para perceber o ambiente e efetuar o planejamento de acordo com a nova topologia encontrada. Uma abordagem comum é utilizar os sensores para criar um mapa do ambiente (mapeamento do ambiente). Uma vez que o ambiente esteja mapeado, as técnicas de planejamento tradicionais podem ser aplicadas. Reconstruir um modelo de planta baseando-se em informações sensoriais não é uma tarefa simples pois os dados geralmente tem um nível de incerteza muito alto (LEONARD; DURRANT-WHYTE; COX, 1990).

No sentido de se introduzir uma revisão bibliográfica a respeito do assunto, é apresentado na seção B.1 algumas das principais técnicas de medição de distância utilizadas pelos robôs móveis. Pode ser visto na seção B.2 uma introdução a respeito das técnicas de planejamento de trajetória. A seção B.3 descreve alguns das mais relevantes técnicas de posicionamento¹ e localização dos robôs móveis. Na seção B.4 é apresentado uma abordagem analítica da fusão de dados sensoriais dos robôs móveis, onde pode ser visto na seção B.4.4 uma introdução ao

¹Do inglês: *pose*

filtro de Kalman, ferramenta computacional utilizada para estimativa de localização de robôs móveis autônomos. Um ensaio de simulação utilizando-se o filtro de Kalman para estimativa do posicionamento de robôs móveis é apresentado na seção B.4.6. Aspectos relacionados aos mapas utilizados nos sistemas robóticos móveis é apresentado de uma maneira sucinta na seção B.5.

B.1 Técnicas de medição de distância

Um dos fatores cruciais para a navegação autônoma dos robôs móveis é a correta localização do mesmo no ambiente de trabalho. Para isto, várias metodologias utilizam a distância entre o robô e os marcadores para o cálculo da triangulação. Existem, basicamente, três tipos de medição de distância empregada nos robôs móveis (BORENSTEIN et al., 1998):

- Medições no deslocamento de fase, cuja técnica envolve a transmissão contínua de um sinal periódico e a comparação com o sinal refletido.
- Medições feitas no tempo de viagem (*ToF*) de um pulso de RF, laser ou ultrasônico emitido pelo robô móvel com uma certa energia, que viaja até um certo obstáculo e é refletido de volta ao robô.
- Medições baseadas em radares e lasers que utilizam frequência modulada (FM). Esta técnica é de alguma forma relacionada com técnica de medição do deslocamento de fase.

B.1.1 Medições no deslocamento de fase

O cálculo da distância do robô móvel até um certo obstáculo utilizando-se da técnica de medições no deslocamento de fase envolve a transmissão contínua de uma onda periódica. Uma irradiação de laser com amplitude modulada, RF ou energia acústica é direcionada ao alvo. Uma pequena porção desta onda é refletida pela superfície de volta ao detector ao longo de um caminho direto (CHEN; NI; WU, 1993). A energia retornada é comparada a uma referência gerada simultaneamente que é retirada do sinal original, e o deslocamento de fase relativo entre os dois sinais é medido, como ilustrado na Figura B.1, para o cálculo da distância entre o robô e o anteparo. Para sinais de alta frequência de RF, ou sistemas baseados em laser, a detecção é usualmente precedida pelo heteródino do sinal de referência com o sinal recebido através de frequências intermediárias, enquanto é preservado o deslocamento de fase

relativo, para permitir que o detector de fase opere com freqüências mais baixas (VUYLSTEKE; PRICE; OOSTERLINCK, 1990)

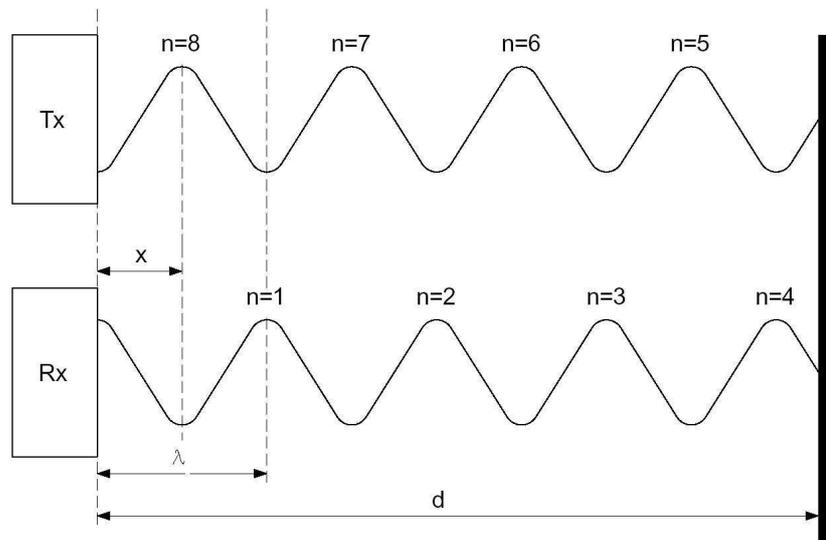


Figura B.1: Relação entre a onda gerada e a refletida, onde x é a medida correspondente à diferença de fase ϕ .

O deslocamento de fase relativo expresso como uma função da distância da superfície refletora do alvo é (WOODBURY; BRUBACHER; WOODBURY, 1993):

$$\phi = \frac{4\pi d}{\lambda}, \quad (\text{B.1})$$

onde ϕ é o deslocamento de fase, d a distância do robô móvel até o alvo e λ o comprimento de onda do sinal modulado. A distância desejada até o alvo (d) como função do deslocamento de fase medido (ϕ) é dado por

$$d = \frac{\phi \lambda}{4\pi} = \frac{\phi}{4\pi f}, \quad (\text{B.2})$$

onde f é a freqüência de modulação da onda emitida. Para modulações do sinal utilizando-se ondas quadradas com freqüência relativamente baixa, típica dos sistemas de ultra som (de 20 kHz até 200 kHz), a diferença de fase entre os sinais transmitidos e refletidos pode ser medida através de um circuito simples, como ilustrado na Figura B.2 (FIGUEROA; BARBIERI, 1991). Para freqüências baixas, típicas de sistemas ultrasônicos, o circuito de detecção de fase simples baseado numa porta lógica OU EXCLUSIVO gera uma tensão analógica (V) que é proporcional à diferença de fase entre o sinal transmitido (V_1) e o recebido (V_2). O saída da

porta lógica OU EXCLUSIVO tem nível lógico alto sempre que suas entradas tiverem níveis lógicos opostos, gerando uma tensão através do capacitor C que é proporcional à diferença de fase. Por exemplo, quando os dois sinais da entrada estiverem em fase (i.e., $\phi = 0$), a saída da porta permanece com nível lógico baixo e $V = 0$. A tensão máxima na saída ocorre quando $\phi = 180^\circ$. Embora o sistema seja simples de ser implementado, sua utilização é limitada à baixas frequências e requer freqüentes calibrações para compensar as flutuações e as tensões residuais (*offsets*) que são geradas pelas variações das condições ambientais.

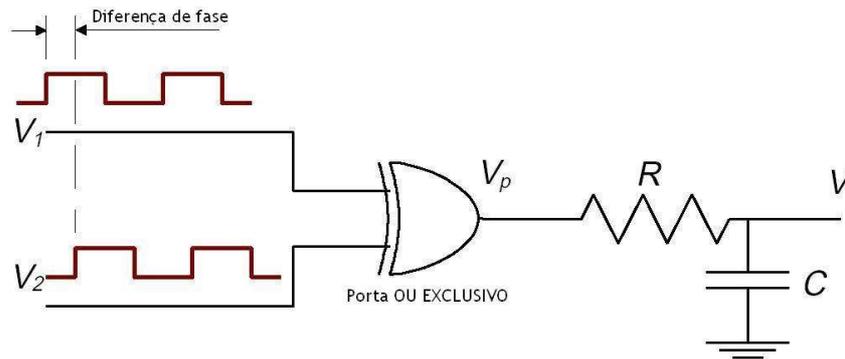


Figura B.2: Circuito simples para detecção da diferença de fase entre o sinal transmitido (V_1) e o recebido (V_2).

Na utilização de altas frequências, o deslocamento de fase entre o sinal senoidal transmitido e o refletido pode ser medido multiplicando-se os dois sinais através de um misturador (*mixer*) eletrônico, e então tirando-se a média do produto de muitos ciclos de modulação (WOODBURY; BRUBACHER; WOODBURY, 1993). Este processo de integração pode consumir um tempo relativamente longo, tornando-se difícil de se trabalhar com taxas extremamente rápidas de atualizações. O resultado pode ser matematicamente expresso como:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \sin\left(\frac{2\pi c}{\lambda}t + \frac{4\pi d}{\lambda}\right) \sin\left(\frac{2\pi c}{\lambda}\right) dt, \quad (\text{B.3})$$

que se reduz à

$$A \cos \frac{4\pi d}{\lambda} = A \cos \phi, \quad (\text{B.4})$$

onde t é o tempo, T é o intervalo de amostragem e A é o fator de amplitude do ganho do amplificador integrador.

Pode ser visto, pela equação B.4, que a quantidade realmente medida é, de fato, o cosseno do deslocamento da fase e não a fase em si. Esta situação introduz um intervalo de ambigüi-

dade para cenários onde a distância de viagem do sinal excedem o comprimento de onda modulado (i.e., a medida da fase se torna ambíguo uma vez que ϕ exceda 360°). Conrad e Sampson (1990) definem o intervalo de ambigüidade R_a como o alcance máximo que permite que a diferença de fase percorra um ciclo completo de 360 graus:

$$R_a = \frac{c}{2f}, \quad (\text{B.5})$$

onde f é a frequência de modulação e c a velocidade da luz.

Como ilustrado na Figura B.1, o trajeto total do sinal é dado por $2d$ que é igual a algum número inteiro de comprimentos de onda $n\lambda$ mais uma fração do comprimento de onda x associado com o deslocamento de fase. Uma vez que a relação do cosseno não é um valor único para todo ϕ , existirá mais do que uma distância d correspondente a uma dada medição de deslocamento de fase:

$$\cos \phi = \cos \left(\frac{4\pi d}{\lambda} \right) = \cos \left[\frac{2\pi(x + n\lambda)}{\lambda} \right], \quad (\text{B.6})$$

onde $d = \frac{(x+n\lambda)}{2}$ é a distância verdadeira até o alvo, x é a distância correspondente à diferença de fase ϕ e n o número de períodos completos do sinal modulado.

O potencial para informações erradas como resultado do intervalo de ambigüidade reduz o interesse em sistemas de detecção de fase. Algumas aplicações evitam estes problemas arranjando o caminho ótico de tal forma que o máximo alcance possível esteja dentro do intervalo de ambigüidade. Alternativamente, medições sucessivas do mesmo alvo utilizando duas modulações de frequências diferentes podem ser feitas, resultando em duas equações, permitindo que ambos, x e n sejam unicamente determinados. Kerr (1988) descreve uma implementação, nestes moldes, utilizando um sinal com frequência modulada em 6 MHz e 32 MHz.

Vantagens de sistemas com emissão contínua de onda sobre métodos que utilizam pulsos para o cálculo do tempo de percurso, incluem a habilidade da medição da direção e velocidade de um alvo móvel. Em 1842, um austríaco de nome Johann Doppler publicou um artigo descrevendo o que passou a ser conhecido como *efeito Doppler*, onde a frequência de uma onda, com uma certa energia, refletida em um objeto em movimento é uma função da velocidade relativa entre o objeto e o observador.

B.1.2 Medições no *ToF* de um sinal

Medições feitas no tempo de viagem (*ToF*) de um pulso de RF, laser ou ultrasônico emitido pelo robô móvel com uma certa energia, que viaja até um certo obstáculo e é refletido de volta ao robô são largamente utilizadas nos sistemas robóticos móveis nos sistemas de navegação.

O Apêndice A.6 ilustra técnicas de medições robô-anteparo feitas com o Sonar. O Apêndice A.6.2 ilustra a utilização do Radar nos sistemas de medição de distância.

B.1.3 Medições baseadas em radares e lasers

Também são muito utilizados na robótica móvel técnicas de medições baseadas em radares e lasers que utilizam frequência modulada (FM). Esta técnica é de alguma forma relacionada com técnica de medição do deslocamento de fase. O Apêndice A.7 ilustra a utilização do Radar nos sistemas de medição de distância. O Apêndice A.8 descreve os sistemas robóticos móveis que utilizam o Laser nos sistemas de navegação.

B.2 Planejamento de trajetória

O planejamento da locomoção, sob vários tipos de restrições, é um campo muito vasto de pesquisas na área da robótica móvel (GRAF; WANDOSELL; SCHAEFFER, 2001). Com o intuito de subsídio para este trabalho, apresenta-se a seguir alguns aspectos mas importantes deste assunto. O planejamento básico de trajetória para os robôs móveis refere-se a determinação de um caminho no espaço-C (*espaço de configuração*) entre uma configuração inicial do robô e uma configuração final, de tal forma que o robô não colida com nenhum obstáculo no ambiente, e que a movimentação planejada seja consistente com as restrições cinemáticas do veículo.

B.2.1 Espaço de configuração

Os robôs móveis apresentam grande variedade de formatos, tamanhos e capacidades. No caso mais geral, um robô móvel pode ser representado através de um espaço-C, mas para muitos veículos autônomos uma representação mais simplificada é o suficiente.

O espaço-C é a chave da construção e do formalismo para o planejamento de movimento

dos sistemas robóticos móveis. Uma configuração q do robô A é uma especificação do estado físico do robô A com respeito a um quadro fixo F_W com relação ao meio (LATOMBE, 1991). Considerando-se um robô rígido A capaz de realizar rotação e translação no plano. A configuração de A pode ser representada por $\mathbf{p} = [x \ y \ \theta]$, onde \mathbf{p} define completamente a configuração do robô. Para robôs mais complexos, como por exemplos os robôs com pernas ou robôs articulados, a estrutura de \mathbf{p} pode ser muito mais complexa. A configuração de um robô com pernas deve incluir um posicionamento rígido para algumas origens associadas com o robô, aumentada com os ângulos das juntas de cada uma das pernas. O espaço de configuração de A é o espaço \mathcal{C} de todas as possibilidades de configurações de A . O espaço \mathcal{C} define todas as configurações válidas do robô em seu ambiente.

A construção física do robô pode proibir certas configurações, assim como a presença de obstáculos no ambiente. Por exemplo, um robô circular com raio r não pode se mover com distâncias menores que r de um objeto, isto gera restrições nas possibilidades de configurações do robô e reduz o tamanho da porção acessível do espaço de configuração do robô. Restrições desta forma podem ser escrita como

$$G(\mathbf{p}) = 0, \quad (\text{B.7})$$

onde \mathbf{p} é o posicionamento do robô, e são conhecidas como restrições holonômicas. Um obstáculo pode ser transformado em um conjunto de pontos no espaço de configuração que uma relação na forma da equação B.7. A região do espaço inacessível é chamada de *obstáculo de espaço de configuração*. Em geral, o planejamento da trajetória para se evitar obstáculos do espaço- \mathcal{C} na presença de restrições holonômicas é direto. A parte do ambiente que é acessível ao robô, ou seja, o espaço livre, é representado por uma região análoga do espaço de configuração denotada por \mathcal{C}_{livre} . Restrições nas *derivadas* do movimento do robô que não podem ser integradas, i.e., reduzidas a restrições holonômicas, são conhecidas como *restrições não holonômicas* (LATOMBE, 1991), conforme pode ser visto na seção 2.3.2. A equação B.7 então toma a forma de

$$G\left(\mathbf{p}, \frac{d\mathbf{p}}{dt}, \frac{d^2\mathbf{p}}{dt^2}, \dots\right) = 0. \quad (\text{B.8})$$

Restrições não holonômicas incluem limites nas velocidades (tangentes no espaço de configuração) permitidas, reduzem a extensão dos movimentos diferenciais permitidos e vêm a complicar de uma maneira significativa o problema de planejamento de deslocação. A essência do problema causado pelas restrições não holonômicas é a seguinte: para um robô se locomover de um estado admissível para outro, mesmo que estes estados sejam adjacentes,

uma trajetória com complexidade arbitrária pode ser requerida. Exemplos comuns de veículos com restrições não holonômicas de movimentos são automóveis, veículos com reboques e, no nosso caso, os robôs móveis com tração diferencial, conforme é abordado na seções 2.2. O estacionamento em vagas paralelas é uma ilustração familiar do tipo de dificuldade associada com o problema de planejamento de trajetória. Uma restrição padrão para os robôs móveis é a limitação no *raio de curvatura* que pode ser executada na trajetória do robô. Para problemas que envolvem limites para o raio de curvatura, trajetórias com comprimento ótimo em ambientes livres de obstáculos são compostas de seqüências de arcos circulares.

Para tornar a relação entre restrições holonômicas e não holonômicas mais concreta, considera-se a representação de espaço de configuração de um robô com tração síncrona A conforme apresenta-se na seção 2.4. Estes sistemas robóticos podem controlar a velocidade de rotação das rodas e a direção para onde devem apontar. A representação de espaço de configuração do robô A pode ser dada por $\mathbf{p} = [x \ y \ \theta]$. Supondo que o raio do robô seja r e que exista um obstáculo infinitesimalmente pequeno na origem. Então, este obstáculo resulta numa restrição holonômica do robô na forma

$$x^2 + y^2 > r^2. \quad (\text{B.9})$$

Se o robô se move na velocidade v , então

$$\begin{aligned} \frac{dx}{dt} &= v \cos(\theta) \\ \frac{dy}{dt} &= v \sin(\theta) \end{aligned} \quad (\text{B.10})$$

e

$$dx \sin(\theta) - dy \cos(\theta) = 0; \quad (\text{B.11})$$

isto é, o robô se move em linha reta na direção em que estiver apontando. Isto é uma restrição holonômica da forma dada na equação B.8 porque envolve \mathbf{p} e \mathbf{p}' , e \mathbf{p}' não pode ser eliminado. Planejamento de trajetória para robôs que incorrem em restrições holonômicas é conhecido como *planejamento de trajetória holonômica*, o que é significativamente diferente do planejamento de trajetória para robôs não holonômicos (*planejamento de trajetória não holonômica*). Na presença de restrições não holonômicas, até problemas simples, como encontrar uma trajetória mínima através de um ambiente desordenado pode ser muito difícil (FIERRO; LEWIS, 1998).

Controlabilidade refere-se a disponibilidade de se mover entre pontos arbitrários no espaço de estado. Sistemas não holonômicos não são localmente controláveis, embora sejam,

na maioria dos casos, globalmente controláveis. Uma abordagem comum ao planejamento de trajetória não holonômica para robôs móveis pode ser dividida em métodos que realizam buscas completas (freqüentemente baseados em discretizações), métodos de refinamentos iterativos, e métodos que utilizam controladores específicos mas que não conduzem muito bem em situações de obstáculos (LATOMBE; BARRAQUAND, 1993).

B.2.1.1 Simplificações do espaço-C

O maior problema associado com a representação do robô no espaço-C é que a representação de espaço de configuração de obstáculos e o resultado da dimensão do espaço de busca, para trajetórias de locomoção, pode ser muito grande, assim como trabalhosa de se encontrar. Várias simplificações tem sido propostas no intuito de se reduzir a complexidade de representação do espaço-C e, desta forma, reduzir o custo associado com a representação de objetos e planejamento da trajetória.

A simplificação clássica é assumir que o robô pode ser representado como um ponto, sendo capaz de movimentos onidirecionais. O problema óbvio desta simplificação é que veículos autônomos não são pontos e muitos projetos de robôs móveis introduzem restrições não holonômicas. Se o robô for reduzido a um ponto, e subseqüentemente o processamento utilizar esta representação pontual para o planejamento de operações, tais como os caminhos a seguir, será importante o conhecimento do tamanho real e formato do robô, assim como qualquer restrições não holonômicas no processo de execução da trajetória. Um mecanismo para o tratamento de tamanhos não pontuais dos robôs móveis é assumir-se que o robô tem um seção circular, com um certo raio r , e então *dilatar-se* todos os obstáculos do ambiente por uma quantidade radial igual a r . A operação de dilatação, também conhecida como soma de Minkowski², pode ser computada pela união do formato dos objetos com um conjunto de círculos (ou esferas, em ambientes 3D) colocados em cada ponto do limite do objeto. Obstáculos poligonais podem ser descritos como um conjunto de linhas e arcos de círculos após a dilatação. Na Figura B.3 um objeto poligonal é dilatado pelo raio r do robô e as quinas arredondadas por uma aproximação polinomial. A dilatação dos objetos do ambiente pode, por sua vez, tornar o ambiente mais complexo.

²Em geometria, a soma de Minkowski de dois conjuntos A e B , no espaço euclideano, é o resultado da soma de cada elemento de A a cada elemento de B , ou seja

$$A + B = \{a + b \mid a \in A, b \in B\}. \quad (\text{B.12})$$

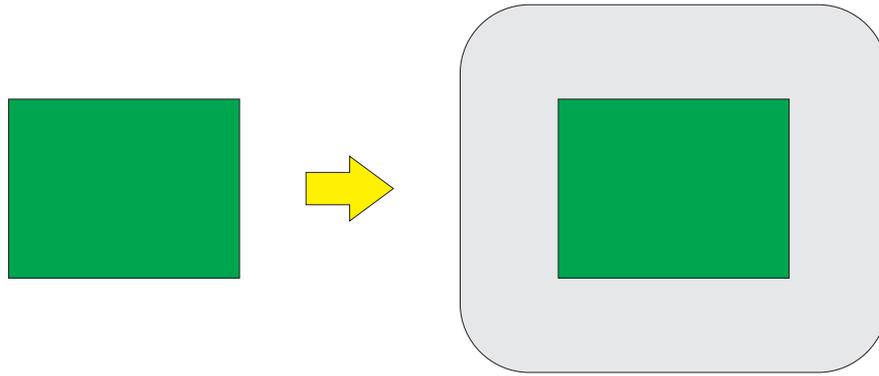


Figura B.3: Efeito da dilatação pelo raio r do robô.

B.2.2 Planejamento de trajetória nos robôs móveis não holonômicos

Conforme visto anteriormente, o planejamento básico de trajetória para os robôs móveis refere-se a determinação de um caminho no espaço- C entre uma configuração inicial do robô e uma configuração final, de tal forma que o robô não colida com nenhum obstáculo no ambiente, e que a movimentação planejada seja consistente com as restrições cinemáticas do veículo. Este problema básico pode ser discutido de várias maneiras, como a seguir:

- É sempre de interesse se considerar o caminho mais curto possível da trajetória a ser percorrida.
- Fórmulas alternativas de caminho de custo mínimo são também importantes. Em particular, o caminho de tempo mínimo não é necessariamente o caminho mais curto. Um exemplo desta situação acontece quando a velocidade máxima do robô é uma função da curvatura da trajetória, como é o caso de muitos robôs com tração síncrona. Como consequência, um caminho mínimo pode envolver velocidades mais baixas do que um caminho mais longo, mas com curvatura menor e consequentemente mais rápida, como caminho alternativo.
- Ambientes com obstáculos móveis pode apresentar desafios adicionais significantes.
- Algoritmos que podem operar sob restrições de tempo têm sido objeto de pesquisas recentes. Uma classe particular de interesse são os algoritmos de todo tempo (*any-time algorithms*) (ZILBERSTEIN, 1996). Eles apresentam soluções sempre que são interrompidos, mas apresentam solução que melhora a qualidade com o incremento do tempo computacional.

- Pode ser de interesse a seleção de trajetórias que satisfaçam outras restrições, assim como terminarem no ponto final (alvo). Por exemplo, a escolha de caminhos livres ou trajetórias que permitam que o robô localize certas marcas pré-estabelecidas no ambiente (NICHERSON et al., 1993).

O planejamento de locomoção considera a habilidade do robô e a estrutura do ambiente (e, desta maneira, os obstáculos). Um algoritmo considerado padrão para o planejamento de trajetória se apóia em um número de formalismos comuns e simplificações do ambiente, como a seguir:

- Um robô rígido A , que é freqüentemente modelado como um ponto.
- O ambiente W , que é estático e conhecido. O ambiente é também conhecido como o domínio ou espaço de trabalho do robô.
- Um conjunto de obstáculos conhecidos B_1, B_2, \dots, B_q em W .
- O robô trafega em segmentos de linha reta.

O problema geral de planejamento de trajetória é encontrar um caminho t , de forma que estando o robô num estado inicial A conduza-o para a posição final desejada (o alvo). Existem várias literaturas que versam sobre o planejamento de trajetória. Os algoritmos são construídos baseados em diferentes suposições teóricas e requisitos a respeito dos seguintes tópicos:

- O ambiente e o robô.** A estrutura do ambiente, as capacidades do robô, seu formato, etc.
- Solidez.** A garantia de que a trajetória planejada será livre de colisões.
- Plenitude.** A certeza de que o algoritmo garante o encontro de um caminho viável, se ele existir.
- Relação de custo.** A relação entre o custo do atual trajeto obtido pelo caminho ótimo.
- Espaço ou complexidade de tempo.** O espaço armazenado ou o tempo de computação gasto para se encontrar uma solução.

Para se tornar o problema de planejamento de trajetória factível é freqüentemente necessário se fazer uma variedade de simplificações com respeito ao ambiente real (KAMBHAMPATI; DAVIS, 1986). Após um algoritmo ter sido desenvolvido baseado num conjunto de suposições, ele deve realmente rodar num ambiente real. Algoritmos idealizados para o planejamento de trajetória devem ser capacitados para lidar com muitas realidades do mundo real que são difíceis de serem modeladas em laboratório, tais como obstáculos móveis, restrições dinâmicas, uma definição complexa do objetivo final do robô, critérios de otimização, incertezas, etc.

B.3 Sistemas de posicionamento dos robôs móveis

Para poder navegar de forma eficiente, um robô precisa ser capaz de determinar sua posição de forma rápida e precisa. Estimativas de posição relativamente precisas podem ser obtidas integrando as informações cinéticas do robô com os seus sensores de velocidade e direção (*encoders*). A localização é um componente chave em vários sistemas de sucesso para o controle de robôs móveis autônomos (BORENSTEIN et al., 1998) (KORTENKAMP, 1998). Encontra-se referências à localização como sendo “o problema fundamental para fornecer capacidades autônomas para um robô móvel” (COX, 1991). Para navegar de forma confiável em ambientes internos (*indoor*), um robô móvel necessita saber sua localização (posição e orientação) dentro deste ambiente. A partir das entradas sensoriais, o robô deve ser capaz de inferir sua posição e orientação relativa a um mapa global. A competência da localização não se restringe a ajudar na navegação, mas também é utilizada para a exploração e para a adaptação do mapa do ambiente pelo robô, bem como para seguir planos. Estimar a localização de um robô baseado em dados sensoriais é um dos problemas fundamentais da robótica móvel. Integrando no sistema de controle robótico um módulo localizador, amplia-se a capacidade de navegação e a robustez do robô móvel autônomo.

Desta maneira, para inúmeras tarefas um sistema robótico móvel precisa saber onde se encontra, para poder planejar caminhos apropriados ou para discernir se a localização atual é o local apropriado para a realização de uma certa operação. O conhecimento de “onde o robô está” tem diferentes conotações. No senso mais abrangente, o conhecimento de onde se encontra o robô envolve a estimação da localização do mesmo, tanto em termos qualitativos quanto quantitativos, com respeito a alguma representação global do espaço: pode-se referir a isto como *localização forte* (KUIPERS; BYUN, 1990). O problema da *localização fraca*, por outro lado, envolve o mero conhecimento de saber se a localização corrente já foi visitada

anteriormente, ou seja, se o robô já esteve neste ponto antes. Em certas situações, mapas qualitativos completos podem ser construídos somente de informações de localização fraca (YAMAUCHI; SCHULTZ; ADAMS, 1998)(DENG; MIRZAIAN, 1996). Desta forma, a localização fraca pode ser usada para a construção de mapas que pode ser utilizado subseqüentemente para localização forte.

Dada as estimativas aproximadas do posicionamento do robô baseadas na odometria e em métodos de predição baseados em inferência, e dado um mapa do ambiente e suficiente sensoriamento, é possível se manter um estimativa corrente da localização do robô com respeito ao mapa. Este processo é algumas vezes referido como *localização, estimação da localização* ou *posicionamento* (LU; MILIOS, 1997b)(THRUN, 2001). A especificação geral deste problema começa com uma estimativa inicial da localização do robô \mathbf{q} no espaço-C dada por uma distribuição de probabilidade $P(\mathbf{q})$. A localização baseada nos sensores fundamenta-se na premissa de que se utiliza os dados \mathbf{s} de um certo sensor em conjunto com o mapa para se produzir uma estimativa de posição refinada $P(\mathbf{q}|\mathbf{s})$, de tal forma que esta estimativa refinada tenha uma densidade de probabilidade incrementada pela posição verdadeira do robô.

Sob certas circunstâncias pode ser necessário a inferência da posição do robô sem uma estimativa anterior de sua localização. Este tipo de posicionamento é referido como *localização global* por analogia com a minimização de função global, por meio do qual um estado ótimo deve ser encontrado sem uma suposição inicial confiável. Uma versão mais comum do problema de localização do robô móvel é a necessidade de se refinar uma estimativa do posicionamento do robô continuamente. Esta tarefa é conhecida como *manutenção de posicionamento* ou *localização local* (SIM; DUDEK, 1999).

Talvez a abordagem mais simples para tarefa de manutenção de posicionamento é a utilização de controle em malha aberta ou odometria para se ficar informado das distâncias percorridas pelo robô em cada direção e então perfazer-se a soma destes deslocamentos para se produzir uma rede de deslocamento que pode ser somada a uma estimativa de posicionamento inicial. Ter o conhecimento a respeito do deslocamento pela observação de parâmetros internos do robô sem referência do mundo exterior é conhecido como *cálculo preditivo por inferência*³ (KLEEMAN, 1992)(YAMAUCHI, 1996). Esta é a técnica utilizada para se medir a distância percorrida por um automóvel usando o odômetro.

Essencialmente toda técnica utilizada para a estimativa de distância percorrida deve

³O termo em inglês observado na literatura especializada é *dead reckoning*, que é o método utilizado para se estimar a localização de navios ou aviões utilizando-se somente as medições internas, sem observações externas.

tratar de erros de medição. Esses erros podem surgir de ruídos elétricos, quantização do sinal, artefatos de digitalização do sinal, deslizamento da roda (no caso de odômetro), dentre outras fontes de erros. se o cálculo preditivo por inferência é utilizado sozinho para estimativa de posicionamento, estes erros são somados à estimativa de posicionamento absoluto e acumulado com as sucessivas locomoções do robô. Isto torna o problema geral de manutenção de um sistema de coordenada absoluta precisa muito difícil, ou até potencialmente impossível, na falta de alguma referência externa para se eliminar os erros acumulados. Localizações a longo prazo, e tarefa associadas, tais como navegação e construção de mapas, devem fazer referência ao mundo exterior para correção de posicionamento, se a precisão deve ser mantida. Em geral, isto envolve a utilização de dados de sensores para a recalibração da percepção do robô quanto ao seu posicionamento no ambiente. Em alguns casos, pode-se optar por simplesmente abandonar a metodologia do posicionamento absoluto em favor de estimativas de posicionamento local ou informações qualitativas (THRUN; BURGARD; FOX, 1998).

Alguns pontos chave no processo de se desempenhar a localização, tanto global como local, envolve combinar um conjunto de observações correntes a algum mapa preestabelecido. Métodos combinacionais padrão podem ser classificados, de uma forma geral, nas seguintes categorias:

- **Combinação dado-dado.** Combinação direta dos dados não processados com os dados prognosticados não processados extraídos do mapa, por modelamento preditivo ou utilizando-se de um conjunto de dados armazenados.
- **Combinação dado-modelo.** Combinação dos dados observados a modelos mais abstratos armazenados no mapa, baseado num modelo de como modelos e dados estão associados.
- **Combinação modelo-modelo.** Combinação de modelos armazenados no mapa com modelos gerados de observações correntes.

Cada uma destas técnicas têm sido utilizadas com algum sucesso, e cada uma tem seu domínio particular de aplicabilidade, dependendo particularmente de características do sensor e da metodologia de aquisição de dados, como pode ser visto em (BURGARD et al., 1996), (THRUN; BURGARD; FOX, 1998), (THRUN, 2001). Em geral, combinação de dados não processados pode reduzir dependência de suposições antecipadas sobre o ambiente mas tende a ser menos robusta, a menos que a técnica de combinação seja muito sofisticada.

B.3.1 Cálculo preditivo por inferência

Muitos dispositivos para medições de posição e distância são ferramentas de medições relativas, e.g., odômetros. Contando-se o número de rotações executadas pela roda de um veículo, por exemplo, e utilizando-se conhecimentos a respeito das dimensões da roda e da cinemática do veículo, uma estimativa da taxa de variação da posição do veículo pode ser obtida. Desta maneira, o cálculo das coordenadas absolutas envolvem a integração de quantidades diferenciais locais, e.g., mudanças na posição, orientação ou velocidade. Para se calcular a posição \mathbf{x} do robô num caso ideal livre de erros, com um vetor velocidade \mathbf{v} , tem-se

$$\mathbf{x} = \int_{t_0}^{t_f} \mathbf{v} dt = \int_{t_0}^{t_f} \frac{d\mathbf{x}}{dt} dt, \quad (\text{B.13})$$

onde o deslocamento ocorreu no intervalo de tempo t_0 até t_f . Mais geralmente, para informações de deslocamento de derivadas de alta ordem, como por exemplo a aceleração, pode-se integrar repetidamente para se conseguir a posição. Isto implica, contudo, que os erros nos sensores ou nos processos de integração são manifestados como polinômios de alta ordem do intervalo de tempo em que se está interessado. Para movimentos discretos, onde as mudanças na posição seja expressa por um vetor intervalo δ_i , pode-se calcular a posição absoluta como

$$\mathbf{x} = \sum \delta_i. \quad (\text{B.14})$$

Este método de estimação de posição é utilizado com freqüência em sistemas robóticos móveis, pela sua simplicidade e pelo baixo custo relativo. Ele tem precisão aceitável sob pequenos intervalos de amostragem, sendo dado terreno e mecanismo de tração apropriados. Desta maneira, a partir de uma posição inicial conhecida, pode-se calcular a posição final após um certo deslocamento ser concluído. Para percursos longos ou trajetórias complexas, contudo, os erros inevitáveis nas estimativas de posicionamento individual têm conseqüências maiores.

B.3.2 Medições com pontos de referência

As abordagens mais utilizadas para a estimação do posicionamento são baseadas nas soluções dos problemas geométricos e trigonométricos que envolvem restrições nos pontos de referências (ou marcadores) do ambiente. Em princípio, o problema está relacionado a estimativa de posicionamento dos pontos de referências com respeito a um sensor fixo. Variações importantes do problema surgem quando os pontos de referências não são rotulados,

i.e., suas identidades individuais são desconhecidas, quando são difíceis de se detectarem ou quando as medições não são precisas.

Fatores que caracterizam um sistema de estimativa de posicionamento particular incluem:

- Os pontos de referência são passivos ou ativos, i.e., são emissores de energia, tal como emissores de radio frequência?
- Qual é a modalidade do sensor, e.g., visão, laser, sonar, etc.?
- Quais são as propriedades geométricas dos pontos de referências? São grandes, pequenos, pontuais, planos, etc.?
- Quão fácil é a detecção, identificação ou medição de um ponto de referência?

Um ponto importante é saber se o marcador a ser utilizado é natural ou sintético. Marcadores artificiais colocados no ambiente especialmente para o propósito da localização do robô são tipicamente mais fáceis de serem detectados e podem ser rotulados individualmente. A amplitude da restrição de posicionamento providas pelas observações de um marcador depende do tipo de sensor e da geometria do ponto de referência. Marcadores planares, por exemplo, podem fornecer somente restrições unidimensional ao posicionamento do robô (i.e., a distância ao longo da normal do marcador). Isto é especialmente verdadeiro nos casos de sensores de medição de distância (laser, sonar, etc.) em particular, porque as medições são invariantes a translação paralela a face do marcador.

No contexto da robótica móvel, várias modalidades de sistemas sensores são de particular interesse para a estimação de posicionamento:

- Sistemas sensores baseados em vídeo (i.e., visão computacional) podem ser usados de vários modos. Na sua forma mais simples, podem fornecer suporte, e talvez extensão para os marcadores definidos visualmente. A lâmpada incandescente comum é um clássico exemplo de um marcador visual artificial. A utilização de marcadores naturais pode necessitar de processamento de imagem com alto grau de complexidade. Muitos ambientes têm marcadores naturais robustos que produzem excelentes alvos. Por exemplo, uma porta num corredor, que pode estar aberta ou fechada, geralmente produz um marcador visual confiável.
- Transmissões a laser acompanhadas de sensores visuais merecem uma atenção especial. Marcadores retro-reflexivos podem prover detectabilidade a longa distância.

- Marcadores ativos que são transmissores de rádio frequência formam uma classe já bem estabelecida de pontos de referência de estimação de posicionamento. O sistema terrestre LORAN ⁴ (*L*ong *R*ange *N*avigation) tem sido utilizado por anos, e o sistema GPS é baseado numa rede de satélites e pode ser utilizado para quase todas as localizações em campo aberto, conforme pode ser visto na seção A.9.
- Devido ao fato do sonar ser ubíquo e relativamente barato, tem sido considerado como opção para estimativa de posicionamento a despeito dos inconvenientes em termos da dispersão do feixe de RF, reflexões especulares e ruídos de fundo. Ele tem sido utilizado, e funciona bem, em ambientes com estruturas geométricas, que são referenciadas como *marcadores geométricos*, conforme pode ser visto na seção A.6.

B.3.3 Triangulação

O termo *triangulação* refere-se às soluções de equações de restrições referente ao posicionamento de um observador com referência a um conjunto de marcadores. Para a estimativa de posicionamento utiliza-se métodos de triangulação a partir de marcadores conhecidos. Isto vem sendo praticado desde épocas memoriais e foi explorado pelos romanos antigos nos mapeamentos e construções de estradas durante o império romano.

O caso mais simples que deu à técnica o nome, é aquele que utiliza direção angular (rumo) ou medições de distância para dois (ou mais) marcadores para se resolver problemas de posicionamento planares. O problema é solucionado a partir dos parâmetros de um dado triângulo com combinações de lados e ângulos. Este método de estimativa de posicionamento tem suas origens na antiguidade no contexto da arquitetura e cartografia e é importante hoje em dia em vários domínios tecnológicos. Embora a geometria triangular não seja a única configuração possível para a utilização de marcadores, é a mais naturalmente empregada.

Embora os marcadores e o robôs móveis coexistam num ambiente 3D, a precisão limitada associada com informações recebidas de marcadores posicionados em alturas um pouco mais elevadas, freqüentemente, na prática, são aproximados para problemas 2D (bidimensionais) (MADSEN; ANDERSEN, 1998). Desta maneira, embora, na realidade, o problema de triangulação para um robô considerado pontual tenha seis parâmetros desconhecidos (três

⁴O LORAN é um sistema de navegação composto de uma rede de transmissores de rádio de baixa frequência que, através das medidas do intervalo de tempo entre os sinais de rádio recebidos a partir de três ou mais estações consegue determinar a posição de uma aeronave ou de um navio. Antes da popularização dos sistemas GPS, era bastante utilizado em aplicações navais e aeronáuticas. A versão corrente do LORAN em uso comum é o LORAN-C, que opera na banda de frequência de 90 à 110 kHz.

posições e três orientações variáveis), freqüentemente são aproximados para o plano, com robô e marcadores 2D.

Dependendo das combinações dos lados L e ângulos A dados, o problema de triangulação pode ser descrito como *lado-lado-lado* (LLL), *lado-ângulo-lado* (LAL), e assim por diante. Todos os casos permitem uma solução, exceto pelo caso AAA cuja escala do triângulo não pode ser determinada pelos parâmetros. Na prática, as tecnologias de sensores geralmente retornam uma medição angular e uma medição de distância com respeito as posições conhecidas dos marcadores. Desta maneira, os casos LAA e LLL são os mais comuns encontrados. Mais geralmente, o problema pode envolver combinações de restrições algébricas que relaciona as medições com respeito aos parâmetros do posicionamento do robô. Estas situações são tipicamente não lineares, e soluções são dependentes das estimativas de posição inicial e restrições. Pode ser formulado como

$$\mathbf{x} = F(m_1, m_2, \dots, m_n), \quad (\text{B.15})$$

onde o vetor \mathbf{x} é composto das variáveis do posicionamento a serem estimadas (normalmente, para os casos 2D, $[x \ y \ \theta]$), e $\mathbf{m} = m_1, m_2, \dots, m_n$ é o vetor de medições a ser utilizado. No caso específico da estimativa de posição de um robô orientado num plano, temos que

$$\begin{aligned} x &= F_1(m_1, m_2, \dots, m_n) \\ y &= F_2(m_1, m_2, \dots, m_n) \\ \theta &= F_3(m_1, m_2, \dots, m_n). \end{aligned} \quad (\text{B.16})$$

Se somente a distância até o marcador é disponibilizada, restringe-se a uma simples medição, a posição do robô ao arco do círculo. A Figura B.4 ilustra talvez, o caso mais simples de triangulação.

Um robô está numa localização desconhecida \mathbf{x}_1 e percebe os dois marcadores P_1 e P_2 medindo-se as distâncias d_1 e d_2 até eles. Este exemplo corresponde ao caso em que os marcadores são ativos, estão localizados em pontos conhecidos, emitem um sinal característico e o robô obtém as distâncias baseado no tempo de atraso de chegada do sinal. O robô deve estar na intersecção do círculo de raio d_1 e centro P_1 com o círculo de raio d_2 e centro P_2 . Assumindo-se que P_1 está na origem e que P_2 está no ponto $(h, 0)$, tem-se que

$$d_1^2 = x^2 + y^2 \quad (\text{B.17})$$

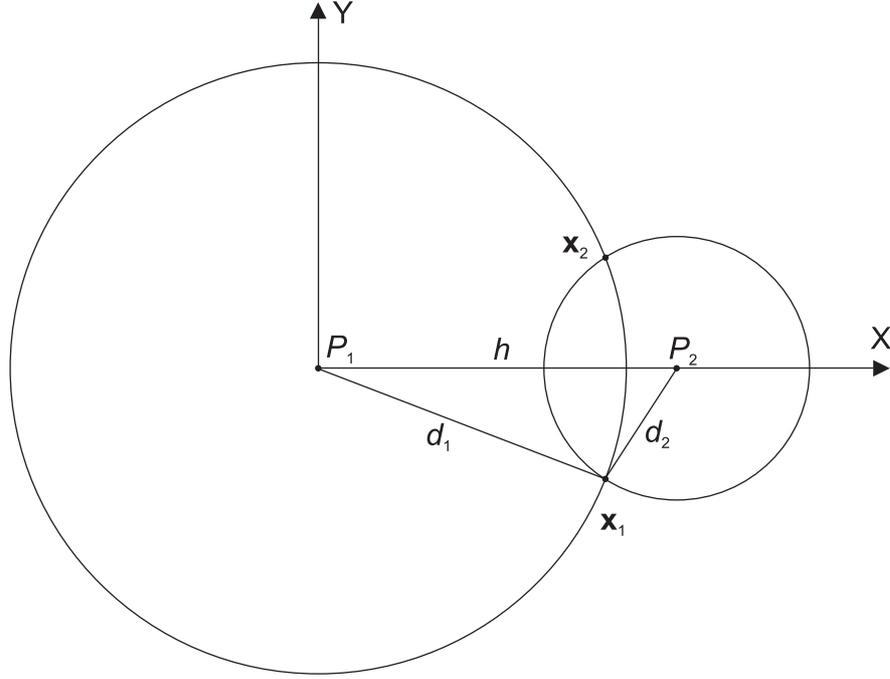


Figura B.4: Exemplo de triangulação simples. O robô móvel está na posição \mathbf{x}_1 .

e

$$d_2^2 = (h - x)^2 + y^2. \quad (\text{B.18})$$

Isolando-se o termo y^2 nas equações B.17 e B.18 e fazendo-se as substituições, resulta-se em

$$x = \frac{h^2 + d_1^2 - d_2^2}{2h} \quad (\text{B.19})$$

e

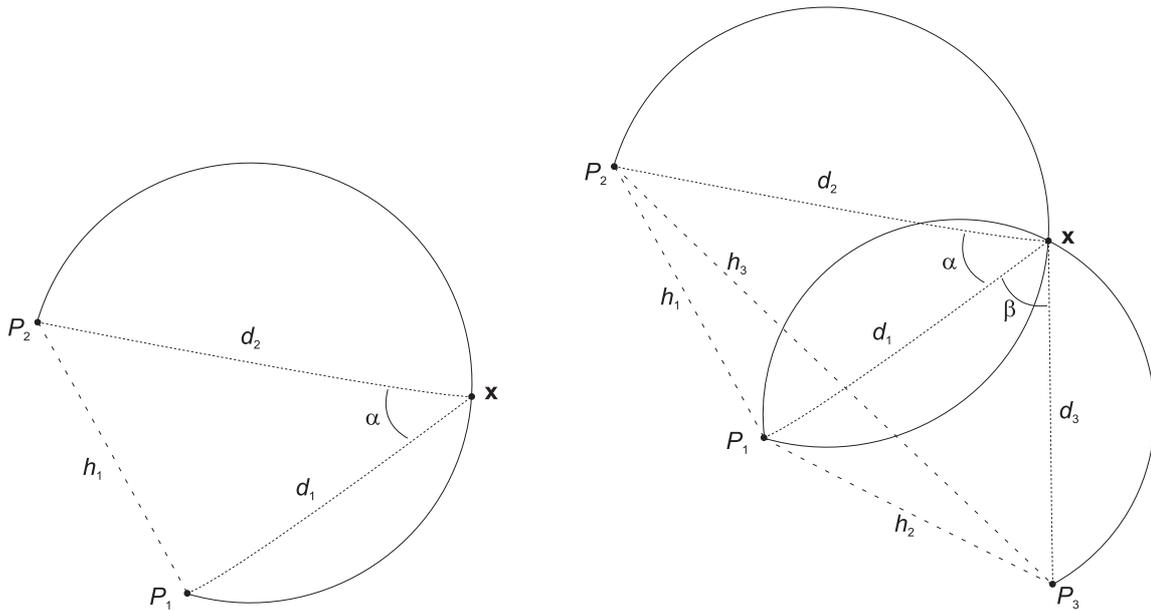
$$y = \pm \sqrt{d_1^2 - x^2} = \pm \sqrt{d_1^2 - \left[\frac{h^2 + d_1^2 - d_2^2}{2h} \right]^2}, \quad (\text{B.20})$$

o que resulta em duas soluções, para \mathbf{x}_1 e \mathbf{x}_2 .

Em aplicações típicas, os marcadores são posicionados nas paredes, e, desta maneira, as soluções espúrias (no nosso exemplo, o \mathbf{x}_2) podem ser identificadas pois correspondem a uma localização do robô no exterior da parede.

Embora distâncias aos marcadores forneçam exemplo simples de triangulação, muitos sistemas sensores em conjunto com marcadores resultam em situações mais complexas. Uma situação comum, que se apresenta nos sistemas robóticos móveis é quando a direção angular relativa ao marcador esteja disponível mas a distância não. Considerando-se, por exemplo, uma câmara que consegue detectar dois pequenos marcadores conhecidos. Assumindo-se que

a câmara tenha sido calibrada, pode-se extrair a separação angular entre os marcadores a partir de suas posições na imagem. Utilizando-se a separação angular ou a direção angular relativa entre os dois marcadores com restrições de posição conhecidas, a posição do robô deve estar localizada sob um par de arcos circulares. Se o robô pode determinar o registro dos marcadores num certo sentido (por exemplo, o sentido horário) ou identificá-los individualmente através de seus respectivos rótulos, a posição do robô fica restringida a apenas um destes dois círculos. A única exceção é quando os marcadores estejam dispostos num círculo comum, o que, em contrapartida, implica que os círculos restritos são coincidentes ao invés de meramente cruzarem-se num par de pontos (SUGIHARA, 1988).



(a) Triangulação com dois marcadores.

(b) Triangulação com três marcadores.

Figura B.5: Estimativa da localização do robô móvel baseado na triangulação com marcadores.

A situação para dois marcadores é ilustrada na Figura B.5(a). O robô localiza dois marcadores pré-estabelecidos e mede a distância para cada um. Desta maneira, a posição do robô fica restrita ao arco mostrado na Figura B.5(a). matematicamente a situação admite dois arcos como solução, mas uma delas pode ser excluída baseado no posicionamento conhecido dos marcadores (normalmente estão localizados nos extremos do ambiente, por exemplo, as paredes, as quinas e os cantos). O conjunto de pontos possíveis da localização do robô é dado por

$$h_1 = \sqrt{d_1^2 + d_2^2 - 2|d_1||d_2| \cos \alpha}, \quad (\text{B.21})$$

onde d_1 e d_2 são as distâncias da posição \mathbf{x} do robô aos marcadores P_1 e P_2 , respectivamente. Nota-se que estas distâncias, d_1 e d_2 , que são as variáveis a serem encontradas, ficam sem solução para o caso de triangulação com dois marcadores, como mostra a Figura B.5(a), pois neste caso tem-se uma equação com duas incógnitas. Se o robô móvel consegue detectar três marcadores, conforme ilustrado na Figura B.5(b), pode-se deduzir pela lei dos cossenos três restrições não lineares à localização do robô, com relação às distâncias d_1 , d_2 e d_3 :

$$h_1 = \sqrt{d_1^2 + d_2^2 - 2|d_1||d_2|\cos\alpha} \quad (\text{B.22})$$

$$h_2 = \sqrt{d_1^2 + d_3^2 - 2|d_1||d_3|\cos\beta} \quad (\text{B.23})$$

$$h_3 = \sqrt{d_2^2 + d_3^2 - 2|d_2||d_3|\cos(\alpha + \beta)}. \quad (\text{B.24})$$

Este conjunto de equações não lineares, podem ser resolvidas usando-se técnicas padrão, tal como o método dos mínimos quadrados, para se obter d_1 , d_2 e d_3 . Com o conhecimento das distâncias d_1 , d_2 e d_3 pode-se deduzir a posição \mathbf{x} e a orientação θ do robô com relação ao plano que se encontra o mesmo e os marcadores (BETKE; GURVITS, 1997).

O arranjo geométrico dos marcadores com respeito às observações feitas pelo robô é crítico para a precisão da solução. Um arranjo particular dos marcadores pode resultar em alta precisão da estimativa de posicionamento de um certo robô móvel, quando observado de uma certa localidade, e baixa precisão quando observado de outras localidades. Por exemplo, num ambiente 2D, um conjunto de três marcadores colineares observados pelo robô, por um certo dispositivo de medição de direção angular, pode resultar em boa precisão de posição feita por triangulação quando observados de um ponto longe da linha de junção dos marcadores (e.g., um ponto que forma um triângulo equilátero com relação aos marcadores). Por outro lado, se o robô estiver localizado no mesmo eixo dos marcadores, a estimativa de posicionamento do mesmo fica expandido a todo este eixo, que pode ser infinito.

B.3.4 Triangulação com incerteza

No caso real de triangulação a partir de medições de distâncias ou de direção angular, aos dados colhidos pelos sensores devem ser associados certa incerteza, devido aos erros nas medições. Como resultado, as estimativas de posição produzidas pelas técnicas de triangulação resultam em distribuições de probabilidade para o posicionamento do robô. Embora casos particulares de degeneração que leva à estimativas de posicionamento ambíguas não sejam interessantes, o aumento dos erros de entrada que enfraquecem a estimativa de posição é uma ocorrência comum (SUTHERLAND; THOMPSON, 1993). A Figura B.6 ilustra dois ca-

de estimativa de posicionamento por triangulação utilizando-se medidas imprecisas de direção angular. Na Figura B.6(a), com dois marcadores, a região ambígua é toda a área em formato de meia lua que pode se encontrar posicionado o robô móvel R . Para o caso de três marcadores, conforme pode ser visto na Figura B.7, a região possível de se encontrar o robô móvel R é a área delimitada pela duas meia luas.

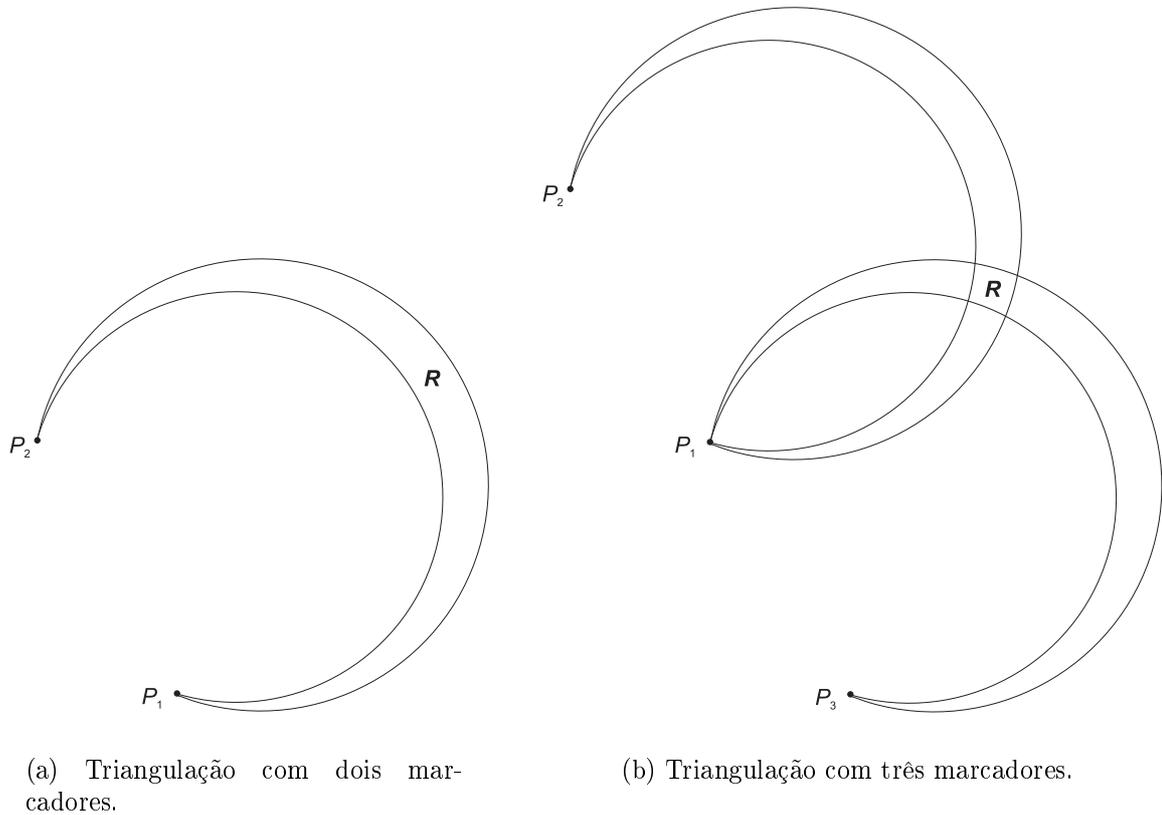


Figura B.6: Estimativa da localização do robô móvel baseado na triangulação com incerteza.

A relação entre a precisão das medições de entrada e precisão da estimativa final das variáveis do posicionamento desejado pode ser dado pela *diluição geométrica da precisão*⁵ (DGP). A DGP é uma medida que expressa a variação na estimativa de saída X (i.e., as

⁵Em inglês: *Geometric Dilution of Precision* (GDOP). No caso da navegação por satélite utilizando-se do sistema GPS (seção A.9), para obter-se a posição com o maior rigor possível, o receptor GPS leva em conta o princípio GDOP, que se refere ao fato de que a solução do problema de determinação da posição pode ser melhor ou pior, dependendo do posicionamento dos satélites utilizados nas medidas. Todos os receptores GPS são projetados para selecionar uma configuração de satélites que proporcione a melhor geometria. A geometria dos satélites é apresentada ao usuário por um fator de diluição da precisão da posição, conhecido como PDOP (*Positional Dilution of Precision*), que deve ser usado como um indicador da qualidade da informação obtida. Os fatores PDOP são números relativos; quanto menor o fator PDOP, melhor a precisão da posição. Inversamente, quanto maior o valor do fator, pior a qualidade da determinação correspondente (ou seja, maior a influência dos erros de observação nos resultados do posicionamento).

variáveis geométricas que constituem o posicionamento) com variações nos parâmetros de entrada S (i.e., os dados dos sensores):

$$DGP = \frac{\Delta X}{\Delta S} \quad (\text{B.25})$$

Quando toma-se $\Delta S \rightarrow 0$, tem-se que

$$\lim_{\Delta S \rightarrow 0} DGP = \mathcal{J}, \quad (\text{B.26})$$

onde \mathcal{J} é o jacobiano das equações de medições. No espaço 3D, a *diluição geométrica horizontal da precisão*⁶ (DGHP) e a *diluição geométrica posicional da precisão*⁷ (DGPP) referem-se a sensibilidade de erro somente de componentes horizontais ou translacionais, enquanto que DGP refere-se a sensibilidade de toda classe de variáveis.

Considerando-se o caso mais simples de marcadores, conforme pode ser visto na Figura B.4, a DGP é dada por

$$DGP = \mathcal{J} = \begin{pmatrix} \frac{d_1}{h} & -\frac{d_2}{h} \\ \frac{d_1(h-x)}{hy} & \frac{xd_2}{hy} \end{pmatrix}. \quad (\text{B.27})$$

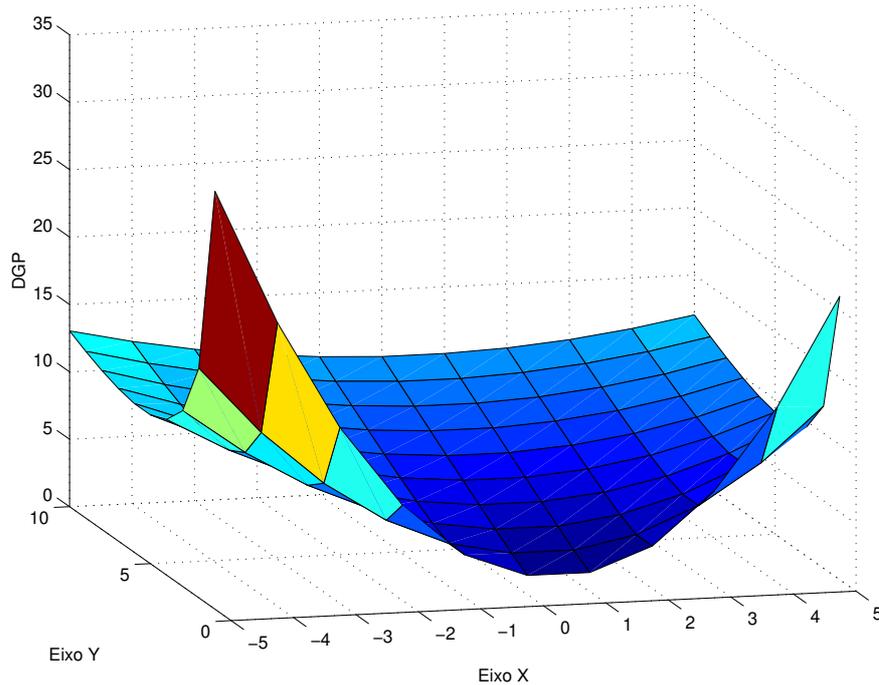


Figura B.7: DGP para dois marcadores posicionados em (0,0) e (1,0).

⁶Em inglês: *horizontal geometric dilution of precision* (HDOP).

⁷Em inglês: *positional geometric dilution of precision* (PDOP).

A magnitude de \mathcal{J} é plotado na Figura B.7, onde os dois marcadores estão localizados nas coordenadas $(0,0)$ e $(1,0)$. O $|\mathcal{J}|$ cresce quando o robô se distancia do eixo x . Isto é esperado pelo fato de que a separação efetiva entre os dois círculos diminui quando o robô se distancia dos marcadores. Como a precisão da maioria dos sensores diminui com a distância, a utilização de marcadores muito distantes deve ser evitado. Observa-se o incremento do DGP ao longo do eixo x ao se distanciar dos marcadores. O erro que aparece em cada estimativa de posição cresce muito rapidamente quando ambos os marcadores estão no mesmo lado e em alinhamento com o robô. O conhecimento da forma do DGP pode ser utilizado pelo robô para se selecionar quais marcadores poderão ser usados se múltiplas opções de marcadores estiverem disponíveis.

Em situações práticas, sistemas robóticos móveis podem encontrar dificuldades nas identificações dos marcadores ou a estimativa de posição pode ser instável devido à geometria dos marcadores. Uma opção para se lidar com informações intermitentes dos marcadores é a combinação das informações baseadas nas triangulações com os marcadores com as informações de posicionamento advindas do odômetro (ou outro sistema de medição de deslocamento). Esta fusão de informações de duas (ou mais) fontes tem sido bastante utilizada com o auxílio do filtro de Kalman por diversos pesquisadores (BOZIC, 1994)(LEONARD; DURRANT-WHYTE, 1991)(WANG; WILSON, 1992)(NASHASHIBI; DEVY, 1996), conforme pode ser visto na seção B.4.4.

B.4 Fusão de dados sensoriais

A questão de como se combinar dados de diferentes fontes gera uma grande massa de pesquisa nos meios acadêmicos e nos laboratórios de pesquisas. No contexto dos sistemas robóticos móveis, a fusão de dados deve ser efetuada em pelo menos três domínios distintos: combinando-se medições de diferentes sensores, combinando-se medições de diferentes posições, e combinando-se medições de diferentes tempos. A seguir é considerado a fusão de dados genéricos, sendo abordando uma pequena parcela de um assunto bem mais amplo.

A combinação de medições de múltiplas posições, tempos ou sensores está proximamente relacionado com interpolações, extrapolações, e o ajuste de funções analíticas para dados. Em adição a associação de um conjunto de medições para se obter uma estimativa desta associação, seria mais interessante uma estimativa da confiança da estimativa final dada a confiança das medições de entrada individuais.

Na forma estatística mais simples, este processo de se combinar múltiplas medições x_i ,

cada uma com igual certeza, toma a forma da média não ponderada e produz uma estimativa combinada \hat{x} , que é a média aritmética:

$$\hat{x} = \frac{\sum x_i}{n}, \quad (\text{B.28})$$

que pode ser representada com medições escalares ou vetoriais. Outrossim, a confiança da média final é expressa pela variância $\hat{\sigma}$ (ou para medições vetoriais, a matriz covariância), que pode também ser derivado diretamente da variância $\sigma_i = x_i - \hat{x}$ das medições individuais, como

$$\hat{\sigma}^2 = \frac{\sigma_i^2}{n}. \quad (\text{B.29})$$

A matriz covariância para o vetor de média corrigida $[x_1, x_2, \dots, x_n]$ é dado por

$$\begin{bmatrix} \sum x_1^2 & \sum x_1 x_2 & \cdots & \sum x_1 x_n \\ \sum x_2 x_1 & \sum x_2^2 & \cdots & \sum x_2 x_n \\ \vdots & \vdots & \ddots & \vdots \\ \sum x_n x_1 & \sum x_n x_2 & \cdots & \sum x_n^2 \end{bmatrix}. \quad (\text{B.30})$$

Freqüentemente as medições feitas com diferentes precisões devem ser combinadas. Neste caso, quanto mais as medições mais precisas tiverem um peso maior, mais fortemente influenciarão na média, e o erro composto deve também ter um peso maior para se produz dados mais precisos e confiáveis (SPIEGEL, 1999).

B.4.1 Modelos de espaço de estado

Em geral, um sistema pode ser descrito por um conjunto de parâmetros ou variáveis que caracterizam os aspectos relevantes de seu comportamento. A qualquer tempo, a informação de interesse com respeito ao sistema pode ser descrita por um vetor \mathbf{x} cujos componentes são as variáveis de interesse. O vetor \mathbf{x} especifica um ponto na representação do espaço de estado do sistema. Nota-se que algumas (ou todas) destas variáveis de interesse não podem ser medidas diretamente. Desta maneira, o vetor de estado deve ser estimado utilizando-se algum vetor de medição \mathbf{z} (KIM; NELSON, 1999).

No contexto dos sistemas robóticos móveis, e mais geralmente no contexto da teoria de controle, é sempre importante considerar-se uma versão dinâmica do problema, cujas estimativas de um conjunto de variáveis de interesse devem ser formadas em uma base contínua utilizando-se observações feitas nos dados, isto é, nos tempos $t = 0, \dots, k$.

Um sistema para o qual o vetor de estado pode ser totalmente determinado a partir de um número suficiente de medições é descrito com sendo *observável*. Conforme utilizado por Bar-Shalom (BAR-SHALOM; FORTMANN; CABLE, 1990), para se descrever a estimativa de estado a ser computada, utiliza-se $\hat{\mathbf{x}}(k|k_1)$ para se denotar a estimativa do vetor \mathbf{x} no passo de tempo k usando-se dados coletados num período de tempo em que esteja incluindo o passo de tempo k_1 . Utilizando-se as observações até o passo k , porém excluindo-se k , para se formar uma predição, esta é em geral expressa como $\tilde{\mathbf{x}}(k)$ que denota a *predição* do vetor de estado $\mathbf{x}(k)$ baseado na disponibilidade de informação estritamente antes do tempo k . É também encontrado na literatura (GELB, 1974) a notação $\underline{\hat{\mathbf{x}}}_k(-)$ como opção à notação $\tilde{\mathbf{x}}(k)$. Baseando-se na disponibilidade de informação até, e incluindo-se, o tempo k , forma-se um estado atualizado *estimado* $\hat{\mathbf{x}}(k)$, que denota a *estimativa* do vetor de estado $\mathbf{x}(k)$ no tempo k , também aludido como $\underline{\hat{\mathbf{x}}}_k(+)$ na literatura (JAZWINSKI, 1970).

Técnicas de estimação de estado, como os filtros Kalman apresentado na seção B.4.4, tipicamente necessitam um modelo de como o sistema de interesse se desenvolve no tempo assim como um modelo de como os sensores operam. No contexto da teoria de controle, a descrição do sistema cujo estado é de interesse é usualmente referida como *modelo de planta*. Um terceiro componente do filtro é a descrição das entradas de controle permitidas $\mathbf{u}(t)$, e como elas afetam o estado do sistema.

B.4.2 Modelo de planta

Um modelo de planta que descreve como o estado do sistema $\mathbf{x}(k)$, que no nosso caso particular representa a posição do robô móvel, muda em função do tempo, da entrada de controle $\mathbf{u}(k)$ e do ruído $\mathbf{v}(k)$ pode ser expresso por

$$\mathbf{x}(k+1) = \mathbf{\Psi}(\mathbf{x}(k), \mathbf{u}(k)) + \mathbf{v}(k), \quad (\text{B.31})$$

onde $\mathbf{\Psi}()$ é a função transição de estado e $\mathbf{v}(k)$ é a função ruído. Uma das formas mais comum e convencional para se representar o modelo de ruído é utilizando-se o modelo de ruído gaussiano de média zero com covariância $\mathbf{Q}(k)$ ⁸ (GELB, 1974). Este modelo particular de ruído vem de encontro aos requisitos do filtro de Kalman, descrito na seção B.4.4.

Um modelo de planta linear, a partir da equação B.31, pode ser escrito como

$$\mathbf{x}(k+1) = \mathbf{\Psi}\mathbf{x}(k) + \mathbf{\Upsilon}\mathbf{u}(k) + \mathbf{v}(k), \quad (\text{B.32})$$

⁸Gelb também utiliza a notação $\mathbf{v}(k) \sim \mathcal{N}(0, \mathbf{Q}(k))$ para se representar o modelo de ruído gaussiano de média zero com covariância $\mathbf{Q}(k)$.

onde a matriz Ψ expressa como o sistema evolui a partir de um estado para outro na ausência de entradas (que é freqüentemente dado como uma matriz identidade) e a matriz Υ expressa como as entradas do controle modificam o estado do sistema.

Considerando-se um robô móvel onidirecional linear com deslocamentos restringidos ao plano, pode-se propor um modelo simples de planta, a seguir apresentado.

O estado do robô $\mathbf{x}(k)$ pode ser descrito como sendo

$$\mathbf{x}(k) = \begin{bmatrix} x(k) \\ y(k) \end{bmatrix}, \quad (\text{B.33})$$

onde $(x(k), y(k))$ descreve a posição do robô no sistema de coordenada global. Supondo-se que o robô esteja equipado com algum sistema de locomoção onidirecional. Então, a entrada de controle $\mathbf{u}(k)$ pode ser descrita como uma mudança independente na localização x e y do robô, e assim

$$\mathbf{u}(k) = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}. \quad (\text{B.34})$$

Se o erro no movimento do robô é independente nas direções x e y , e se este erro pode ser modelado por alguma função de ruído $v_x(k)$ e $v_y(k)$, então o modelo de planta do robô é dado pela equação B.31, que para o nosso caso em particular se torna

$$\begin{bmatrix} x(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} x(k) + \Delta x(k) + v_x(k) \\ y(k) + \Delta y(k) + v_y(k) \end{bmatrix}. \quad (\text{B.35})$$

O robô se locomove para onde for comandado com cada movimento sendo corrompido pelo processo de ruído. A equação B.35 descreve um modelo de planta linear.

B.4.3 Modelo de medição

O modelo de medição descreve como os dados dos sensores variam em função do estado do sistema. Invertendo-se o modelo do sensor (caso hipotético) permite-se que os dados do sensor sejam utilizados para os cálculos do estado. Conforme explanado por Leonard (LEONARD; DURRANT-WHYTE, 1992), que desenvolveu trabalhos com robôs móveis equipados com sensores ultra-sônicos, o modelo de medição relata uma observação do sensor para posição do robô e a geometria do alvo (anteparo) que produz a observação, e tem a forma da equação:

$$\mathbf{z}_i(k) = \mathbf{h}_{st}(\mathbf{x}(k), \mathbf{p}_t) + \mathbf{w}_i(k), \quad (\text{B.36})$$

onde $\mathbf{w}_i(k)$ é a função ruído para $\mathbf{w}_i(k) \sim N(0, \mathbf{R}_i(k))$ que representa o modelo de ruído gaussiano de média zero com a matriz covariância $\mathbf{R}_i(k)$, \mathbf{p}_t é o vetor de estado de alvo e varia conforme o formato do alvo, que podem ser basicamente cantos, quinas, superfícies cilíndricas ou superfícies planas. A função de medição $\mathbf{h}_{st}(\mathbf{x}(k), \mathbf{p}_t)$ expressa uma observação $\mathbf{z}(k)$ do sensor s para o alvo t como um função da localização do veículo $\mathbf{x}(k)$ e da geometria do alvo.

Assim como no modelo de planta, um modelo de medição linear é particularmente interessante. Da equação B.36, este toma a forma de

$$\mathbf{z}_i(k) = \mathbf{\Lambda}_E \mathbf{x}(k) + \mathbf{w}_i(k), \quad (\text{B.37})$$

onde $\mathbf{\Lambda}_E$ é a matriz que expressa como as medições são derivadas com uma transformação linear do estado. Este simples caso ilustra como uma estimativa do estado pode ser recuperada a partir das medições:

$$\hat{\mathbf{x}}(k) = \mathbf{\Lambda}_E^{-1} \mathbf{z}_i(k) \quad (\text{B.38})$$

se for assumido que a matriz $\mathbf{\Lambda}_E$ seja invertível.

Como exemplo de um modelo simples de medição linear, assume-se que um certo robô móvel seja equipado com um sensor capaz de estimar o deslocamento do robô desde a origem até um cetro ponto. Então, da equação B.36 tem-se que

$$\mathbf{z}_1(k) = \begin{bmatrix} x(k) + w_x(k) \\ y(k) + w_y(k) \end{bmatrix}. \quad (\text{B.39})$$

Neste exemplo de modelo linear, tem-se que

$$\mathbf{\Lambda}_E(k) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (\text{B.40})$$

e a estimativa da posição do robô móvel é apresentada como

$$\hat{\mathbf{x}}(k) = \mathbf{\Lambda}_E^{-1} \mathbf{z}_i(k) = \mathbf{z}_1(k) \quad (\text{B.41})$$

porque este modelo de sensor é invertível.

O exemplo apresentado anteriormente utiliza modelos lineares simples para o estado, as entradas, e os processos de medições. Infelizmente, a maioria dos robôs móveis tracionados por rodas (RMTR)⁹ não podem ser modelados desta maneira, e é necessário se considerar

⁹Do inglês: WMRs - *wheeled mobile robots*

planta, modelo e processos de estimação não lineares.

Considerando-se agora, como exemplo, um sistema não linear, onde o robô pode ser modelado como um robô pontual e que tem-se controle independente sobre a orientação e a velocidade do robô, como é mostrado no caso dos robôs móveis com transmissão síncrona (seção 2.4). Então, para este tipo de robô, a entrada de controle $\mathbf{u}(k)$ pode ser descrita como

$$\mathbf{u}(k) = \begin{bmatrix} T(k) \\ \Delta\theta(k) \end{bmatrix}, \quad (\text{B.42})$$

ou seja, no período k até $k + 1$ o robô se movê de uma distância $T(k)$ para frente, na direção que estiver apontando, e então se rotaciona $\Delta\theta(k)$. O estado de sistema é dado por

$$\mathbf{x}(k) = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix}, \quad (\text{B.43})$$

e o modelo de planta não linear é dado por

$$\Psi(\mathbf{x}(k), \mathbf{u}(k)) = \begin{bmatrix} x(k) + T(k) \cos[\theta(k)] \\ y(k) + T(k) \text{sen}[\theta(k)] \\ \theta(k) + \Delta\theta(k) \end{bmatrix}, \quad (\text{B.44})$$

como é ilustrado na Figura B.8. Cada movimento do robô tem uma parcela de processo de ruído $\mathbf{v}(k)$, que tem uma matriz covariância conhecida ou estimável $\mathbf{Q}(k)$. Assume-se para este processo de ruído que o mesmo satisfaz as condições necessárias que assegura a utilização do filtro de Kalman (apresentado na seção B.4.4). Se, na prática, o robô se move em passos distintos compostos de rotações puras ou translações puras (i.e., apenas um entre $\Delta\theta(k)$ e $T(k)$ é diferente de zero), então apenas duas versões de $\mathbf{Q}(k)$ são necessárias.

Supondo-se agora que o robô seja equipado com um sensor que pode determinar a distância do robô de um determinado marcador alvo no ambiente. Por exemplo, o alvo pode emitir um único som a uma frequência conhecida, e o robô é equipado com um receptor que capta o som. Se o robô e o som emitido pelo alvo tiverem os relógios sincronizados, a distância entre o alvo e o robô pode ser estimada. Se o alvo estiver localizado em (x_s, y_s) , o modelo de medição para este robô vem da equação B.36 e é dado por

$$\mathbf{z}_1(k) = \sqrt{[x(k) - x_s]^2 + [y(k) - y_s]^2} + \mathbf{w}_1(k). \quad (\text{B.45})$$

Este modelo de medição tem a parcela de degradação das medidas dada pelo processo de

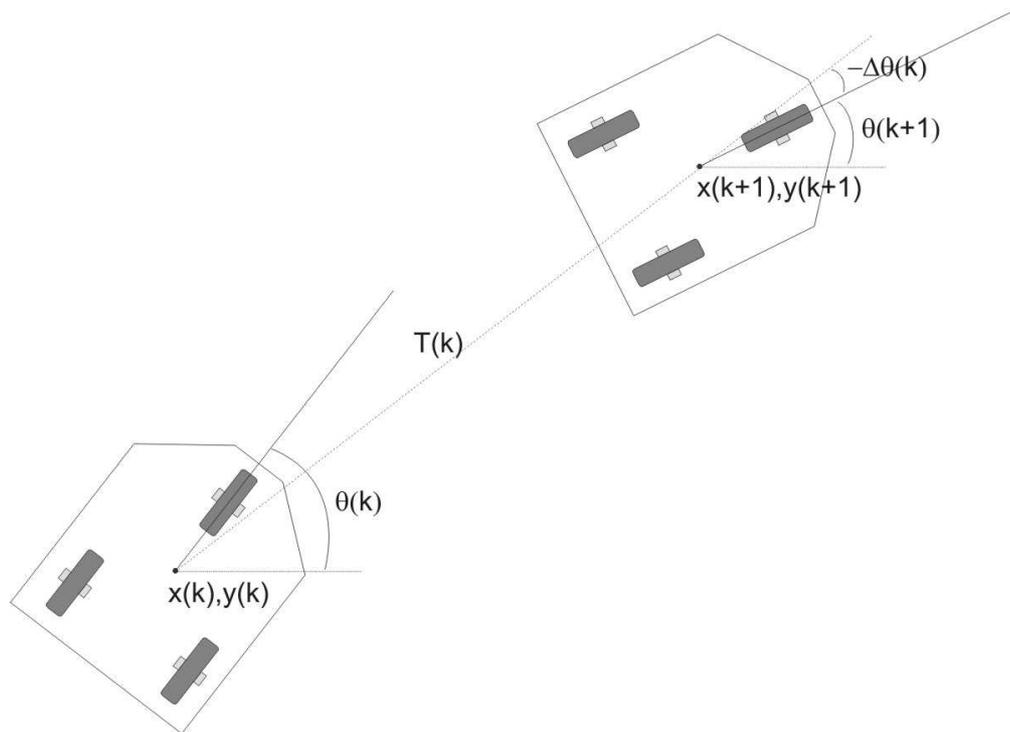


Figura B.8: Deslocamento do vetor posicionamento $\mathbf{x}(k)$.

ruído $\mathbf{w}_1(k)$ com a matriz covariância $\mathbf{R}_1(k)$.

B.4.4 Filtro de Kalman

Para se controlar um robô móvel, freqüentemente deve-se combinar informações de múltiplas fontes. As informações advindas de fontes confiáveis devem ter um peso maior sobre aquelas coletadas por sensores menos confiáveis. Um esquema geral para se computar as fontes que são mais ou menos confiáveis e quais pesos devem ser dados aos dados de cada fonte, perfazendo-se uma soma ponderada das medições, é conhecido com filtro de Kalman.

O filtro de Kalman recebe o nome do seu inventor, Rudolf E. Kalman, que publicou a idéia em 1960 (KALMAN, 1960). É um dos métodos mais amplamente utilizados para fusão sensorial em aplicações de robótica móvel (LEONARD; DURRANT-WHYTE, 1991). Este filtro é freqüentemente usado para combinar dados obtidos de diferentes sensores em uma estimativa estatisticamente ótima. Se um sistema pode ser descrito através de uma modelo linear e as incertezas dos sensores e do sistema podem ser modelados como ruídos gaussianos brancos, então o filtro de Kalman proverá uma estimativa estatisticamente ótima para os dados fundidos. Isto significa que, sob certas condições, o filtro de Kalman é capaz de

encontrar a melhor estimativa baseada em correção de cada medida individual (DUDEK et al., 1996). Além da robótica, o filtro de Kalman é usado em uma ampla gama de aplicações na engenharia, dos radares à visão computacional.

Se os processos de medições satisfazem certas propriedades, tal como erro de média zero, o filtro de Kalman provê um método ótimo demonstrável para a fusão de dados, principalmente utilizando-se o método dos mínimos quadrados. Particularmente, em aplicações da robótica móvel, o filtro de Kalman é utilizado para se manter uma estimativa avançada da posição e orientação do veículo, ou dos parâmetros que descrevem objetos de interesse no ambiente, tal como outro robô móvel que estiver trafegando no ambiente. O filtro de Kalman permite que uma estimativa existente da posição do robô, por exemplo, possa ser combinada com a informação de posição a partir de um ou mais sensores. Um atributo do filtro de Kalman é que ele fornece uma estimativa avançada não somente como uma variedade de parâmetros, mas também como a confiança nestas estimativas na forma da matriz covariância. Sobre certas circunstâncias, o filtro de Kalman realiza estas atualizações de uma maneira ótima de tal forma que minimiza o erro esperado na estimativa.

As suposições feitas pelo filtro de Kalman são as seguintes:

- ruído de sistema de média zero $E[v_i] = 0$ onde $E[]$ é o valor esperado;
- ruído independente: $E[v_i v_j] = 0 \forall i \neq j$. Caso contrário, a covariância de ruído de sistema é dado por $E[v_i v_j] = \sigma_i(k)$;
- um modelo linear da evolução do sistema sobre o tempo;
- uma relação linear entre o estado de sistema (i.e., o posicionamento) e as medições sendo realizadas.

Infelizmente, é raro o caso em que todos estas condições sejam satisfeitas em aplicações práticas. Neste caso, o filtro de Kalman pode ainda ser utilizado, mas a certeza de atingir o ponto ótimo não será válida. Se as condições forem distorcidas de maneira severa, o filtro pode, de fato, produzir resultados muito pobres (SORENSEN, 1970).

B.4.4.1 Combinando-se medições e estimativa

No caso de dados quantitativos, e em particular nos dados estimados de estado, um caminho natural para descrever a confiança numa medição é pelo uso da matriz covariância.

Assumindo-se que os erros são distribuídos normalmente, é possível então descrever-se a distribuição da probabilidade para erros explicitamente em torno da média. O filtro de Kalman é um mecanismo utilizado para combinar informações de tal forma que as confiáveis tenham um maior peso. Para tanto, utiliza-se o ganho de Kalman para se atribuir pesos às contribuições relativas as novas medições com relação às expectativas anteriores. O ganho de Kalman varia na proporção direta à matriz covariância de estado e inversa à matriz covariância de medições (GELB, 1974).

O filtro de Kalman consiste dos estágios a seguir apresentados em cada passo de tempo, exceto pelo passo inicial. Assume-se, para simplificação do modelo, que a matriz de transição de estado Ψ e a função observação Λ_E permanecem constante em função do tempo. Utilizando-se o modelo de planta da equação B.32 e computando-se uma estimativa de estado de sistema no tempo $(k + 1)$ baseado no conhecimento da posição do robô no instante de tempo k , tem-se como o sistema evolui no tempo com a entrada de controle $\mathbf{u}(k)$:

$$\hat{\mathbf{x}}(k + 1) = \Psi\hat{\mathbf{x}}(k) + \Upsilon\mathbf{u}(k). \quad (\text{B.46})$$

Em alguns equações práticas a entrada $\mathbf{u}(k)$ não é utilizada. Pode-se também, atualizar-se a certeza do estado como expressado pela matriz covariância de estado $\mathbf{P}()$ através do deslocamento no tempo, como:

$$\mathbf{P}(k + 1|k) = \Psi\mathbf{P}(k)\Psi^T + \mathbf{Q}(k). \quad (\text{B.47})$$

A equação B.47 expressa a maneira com o qual o conhecimento sobre o estado do sistema gradualmente decai com o passar do tempo, na ausência de correções externas. O ganho de Kalman pode ser expresso como

$$\mathbf{K}(k + 1) = \mathbf{P}(k + 1)\Lambda_E^T\mathbf{R}_i^{-1}(k + 1), \quad (\text{B.48})$$

mas, como não se computou $\mathbf{P}(k + 1)$, pode ser computado como

$$\mathbf{K}(k + 1) = \mathbf{P}(k + 1|k)\Lambda_E^T[\Lambda_E\mathbf{P}(k + 1|k)\Lambda_E^T + \mathbf{R}_i(k + 1)]^{-1}. \quad (\text{B.49})$$

Utilizando-se esta matriz, pode-se calcular uma estimativa de estado revisado que inclui a informação adicional obtida pela medição. Isto envolve a comparação dos dados atuais dos sensores $z(k + 1)$ com os dados dos sensores previstos utilizando-se a estimativa de estado. A diferença entre os dois termos

$$\mathbf{r}(k + 1) = \mathbf{z}(k + 1) - \mathbf{h}_{st}(\hat{\mathbf{x}}(k + 1|k), \mathbf{p}_t) \quad (\text{B.50})$$

ou no caso linear

$$\mathbf{r}(k+1) = \mathbf{z}(k+1) - \mathbf{\Lambda}_E \hat{\mathbf{x}}(k+1|k) \quad (\text{B.51})$$

é referido como a inovação. Se a estimativa de estado for perfeita, a inovação deve ser não zero apenas devido ao ruído do sensor. A estimativa de estado atualizada é então dada como

$$\hat{\mathbf{x}}(k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{K}(k+1)\mathbf{r}(k+1), \quad (\text{B.52})$$

e, a matriz covariância de estado atualizada é dada por

$$\mathbf{P}(k+1) = [\mathbf{I} - \mathbf{K}(k+1)\mathbf{\Lambda}_E]\mathbf{P}(k+1|k), \quad (\text{B.53})$$

onde \mathbf{I} é a matriz identidade.

Quando este processo é utilizado na prática, o sistema é inicializado usando o estado estimado inicial, e $\mathbf{P}(0) = \mathbf{R}_i(0)$.

O sistema robótico móvel linear onidirecional citado como exemplo nas seções B.4.2 e B.4.3, pode manter uma estimativa ótima de sua posição utilizando-se o filtro de Kalman. Apresenta-se a seguir um exemplo desta utilização. Para se simplificar a exposição, é assumido que $\mathbf{Q}(k) = \mathbf{Q}$ e $\mathbf{R}_i(k) = \mathbf{R}_i$. Para cada movimento do robô, as seguintes etapas são seguidas:

1. O robô se locomove e utiliza a função da equação B.46. Os parâmetros de controle conhecidos são utilizados para a estimativa da posição do robô no tempo $k+1$, como sendo:

$$\hat{\mathbf{x}}(k+1|k) = \begin{bmatrix} x(k) + \Delta x(k) \\ y(k) + \Delta y(k) \end{bmatrix}. \quad (\text{B.54})$$

2. A incerteza do estado é gerada pela atualização da matriz covariância de estado, utilizando-se medições obtidas até o tempo k , incluindo-se o mesmo. Como $\mathbf{\Psi} = \mathbf{I}$, da equação B.47 tem-se que:

$$\mathbf{P}(k+1|k) = \mathbf{P}(k) + \mathbf{Q}. \quad (\text{B.55})$$

A incerteza na posição do robô cresce de \mathbf{Q} para cada movimento do mesmo.

3. Como num modelo de medição linear a matriz $\mathbf{\Lambda}_E = \mathbf{I}$, então da equação B.49 tem-se que o ganho de Kalman é computado como

$$\mathbf{K}(k+1) = \mathbf{P}(k+1|k)[\mathbf{P}(k+1|k) + \mathbf{R}_i]^{-1}. \quad (\text{B.56})$$

4. Uma medição é feita com o sensor, e uma estimativa de estado revisada é obtida da equação B.52 como

$$\hat{\mathbf{x}}(k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{K}(k+1)[\mathbf{z}(k+1) - \hat{\mathbf{x}}(k+1|k)]. \quad (\text{B.57})$$

Analisando-se agora a magnitude do ganho de Kalman. Se

$$|\mathbf{R}_i| \gg |\mathbf{P}(k+1|k)|, \quad (\text{B.58})$$

então a magnitude de $\mathbf{K}(k+1)$ é pequena, isto é, a certeza associada com a medição é pequena relativamente com a certeza do modelo de estado corrente, e conseqüentemente o modelo de estado antigo $\hat{\mathbf{x}}(k+1|k)$ é uma melhor estimativa de $\hat{\mathbf{x}}(k+1)$ do que é a medição do deslocamento $\mathbf{z}(k+1)$. Por outro lado, se

$$|\mathbf{P}(k+1|k)| \gg |\mathbf{R}_i|, \quad (\text{B.59})$$

então a estimativa $\hat{\mathbf{x}}(k+1)$ é atualizada mais em função da medição $\mathbf{z}(k+1)$ do que da estimativa prévia $\hat{\mathbf{x}}(k+1|k)$.

5. Finalmente, a matriz covariância de estado atualizada é dada por

$$\mathbf{P}(k+1) = [\mathbf{I} - \mathbf{K}(k+1)]\mathbf{P}(k+1|k). \quad (\text{B.60})$$

B.4.5 Filtro de Kalman estendido

Em muitas aplicações robóticas com fusão de dados de sensores, o sistema a ser modelado falha por ter uma distribuição de ruído não gaussiano ou por não ser linear. Enquanto os erros forem *aproximadamente* gaussianos, o filtro de Kalman pode ser utilizado, embora, provavelmente não será ótimo. Para sistemas não lineares, utiliza-se o *Filtro de Kalman Estendido* (FKE). Isto envolve a linearização da planta, equação B.31 e, se necessário, a linearização da medição (equação B.36) cancelando-se termos de alta ordem da expansão de Taylor (MAYBECK, 1979).

A linearização do modelo de planta envolve o cálculo do jacobiano do modelo de planta $\nabla\mathbf{\Psi}(\hat{\mathbf{x}}(k), \mathbf{u}(k))$ e utilizá-lo como uma estimativa linear de $\mathbf{\Psi}$ no filtro de Kalman. A linearização do modelo de medição envolve o cálculo do jacobiano do modelo de medição $\nabla\mathbf{h}_{st}(\mathbf{x}(k), \mathbf{p}_t)$ e utilizá-lo como estimativa linear de $\mathbf{\Lambda}_E$.

Como exemplo da utilização do FKE, toma-se o exemplo do modelo de planta e modelo

de medição não linear apresentado na seção B.4.3. Para se simplificar a exposição, é assumido que $\mathbf{Q}(k) = \mathbf{Q}$ e $\mathbf{R}_i(k) = \mathbf{R}_i$. Para cada movimento do robô, as seguintes etapas são seguidas:

1. Para se representar o deslocamento do robô, utilizam-se as equações B.31 e B.44. Os parâmetros de controle conhecidos são utilizados para a estimativa da posição do robô no tempo $k + 1$, como sendo:

$$\hat{\mathbf{x}}(k + 1|k) = \begin{bmatrix} x(k) + T(k) \cos[\theta(k)] \\ y(k) + T(k) \text{sen}[\theta(k)] \\ \theta(k) + \Delta\theta(k) \end{bmatrix}. \quad (\text{B.61})$$

2. Uma versão linearizada do modelo de planta é gerado na estimativa corrente do posicionamento do robô $\hat{\mathbf{x}}(k)$ como sendo:

$$\nabla\Psi = \begin{bmatrix} 1 & 0 & -T(k) \text{sen}[\theta(k)] \\ 0 & 1 & T(k) \cos[\theta(k)] \\ 0 & 0 & 1 \end{bmatrix}. \quad (\text{B.62})$$

3. A incerteza do estado é gerada pela atualização da matriz covariância de estado utilizando-se medições obtidas até o tempo k , incluindo-se o mesmo, através da equação:

$$\mathbf{P}(k + 1|k) = \nabla\Psi\mathbf{P}(k)\nabla\Psi^T + \mathbf{Q} \quad (\text{B.63})$$

que é o resultado da linearização da equação B.47.

4. O modelo de sensor é linearizado em torno da estimativa corrente da posição do robô $\hat{\mathbf{x}}(k)$ como sendo:

$$\mathbf{\Lambda}_E = \nabla\mathbf{h}_{st}(\mathbf{x}(k), \mathbf{p}_t). \quad (\text{B.64})$$

Se o sensor tem como alvo um ponto de referência em (x_b, y_b) , então

$$\mathbf{\Lambda}_E = \begin{bmatrix} 2x(k)[x(k) - x_b] \\ 2y(k)[y(k) - y_b] \\ 0 \end{bmatrix}^T. \quad (\text{B.65})$$

5. Utilizando-se este valor de $\mathbf{\Lambda}_E$, o ganho de Kalman, que vem da equação B.49, é computado como

$$\mathbf{K}(k + 1) = \mathbf{P}(k + 1|k)\mathbf{\Lambda}_E^T[\mathbf{\Lambda}_E\mathbf{P}(k + 1|k)\mathbf{\Lambda}_E^T + \mathbf{R}_i]^{-1}. \quad (\text{B.66})$$

6. A inovação, conforme equação B.50, neste caso é calculada como

$$\mathbf{r}(k+1) = \mathbf{z}(k+1) - \mathbf{h}_{st}(\hat{\mathbf{x}}(k+1|k), \mathbf{p}_t) \quad (\text{B.67})$$

7. Agora é possível calcular a estimativa da posição do robô, conforme mostrado na equação B.52, como sendo

$$\hat{\mathbf{x}}(k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{K}(k+1)\mathbf{r}(k+1), \quad (\text{B.68})$$

8. Finalmente, a matriz covariância atualizada, conforme visto na equação B.53, é calculada como

$$\mathbf{P}(k+1) = [\mathbf{I} - \mathbf{K}(k+1)\mathbf{\Lambda}_E]\mathbf{P}(k+1|k), \quad (\text{B.69})$$

Após certo intervalo de tempo, que seja o mais curto possível, as derivadas utilizadas no modelo de linearização devem ser recalculadas através do estado estimado corrente. Isto trás uma deficiência no FKE: se o estado estimado está muito longe do estado atual, a aproximação linear do comportamento do sistema não será suficientemente precisa. Isto pode levar a erros incrementais na estimativa de estado, um fenômeno conhecido como *divergência do filtro* (SORENSEN, 1970). Devido a possibilidade de divergência, os algoritmos do FKE tipicamente computam todas as entradas antes do processamento. Isto envolve a computação da inovação, o teste para se verificar se é muito grande, e as medições rejeitadas que falharam no teste.

B.4.6 Utilizando o Filtro de Kalman na localização de robôs móveis

A idéia básica da localização de robôs móveis é combinar medidas de sensores internos, que fornecem informação sobre o movimento do robô, com sensores externos, que fornecem uma representação do ambiente. No entanto, as informações que os robôs recebem de seus sensores são incertas. No caso de sensores externos as medidas são imprecisas e ruidosas e estão limitadas pelo fato dos sensores, em geral, poderem medir apenas parte do ambiente. Para os sensores internos, o controle dos atuadores, no caso as rodas do robô, é também impreciso. Desta maneira, para se obter a localização de robôs móveis tem-se que fundir medidas de diferentes sensores, obtidas em posicionamento e tempos distintos.

Uma maneira de computar a estimação do posicionamento do robô é através do filtro de Kalman. O filtro também computa um valor de incerteza associado ao posicionamento estimado. O filtro de Kalman realiza a estimação do posicionamento utilizando toda informação

disponível: todas as observações realizadas, que dependem dos sensores empregados, bem como o conhecimento sobre os modelos de movimento e de sensores. A localização utilizando o filtro de Kalman baseia-se na correspondência de marcadores observados com um mapa conhecido *a priori* para atualizar a estimação do posicionamento do robô. Neste contexto, um marcador é uma estrutura do ambiente que pode ser observada pelos sensores do robô e um mapa é um conjunto de marcadores conhecidos no ambiente.

Muitos trabalhos na literatura empregam filtro de Kalman, ou sua versão não linear, o FKE, para localização de robôs móveis. O trabalho de Leonard e Durrant-Whyte (1991) utiliza o FKE para localização de robôs utilizando informações de odômetros e sonares. Trabalhos mais recentes, como o de Tardos et al. (2002), utilizam o filtro para solucionar tarefas de SLAM (*Simultaneous Localization and Mapping*), que é o problema de simultaneamente localizar o robô e construir um mapa do ambiente. Existe ainda a aplicação do filtro de Kalman para localizar um grupo de robôs, como o trabalho de Roumeliotis e Bekey (2002), em que cada robô observa os outros robôs do grupo, estima o posicionamento dos colegas e comunica os posicionamentos estimados.

O filtro de Kalman, então, estima o estado do sistema utilizando os conhecimentos de:

- Modelo da dinâmica do sistema – modelo do movimento do robô;
- Modelo de observação – modelo dos sensores do robô;
- Estatísticas dos ruídos da dinâmica e erros de observação – incerteza associada à odometria e aos sensores de localização do robô;
- Informações das condições iniciais – localização inicial do robô.

O filtro de Kalman opera em um ciclo de predição-emparelhamento-atualização, como mostra a Figura B.9. Na fase de predição, o estado do sistema e as observações são estimados num tempo futuro utilizando o modelo da dinâmica do sistema e o modelo de observação, respectivamente. Na fase de emparelhamento, as observações reais são correspondidas com as observações preditas, selecionando as observações válidas. Por fim, na atualização, a predição do estado é corrigida utilizando as observações válidas.

Um ensaio de simulação utilizando-se o filtro de Kalman para estimativa do posicionamento de robôs móveis, com os resultados para duas trajetórias distintas, é mostrado nas Figuras B.10 e B.11. Como a utilização das técnicas de navegação autônoma para robôs móveis não é o objetivo deste trabalho, esta apresentação a respeito da utilização do filtro

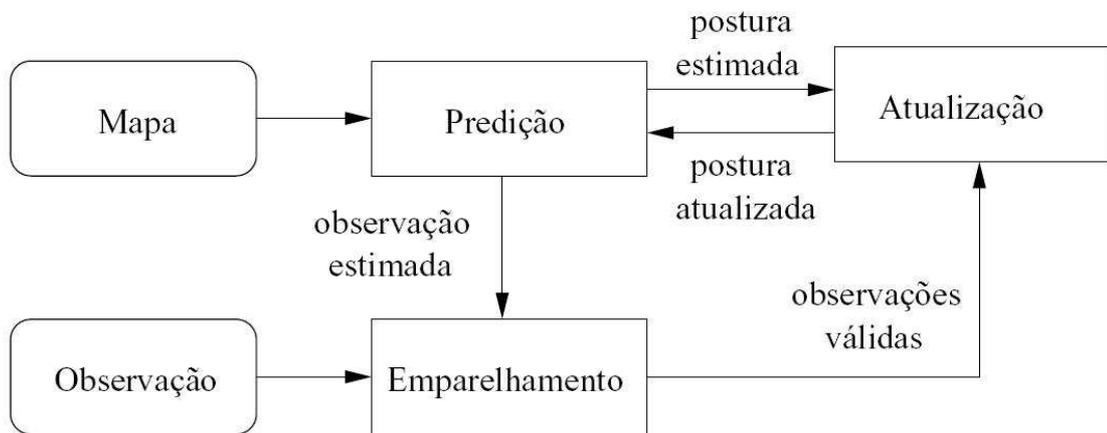


Figura B.9: O filtro de Kalman utilizado na localização de robôs móveis.

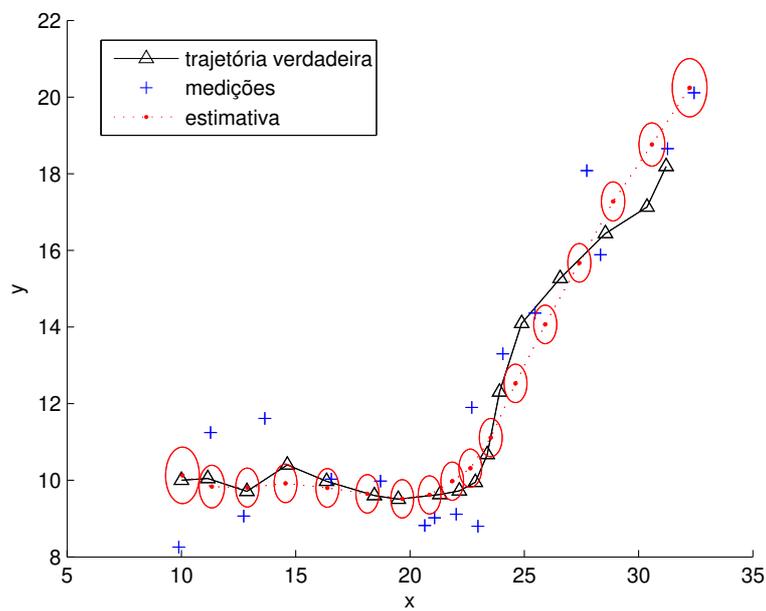


Figura B.10: O filtro de Kalman utilizado na estimativa de localização de robôs móveis.

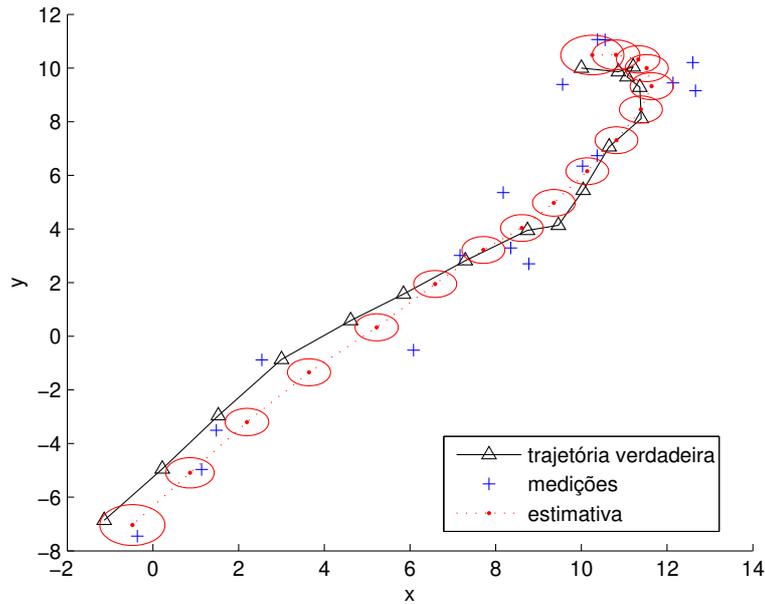


Figura B.11: Simulação utilizando o filtro de Kalman na estimativa de localização de robôs móveis.

de Kalman é superficial, não sendo apresentada em detalhes. Não obstante, devido a sua importância crucial, elas foram aqui introduzidas, haja visto que uma das sugestões para continuação deste trabalho é justamente a implementação de um sistema de navegação autônoma virtual para robôs móveis, utilizando-se destes conceitos.

Os resultados experimentais mostraram que as simulações feitas para localização do robô móvel utilizando-se o filtro de Kalman pode estimar a posição do robô e, ainda, obter valores de estimação de posição próximos da posição real. Estes resultados encorajam a aplicação do algoritmo proposto para localização de robôs reais num sistema de navegação autônoma.

B.5 Mapas

Para os sistemas robóticos móveis autônomos, os mapas representam os elementos estruturais em algum domínio abstrato, onde o robô seja capaz de relacionar sua localização corrente, diretamente através de suas próprias percepções, com respeito ao ambiente.

As características sensoriais ou funcionais de relevância para os seres humanos, com relação ao meio ambiente, pode não ser de similar relevância para os robôs móveis, e vice-versa. Isto sugere que um mapa apropriado para um robô autônomo deva relacionar os tipos de da-

dos coletados pelos sensores do robô com a estrutura do ambiente em que esteja navegando. Desta maneira, pode ser apropriado a indicação de localizações associadas com ecos espúrios de sonares, regiões de muita interferências de RF ou eletromagnéticas, etc. Em geral, estes fatos implicam que a habilidade de executar algum grau de liberdade na construção, atualização e validação autônoma de mapas tem importância fundamental (YAMAUCHI; SCHULTZ; ADAMS, 1998).

Mapas podem ter vários formatos e padrões, mas para relevância dos sistemas robóticos móveis, duas representações específicas são interessantes (KUIPERS; BYUN, 1990):

- **Mapas métricos.** São baseados em quadros com referência absoluta e estimativas numéricas a respeito da posição dos objetos no espaço.
- **Mapas topológicos** (ou mapas relacionais). Representam informações de conexões, tipicamente em forma de gráficos.

Cada uma destas categorias de representação admitem várias especificações e descrições específicas. Além disto, muitas representações reais possuem ambos, componentes topológicos e métricos (THRUN et al., 1998). Em termos da robótica móvel, talvez a forma mais explícita de mapeamento seja o mapa métrico no espaço de configuração (conforme seção B.2.1) cujas possibilidades de movimentações (exceto aquelas regidas pelas restrições não holonômicas, seções 2.2 e B.2.1) são dadas explicitamente. Neste caso, alguns obstáculos, quando transformados dentro do espaço de configuração, na forma de obstáculos do espaço-C, podem assumir formas mais complexas do que no espaço atual. Por exemplo, no espaço de configuração definido por rotações e translações 2D, obstáculos poligonais tornam-se superfícies arredondadas, conforme pode ser visto na seção B.2.1.1.

Mapas topológicos, em contraste com mapas métricos, naturalmente capturam informações qualitativas e rotacionais enquanto dão menos ênfase ao que pode ser irrelevante ou que apresentem detalhes confusos. Como resultado, estes têm conexões muito explícitas com as tarefas e as semânticas do problema.

No sentido de se explorar as vantagens das representações métricas e topológicas, pode ser apropriado considerar-se a construção de uma representação utilizando-se observações de outras representações menos abstratas. Adicionalmente, é natural se considerar as descrições locais antes de inter-relações em larga escala. Isto conduz naturalmente à camadas hierárquicas de representações sucessivas dos dados do mapa, como as cinco apresentadas a seguir (DUDEK, 1996):

- 1.**Sensorial.** Sinais de dados não processados ou transformações sinal-domínio destes sinais.
- 2.**Geométrica.** Objetos 2D ou 3D inferidos a partir dos dados dos sensores.
- 3.**Relacional local.** Relações funcionais, estruturais ou semânticas entre objetos geométricos que estão próximos uns aos outros.
- 4.**Topológica.** As ligações relacionais em larga escala que conectam objetos e localizações através do ambiente com um todo.
- 5.**Semântica.** Rótulos funcionais associados com os componentes do mapa.

Giralt (GIRALT; CHANTILA; VAISSET, 1990) expõe um conjunto descrições que é derivado da representação topológica do ambiente a partir de dados métricos. Em contraste, Levitt (LEVITT; KUIPERS, 1988) e (KUIPERS; BYUN, 1990) consideram a inferência dos dados métricos a partir de representações essencialmente topológicas e semânticas. O seus métodos de exploração, consideram o uso de marcadores topológicos de baixo nível como as observações primitivas básicas de onde a inferência métrica pode ser derivada.

A geração de mapa é uma necessidade para muitas aplicações nos sistemas robóticos móveis. Quase todos ambientes ocupados por pessoas passam por mudanças contínuas. Como resultado, os sistemas robóticos móveis têm que estarem aptos a acomodarem as mudanças nos seus ambientes. Se forem capazes de manter os mapas, devem ser capazes de atualizá-los para se tornarem autônomos.

B.5.1 Mapas sensoriais

Mapas baseados nas leituras diretas de sensores oferecem a possibilidade de associar a representação do ambiente, tão diretamente quanto possível, com os dados dos sensores do robô. A idéia básica é fundir as medições dos sensores com as informações da odometria e então utilizar técnicas para se navegar, com respeito ao mapa.

B.5.1.1 Mapas baseados em imagem

Quando o robô move-se pelo ambiente pode coletar as leituras feitas pelos sensores. Se for assumida odometria perfeita, após algum período, o robô terá coletado um conjunto de

medições

$$[I_i(x_i, y_i, \theta_i)]. \quad (\text{B.70})$$

Se for coletado suficientes I_i , de tal forma que se possa computar uma aproximação contínua para I , qual seja $I(x, y, \theta)$, então pode-se utilizar algum método de navegação com respeito à $I(x, y, \theta)$. A dificuldade com esta abordagem, é que deve-se saber como se amostrar o conjunto de possíveis medições e como se construir o I contínuo a partir das medições $[I_i(x_i, y_i, \theta_i)]$.

Em algumas aplicações, ao invés de se utilizar I para construir um mapa diretamente a partir das leituras dos sensores, o dado I_i pode ser utilizado como medições primárias representadas no mapa. Li (LI et al., 2000) propõe um método para fazer um mapa do ambiente em larga escala baseado nas cenas de rotas, supondo que a relação topológica das rotas nas intersecções é conhecida. A idéia é decompor um gráfico 2D em um número de laços fechados. O robô móvel constrói um gráfico de representação de espaço onde as margens do diagrama correspondem às ruas e onde os nós correspondem às intersecções.

B.5.1.2 Modelos de Markov

Uma alternativa para se computar a posição dos obstáculos no ambiente em conjunto com o posicionamento do robô, é o modelamento das células de ocupação, assim como o posicionamento do robô e o resultado de todas as ações, como distribuições de probabilidade. Nestes casos, a escolha de como se coleta dados, ou mais geralmente, como se comporta num estado particular, pode ser referido como uma *política*. O que é desejável de um dado estado, o qual compreende a ocupação do ambiente assim como o posicionamento do robô, é expresso pela *função utilidade*, que é um valor escalar associado com um estado.

Considerando-se, por exemplo, o caso em que o posicionamento do robô seja dado num ambiente 2D por $\mathbf{p} = (x, y, \theta)$. A probabilidade, ou confiança¹⁰, que o robô tenha um certo posicionamento particular, pode ser modelado por uma grade 3D $\mathcal{B}el(p)$. Uma ação α tem um conjunto de efeitos possíveis $P_\alpha(\mathbf{p})$ mesmo sabendo-se exatamente qual estado p em que o robô se encontra. Para este estado, pode existir uma variedade de erros ou incertezas. Desta forma, após uma certa ação, a grade é atualizada utilizando-se

$$\mathcal{B}el(p) \leftarrow \int P_\alpha(p|p')\mathcal{B}el(p')dp', \quad (\text{B.71})$$

onde $P_\alpha(p|p')$ descreve a probabilidade de uma ação α tomada a partir do estado p' até o estado p .

¹⁰Em inglês: *belief*, de onde vem o símbolo $\mathcal{B}el$.

De maneira análoga, uma observação s tomada a partir dos sensores do robô pode ser interpretada como uma aplicação simples da regra de Bayes,

$$\mathcal{B}el(p) \leftarrow \frac{P(s|p)\mathcal{B}el(p)}{P(s)}, \quad (\text{B.72})$$

onde $P(s|p)$ é a probabilidade da leitura de s ocorrendo no estado p . $P(s)$ é a probabilidade intrínseca da leitura s .

Thrun (THRUN, 1998) implementa uma representação do problema de localização do robô, permitindo que o mesmo estime localmente sua próxima ação, que incrementa sua certeza com respeito ao seu posicionamento. Uma boa estimativa de posicionamento está associada com uma crista aguda na distribuição de posicionamento $\mathcal{B}el(p)$, que pode ser expressado computando-se sua entropia, como a seguir:

$$\mathcal{H} = - \int \mathcal{B}el(p) \log \mathcal{B}el(p) dp. \quad (\text{B.73})$$

Sensoriamento e ação podem então serem selecionados para minimizar a entropia futura esperada:

$$\begin{aligned} E_\alpha[\mathcal{H}] &= E \left\{ \int \mathcal{B}el_{\alpha,s}(p) \log \mathcal{B}el_{\alpha,s}(p) dp \right\} \\ &= - \int \left[\int \mathcal{B}el_{\alpha,s}(p) \log \mathcal{B}el_{\alpha,s}(p) dp \right] P(s) ds \\ &= - \int \int P(s|p) \mathcal{B}el_\alpha(p) \log [P(s|p) \mathcal{B}el_\alpha(p) P(s)^{-1}] dp ds \end{aligned} \quad (\text{B.74})$$

Uma abordagem similar é utilizada para a estimação do mapa, descrita por uma grade de ocupação, que explora os dados coletados dos sensores a partir de uma determinada trajetória (BURGARD et al., 1996)(FOX et al., 1998). Neste caso, um conjunto de medições d tomadas ao longo da trajetória do robô é fornecido *a priori*, em conjunto com uma função $P(o|\mathbf{p}, m)$ que expressa a probabilidade do robô observar o enquanto tem o posicionamento \mathbf{p} no mapa m , que pode ser reformulado como

$$P(o|\mathbf{p}, m) = \frac{P(o|\mathbf{p}, m)P(\mathbf{p}|m)}{\int P(o|\mathbf{p}', m)P(\mathbf{p}'|m)d\mathbf{p}'} \quad (\text{B.75})$$

$$= \eta P(o|\mathbf{p}, m)P(\mathbf{p}|m) \quad (\text{B.76})$$

O objetivo é maximizar a probabilidade do mapa, a partir dos dados observados de um

conjunto de posicionamentos nos instantes de tempo $t = 1 \dots T$:

$$P(m|d) = \int \dots \int P(m|\mathbf{p}^{(1)}, \dots, \mathbf{p}^{(T)}, d) P(\mathbf{p}^{(1)}, \dots, \mathbf{p}^{(T)}|d) d\mathbf{p}^{(1)} \dots d\mathbf{p}^{(T)} \quad (\text{B.77})$$

ou

$$P(m|d) = \int \dots \int \frac{1}{P(d|\mathbf{p}^{(1)}, \dots, \mathbf{p}^{(T)})} \prod_{t=1}^T P(o^{(t)}|m, \mathbf{p}^{(t)}) P(m) \prod_{t=1}^{T-1} P(\mathbf{p}^{(t+1)}|u^{(t)}, \mathbf{p}^{(t)}) d\mathbf{p}^{(1)} \dots d\mathbf{p}^{(T)}, \quad (\text{B.78})$$

que é executado por computação iterativa para o cálculo do posicionamento do robô dado o mapa, e então reestimando-se o mapa, dado uma estimativa atualizada do posicionamento do robô. Este tipo de procedimento foi utilizado por Fox (FOX et al., 1998), dentre outros, para a estimativa de posicionamento de um robô móvel.

B.5.2 Mapas geométricos

Mapas geométricos podem ser descrições eficientes do ambiente, se for assumido que os sensores fornecem dados apropriados e que o ambiente a ser descrito seja adaptável às primitivas de modelamento geométrico a serem utilizadas.

A exploração de ambientes desconhecidos, a construção de mapas geométricos a partir das explorações, e outros problemas correlatos, têm sido estudados extensivamente no campo da geometria computacional (LI et al., 2000)(LEVITT; KUIPERS, 1988)(THRUN et al., 1998). Descrevem uma variedade de capacidades, tais como a procura por um objeto específico ou uma posição alvo, procura por uma rota com propriedades específicas, cobertura de espaços livres, e aprendizado sobre a ocupação do espaço (i.e., mapeamento).

Vários algoritmos tem sido desenvolvidos para a navegação em ambientes com incertezas. Estes algoritmos procuram encontrar a posição do alvo e a posição dos obstáculos no ambiente, no curso da navegação e, desta maneira, são naturalmente classificados como um forma de algoritmo de exploração.

Em muitos casos, uma mapa global do ambiente não esteja disponível quando o robô começa a se mover em direção à posição alvo. Sobre certas circunstâncias, é possível ao dispositivo planejar métodos que podem ser utilizados na presença de incertezas para prover garantia de performance. O algoritmo Bug (LUMELSKY; STEPANOV, 1987) é um exemplo.

Estes tipos de algoritmos são utilizados para o planejamento de trajetória a partir de uma posição inicial até um local alvo com coordenadas conhecidas, assumindo-se que o robô móvel seja pontual, possua restrições holonômicas, tenha odometria perfeita, tenha sensores de contato ideais e memória de armazenamento de dados infinita. Basicamente, o algoritmo opera alternando-se entre dois comportamentos simples: (1) o robô move-se diretamente para a localização alvo e (2) circunavega os obstáculos, se existirem.

B.5.3 Mapas topológicos

Representações geométricas são criadas a partir de dados métricos como sendo o núcleo da representação. Infelizmente, estes dados métricos são, na maioria das vezes, corrompidos pelos erros das medições dos sensores. Uma alternativa para os mapas geométricos são os mapas topológicos, que são baseados em gráficos. As representações topológicas, além de evitarem os erros das medições métricas dos sensores, evitam os custos de armazenamento maciço associados com as representações métricas. Eles têm analogias com as percepções espaciais humanas (PARK; KENDER, 1995) (e.g., quando pede-se alguma informação para alguém na rua, geralmente a resposta vem de alguma forma parecida com: “vire a direita, vá até o fim da rua, passe pelo sinal, vire a esquerda, etc...”).

A chave do relacionamento topológico é alguma representação explícita das conectividades entre as regiões e os objetos. Na sua forma mais pura, pode não conter nenhum dado métrico. Uma representação topológica é baseada numa abstração do ambiente em termos de lugares discretos com conexões de arestas entre eles. Por exemplo, um mapa baseado em gráficos pode ser dado por

$$G = (V, E). \tag{B.79}$$

com um conjunto de N nós ou vértices V e um conjunto de M arestas E que conectam os nós. Normalmente existe relações métricas no gráfico G , pois as arestas têm uma certa dimensão e orientação com respeito aos nós. Os nós são denotados por

$$V = \{v_1, v_2, \dots, v_N\} \tag{B.80}$$

e arestas por

$$E = \{e_1, e_2, \dots, e_M\}, \tag{B.81}$$

sendo que a aresta e_{ij} é dado por

$$e_{ij} = \{v_i, v_j\}. \tag{B.82}$$

Se a ordem de v_i e v_j é significativa, assim como é em modelos que pressupõem que caminhos são unidirecionais, então tem-se um *gráfico direto*. Nota-se que $M < N(N - 1)$ para gráficos sem arestas transitivas (arestas entre nós e elas próprias), enquanto que para *gráficos planares* (gráficos sem nenhuma aresta que se cruzam) tem-se um salto no número de arestas $M < 3N - 1$

O uso de gráficos tem sido explorado por muitos sistemas robóticos para a representação do ambiente. O exemplo a seguir, a partir de (ELLEPOLA; KOVESI, 1997) é representativo. O ambiente do robô é modelado como um gráfico cujos nós correspondem a marcadores visuais colocados no teto. Cada marcador é único e também define uma orientação local, e são localizados utilizando-se uma câmara montada no robô que aponta para cima. Uma representação baseada em gráfico do ambiente amostrado é ilustrada na Figura B.12. Cada nó corresponde a um único marcador, enquanto arestas correspondem aos caminhos conhecidos entre os marcadores.

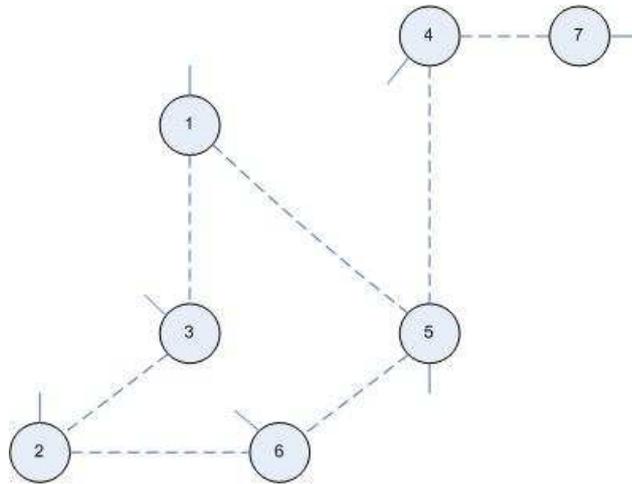


Figura B.12: Representação baseada em gráfico de um ambiente do robô móvel.

Cada aresta no gráfico é rotulada com a distância que é necessária para se alcançar o próximo marcador, que também são rotulados para mostrarem as direções com respeito à orientação local definidos pelos marcadores. O robô não tem um discernimento real da relação geométrica entre os locais do ambiente. Os locais são apenas unidos pelas suas representações topológicas.

B.6 Considerações finais

A navegação de robôs móveis requer técnicas que forneçam os meios para que um robô autônomo se mova de forma segura de um local a outro do ambiente. Encontrar um caminho de uma determinada posição até um destino é um dos problemas fundamentais da robótica móvel autônoma. Destarte, este capítulo introduziu uma revisão bibliográfica a respeito de algumas das técnicas de navegação mais importantes para os sistemas robóticos móveis. Descrevem-se, dentre outras, as técnicas de medição de distância, planejamento de trajetória e técnicas de triangulação, algumas delas utilizadas no simulador virtual do robô móvel, capítulo 5 e no bloco gerador de trajetória, seção 5.2.

Apêndice C – Sistemas de transmissão digital

Neste capítulo são apresentadas algumas das técnicas de codificação de dados utilizadas nas transmissões digitais.

C.1 Codificação para transmissão digital

A codificação de dados digitais é utilizada para que uma informação digital seja enviada através de um canal de transmissão digital. Existem várias formas e técnicas de se codificar o sinal digital (normalmente um sinal binário) para que seja transmitido. A seguir, são apresentadas de uma maneira simplificada, algumas das técnicas mais tradicionais para codificação de dados digitais.

C.1.1 Codificação NRZ

A forma mais simples de codificação consiste em associar um nível de tensão a cada bit. Esta codificação é conhecida por NRZ-L (*Nonreturn to zero - Level*), um bit 1 será codificado sob a forma de uma tensão elevada e um bit 0 sob a forma de uma tensão baixa. Existem mais duas codificações NRZ, nestes casos o sinal não retorna ao nível zero após uma transição provocada pelos dados a serem transmitidos. A codificação NRZ-M (NRZ - *Mark*) produz uma transição de nível sempre que surge um bit 1. A codificação NRZ-S (NRZ - *Space*) produz uma transição de nível sempre que surge um bit 0. A frequência máxima gerada é igual a metade da taxa de transmissão (2 bit/Hz).

C.1.2 Codificação RZ

A codificação RZ (*Return-zero*) difere das anteriores pelo fato do nível de tensão retornar sempre ao nível zero após uma transição provocada pelos dados a serem transmitidos (no

meio da transmissão do bit). Geralmente um bit 1 é representado por um nível elevado, mas no meio da transmissão do bit o nível retorna a zero. Devido a este fato a frequência máxima gerada é o dobro da anterior, sendo igual à taxa de transmissão (1 bit/Hz).

O grande problema das codificações NRZ e RZ é que geram uma componente contínua (a média do sinal não é zero) este fato dificulta o isolamento entre emissor e receptor. Mais grave ainda é que certas seqüências de bit a 1 ou 0 produzem um sinal sem qualquer variação. O receptor deve estar sincronizado com o emissor para que possa ler os bits no ponto correto. Sem transições de nível por períodos longos, o receptor tende a desviar-se e efetuar a leitura fora do ponto correto, originando erros. A Figura C.1 apresenta exemplos de codificações NRZ e RZ.

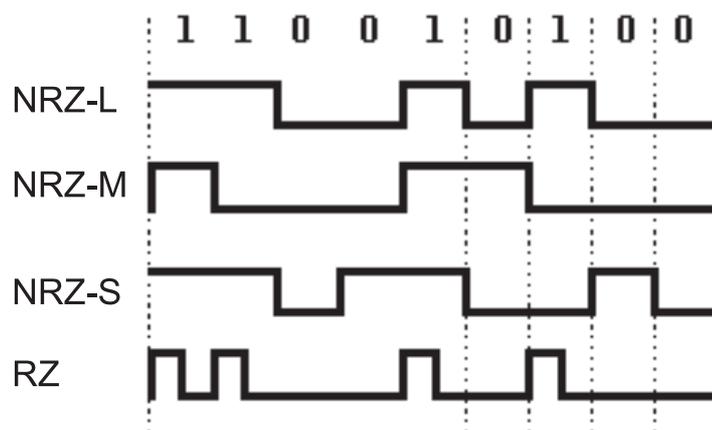


Figura C.1: Exemplo das codificações NRZ e RZ.

C.1.3 Codificações bifásicas

As codificações bifásicas caracterizam-se por transições de nível em todos os bits. O ponto de transição vai depender dos dados a serem transmitidos. Na codificação bifásica de nível conhecida por "*Manchester*", os bits 1 produzem uma transição de nível elevado para baixo, no meio do bit, e os bits 0 produzem transições de nível baixo para nível elevado também no meio do bit. No início de cada bit são produzidas as transições de nível necessárias para manter a codificação coerente:

- se o bit é um e o nível está baixo;
- se o bit é zero e o nível está alto.

A variante *Manchester diferencial* produz sempre uma transição de nível no meio dos bits e uma transição no início dos bits zero. As variantes bifásicas “*mark*” e “*space*” provocam sempre uma transição de nível entre os bits. No caso bifásica-M os bits 1 provocam uma transição no meio do bit. No caso de bifásica-S os bits 0 provocam uma transição no meio do bit. A Figura C.2 apresenta exemplos de codificações bifásicas.

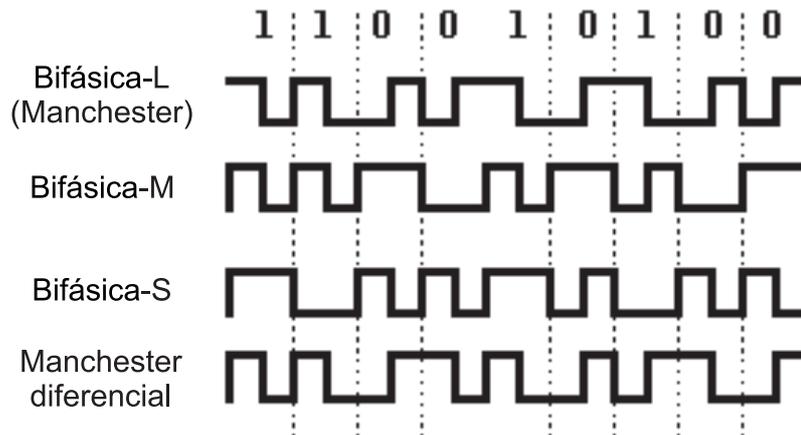


Figura C.2: Exemplo das codificações bifásicas.

As vantagens das codificações bifásicas relativamente aos métodos NRZ e RZ são evidentes, a componente continua do sinal é muito menor e a existência de transições de nível em todos os bits torna a sincronização muito simples. Outra vantagem das codificações bifásicas é o fato de algumas transições de nível serem pré-determinadas facilitando a detecção de erros. Como facilmente se observa na Figura C.2, a frequência máxima gerada é igual à taxa de transmissão (1 bit/Hz).

A codificação de “*Miller*” é um pouco mais complexa. Seu objetivo é manter uma sincronização facilitada, mas sem aumentar a largura de banda. Os bits 1 provocam uma transição de nível no meio do bit. Entre dois bits zero consecutivos produz-se igualmente uma transição de nível. A Figura C.3 ilustra esta codificação.

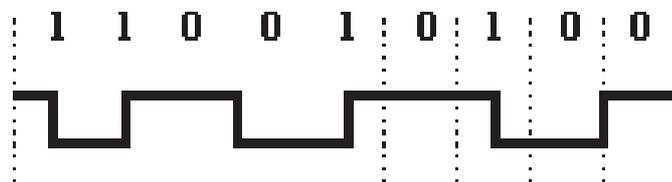


Figura C.3: Exemplo da codificações bifásica de *Miller*.

Com este método a frequência base máxima gerada nunca é superior a metade da taxa de transmissão (2 bit/Hz).

Todos os métodos descritos até aqui são unipolares, isto é apenas assumem uma polaridade (e.g. são sempre positivos). Existem técnicas polares em que o sinal assume também valores negativos, neste caso a codificação deixa de ser binária e o sinal passa a ter 3 níveis distintos. O exemplo mais corrente é a codificação bipolar simples ou AMI (*Alternate Mark Inversion*), em que os bits 1 são representados por impulsos com metade da duração do bit, com sentidos alternados. A Figura C.4 ilustra a codificação AMI.

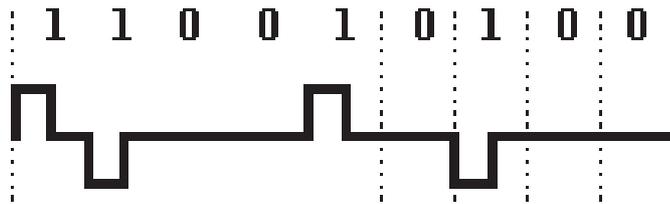


Figura C.4: Exemplo da codificações bipolar AMI.

A frequência máxima gerada é metade da taxa de transmissão (2 bit/Hz), não possui componente contínua e a alternância de sentidos facilita a detecção de erros, contudo uma sequência de zeros produz um sinal nulo que dificulta a sincronização.

C.2 Modulação em telecomunicações

A modulação, em telecomunicações, é a modificação de um sinal eletromagnético inicialmente gerado, antes de ser irradiado, de forma que esta transporte informação sobre uma onda portadora, normalmente uma onda senoidal. O dispositivo que realiza a modulação é chamado modulador. Basicamente, a modulação consiste em fazer com que um parâmetro da onda portadora mude de valor de acordo com a variação do sinal modulado, que é a informação que se deseja transmitir. Dependendo do parâmetro sobre o qual se atue, temos as seguintes tipos de modulação:

- Modulação em amplitude (AM);
- Modulação em fase (PM);
- Modulação em frequência (FM);
- Modulação por divisão ortogonal de frequência (OFDM).

Também se empregam técnicas de modulação por pulsos, entre elas:

- Modulação por pulso codificado (PCM);
- Modulação por largura de pulso (PWM);
- Modulação por amplitude de pulso (PAM);
- Modulação por posição de pulso (PPM).

C.2.1 Modulação em amplitude

A modulação em amplitude, ou simplesmente AM¹, é a forma de modulação em que a amplitude de um sinal senoidal, chamado portadora, varia em função do sinal de interesse, que é o sinal modulador. A frequência e a fase da portadora são mantidas constantes. Matematicamente, é uma aplicação direta da propriedade de deslocamentos em frequências da transformada de Fourier, assim como da propriedade da convolução. A Figura C.5 ilustra o formato da onda da portadora, o sinal de baixa frequência a ser transmitido e o sinal modulado final.

Existem, basicamente, quatro métodos de modulação em amplitude:

- Modulação em banda lateral dupla (DSB);
- Modulação em banda lateral única (SSB);
- Modulação de banda lateral vestigial (VSB, ou VSB-AM);
- Modulação de amplitude em quadratura (QAM).

C.2.1.1 Modulação AM-ASK

A modulação por chaveamento de amplitude (ASK²), pode ser produzido empregando-se as mesmas técnicas da modulação AM-DSB. As principais características da modulação ASK são:

- Facilidade de modular e demodular;

¹Do inglês: *Amplitude Modulation* - Amplitude Modulada.

²Do inglês: *Amplitude Shift Keying*

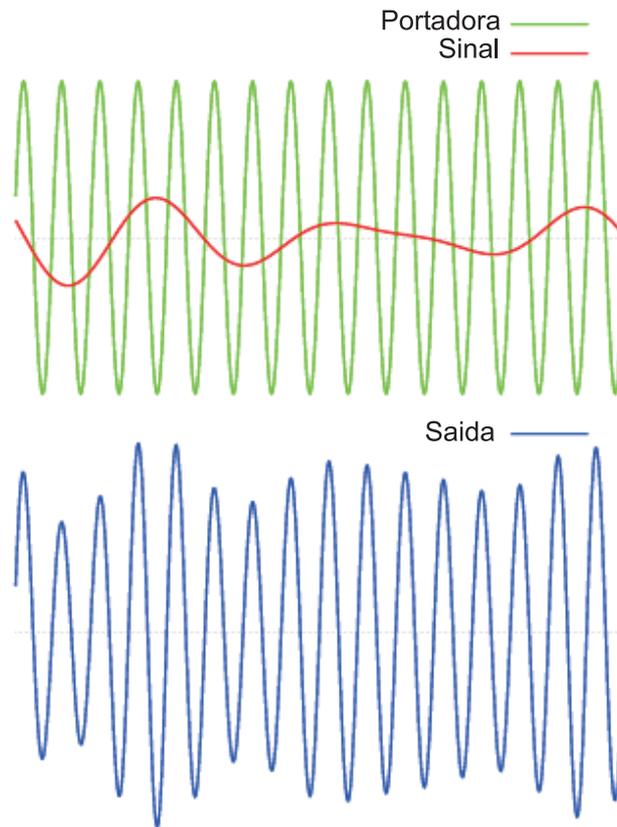


Figura C.5: Modulação em amplitude.

- Pequena largura de faixa;
- Baixa imunidade a ruídos.

Devido a essas características, a modulação por chaveamento de amplitude é indicada nas situações em que exista pouco ruído para interferir na recepção do sinal ou quando o baixo custo é essencial. A modulação ASK é utilizada em aplicações, tais como:

- Transmissão via fibra óptica, onde não existe ruído para interferir na recepção do sinal;
- Transmissão de dados por infravermelho, como os usados em algumas calculadoras, computadores, aparelhos de telefone celular, computadores, etc.;
- Controle remoto por meio de raios infravermelhos, como os usados em aparelhos de TV;
- Controle remoto por meio de RF, como os usados para ligar e desligar alarmes de carros, residências ou abrir portões.

O sinal ASK divide-se em:

1. se o sinal for binário, variando-se dois níveis (0, *space* e 1, *mark*) teremos o ASK binário ou BASK. A Figura C.8 ilustra um exemplo de modulação BASK.

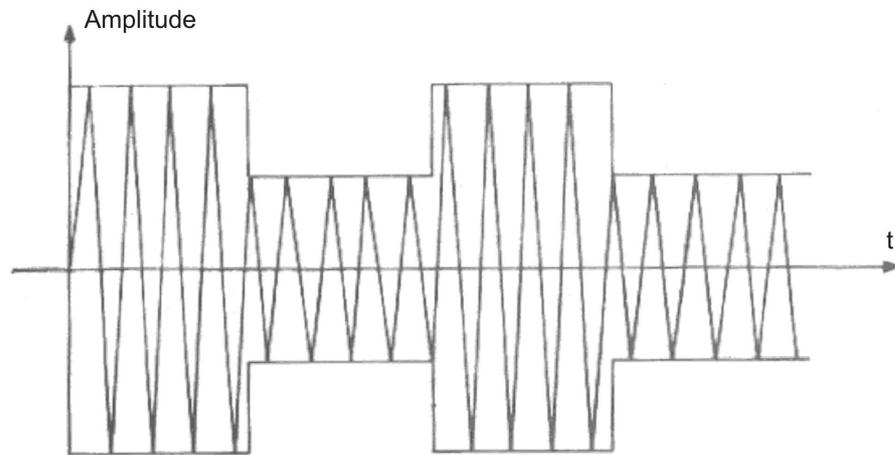


Figura C.6: Modulação BASK.

2. Se o sinal tiver m níveis, sinal multinível, teremos o ASK multinível, também chamado MASK. A Figura C.7 ilustra um exemplo de modulação MASK.

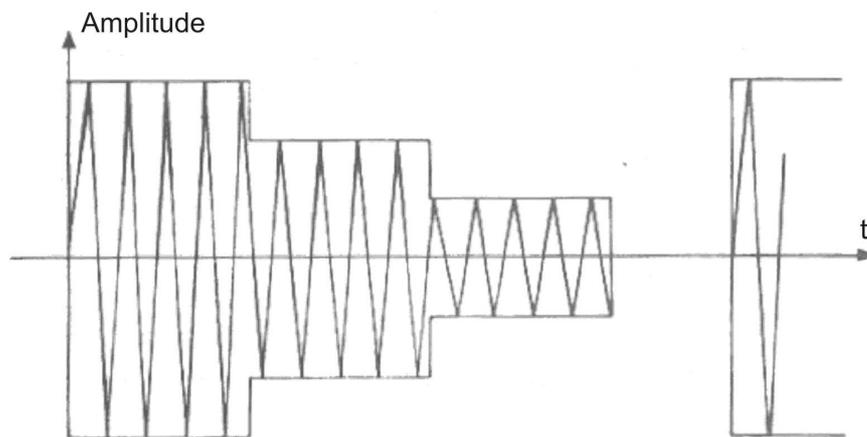


Figura C.7: Modulação MASK.

No caso particular do sinal BASK em que um dos níveis é zero, o sinal produzido equivale a senóide interrompida e por isso é ainda designado por chaveamento liga-desliga (OOK³). A Figura C.8 ilustra um exemplo de modulação OOK.

³Do inglês: *On-Off Keying*

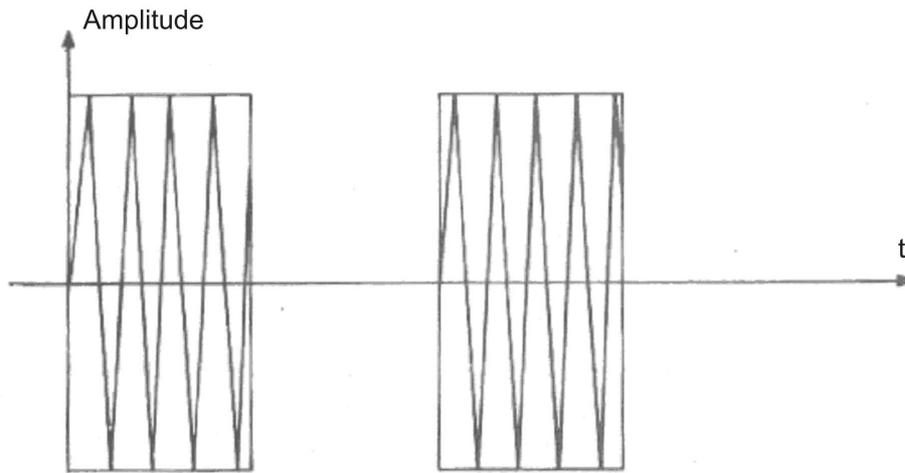


Figura C.8: Modulação OOK.

O sinal OOK equivale simplesmente a uma senóide interrompida, acompanhando o sinal digital. O sinal OOK é o mais antigo e o mais simples dos métodos de modulação por sinal digital. Dispondo-se apenas de um oscilador e de estágios amplificadores de potência, a manipulação apenas atua cortando o sinal quando desejado.

Apêndice D – Proposta de implementação de um sistema de navegação autônoma

O sistema de comunicação entre o robô móvel e os marcadores é descrito a seguir. O robô móvel, e cada um dos marcadores, possuem um módulo de controle e recepção dos códigos de endereço e um módulo de transmissão. O protocolo de comunicação entre o sistema de controle embarcado, localizado no robô móvel, e os marcadores ativos, que estão localizados em pontos estratégicos no ambiente, é composto de um quadro formado por cinco códigos quaternários.

D.1 Protocolo de comunicação

O diagrama de tempo mostrado na Figura D.1 ilustra como é formada cada um dos códigos em função do sinal de clock.

Cada $1/2$ período do clock corresponde a um tempo de $896 \mu s$. Cada código têm um período de 8 ciclos de clock, ou seja $14,336 \text{ ms}$. A Tabela D.1 ilustra de uma maneira lógica a formação de cada código.

Tabela D.1: Formação lógica dos códigos.

Código	Seqüência lógica
$C0$	$A_C + B_L + A_C + B_L$
$C1$	$A_L + B_C + A_L + B_C$
$C2$	$A_L + B_C + A_C + B_L$
$C3$	$A_C + B_L + A_L + B_C$

Cada código é formado por uma seqüência de sinais lógicos, cada qual com um determinado período. A Tabela D.2 mostra como é formado cada sinal lógico de cada código.

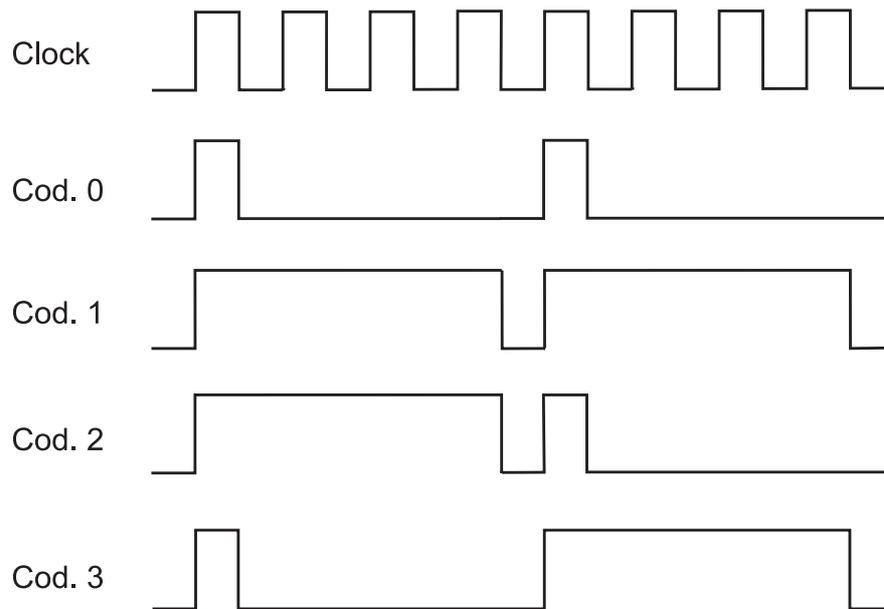


Figura D.1: Códigos do protocolo de comunicação.

Tabela D.2: Formação lógica dos códigos.

Sinal Lógico	Significado	Duração	Tempo
A_C	Nível lógico "1" de duração curta	$\frac{1}{2}$ clock	0,896 ms
A_L	Nível lógico "1" de duração longa	$3\frac{1}{2}$ clock	6,272 ms
B_C	Nível lógico "0" de duração curta	$\frac{1}{2}$ clock	0,896 ms
B_L	Nível lógico "0" de duração longa	$3\frac{1}{2}$ clock	6,272 ms

A idéia é montar um codificador quaternário utilizando-se níveis lógicos binários, associados de tal maneira que os níveis lógicos se alternem e que o período total de cada código seja o mesmo. Conforme vê-se na seção C.1, a codificação aqui proposta e implementada foi concebida visando a minimização de erros, tais como a transmissão em que a um mesmo nível de sinal é transmitido sem transições de nível por períodos longos. Neste caso, o receptor tende a desviar-se e efetuar a leitura fora do ponto correto, originando erros. Desta maneira, a transmissão por RF dos códigos fica bastante robusta e confiável, praticamente extinguindo erros de decodificação do sinal dentro a área de alcance do sistema.

D.2 O quadro de comunicação

O quadro de comunicação entre o robô móvel e os marcadores é composto por cinco códigos quaternários. A Figura D.2 ilustra um exemplo de um quadro do protocolo de comunicação. Como cada código tem um período de 14,336 ms, o quadro todo tem tempo de transmissão de 71,68 ms. O número máximo de combinações possíveis é dado por

C2	C0	C1	C3	C0
----	----	----	----	----

Figura D.2: Quadro do protocolo de comunicação.

$$N_e = 4^5 = 1024. \quad (\text{D.1})$$

Cada marcador possui um endereço fixo, composto por cinco códigos. Desta maneira, o sistema suporta até 1024 marcadores com seus respectivos endereços individuais.

D.2.1 O link de RF

O sinal codificado é transmitido em RF modulado pela técnica BASK-OOK, como pode ser visto na seção C.2.1.1. A frequência do sinal da portadora está na faixa de UHF (*Ultra high frequency*) e é de 330 MHz.

O link de RF é feito num canal *half-duplex* entre o robô móvel e cada um dos marcadores. O sistema de controle do robô móvel é programado previamente com o número e o endereço de cada um dos marcadores. A Figura D.3 ilustra um exemplo de configuração física do sistema de comunicação entre o robô móvel e os marcadores.

D.2.2 O módulo de transmissão-recepção

O módulo de controle embarcado de transmissão e de recepção dos códigos de endereços é um sistema composto por um microcontrolador 16F629 da família PIC¹ operando numa frequência de clock de 4 MHz. Este sistema pode funcionar de maneira autônoma, sendo programado com a quantidade e os endereços dos marcadores do ambiente. A Figura D.4 mostra o diagrama de blocos do sistema implementado.

¹Fabricado pela Microchip Corp.

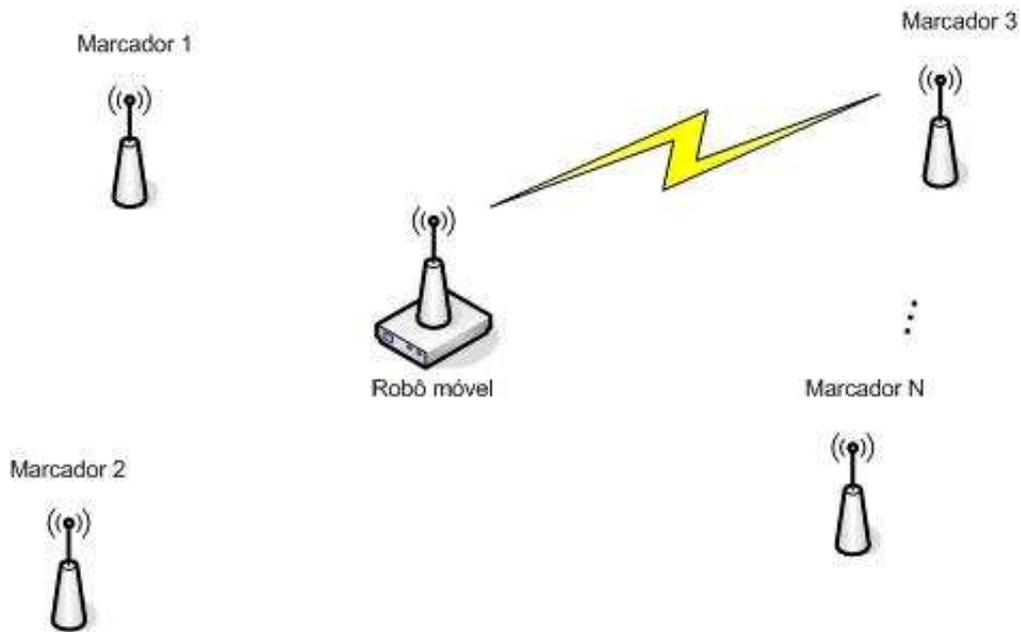


Figura D.3: Comunicação entre o robô móvel e os marcadores.

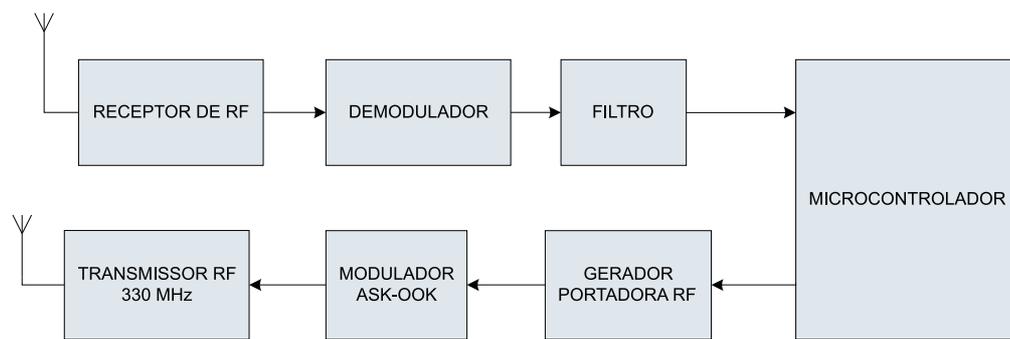


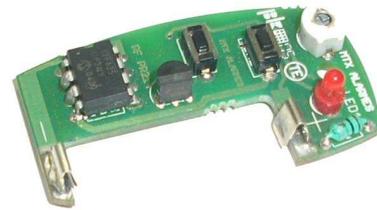
Figura D.4: Módulo de controle embarcado de transmissão-recepção.

A Figura D.5 ilustra o módulo de controle embarcado com os sistemas de recepção e transmissão do sinal de RF. O módulo de transmissão (Figura D.5(b)) pode funcionar de maneira autônoma assíncrona, emitindo um quadro de código de endereço num certo período de tempo pré-determinado, ou de maneira síncrona comandada pelo módulo de controle e de recepção (Figura D.5(a)). No primeiro caso, utiliza-se uma bateria de 12V modelo A23 que permite autonomia de mais de 3 meses de uso contínuo, devido ao baixo consumo de energia do sistema por se tratar de microcontrolador com tecnologia de nanowatt. No segundo caso, a alimentação e o comando da transmissão é feita pelo módulo de controle de recepção, ilustrado na Figura D.5(a).

Tanto o robô móvel, como cada um dos marcadores, possuem um sistema de um sistema



(a) Módulo de recepção



(b) Módulo de transmissão

Figura D.5: Módulo de controle embarcado para transmissão-recepção do sinal de RF.

de controle de transmissão-recepção composto pelo módulo de recepção e controle e pelo módulo de transmissão. Como o objetivo do nosso sistema é prover uma triangulação entre o posicionamento do robô móvel e os marcadores, os módulos de transmissão trabalham de maneira síncrona.

Supondo-se que o módulo de controle de recepção-transmissão do robô móvel tenha sido previamente carregado com a quantidade de marcadores existentes no ambiente e com seus respectivos códigos de endereços. O funcionamento do sistema segue o seguinte procedimento:

- 1.O robô móvel emite um quadro-código de endereço para o primeiro marcador. Neste instante envia um sinal de controle ao nível de processamento central para que seja disparado um contador de tempo. O robô então, aguarda o retorno do sinal. Este retorno deve ocorrer em até 100 ms.
- 2.Se o sinal for retornado, significa que o marcador reconheceu o código e reenviou o mesmo código de volta. Neste instante é enviado um sinal ao nível de processamento central para o cálculo do tempo de atraso do retorno do sinal.
- 3.Se o sinal não foi retornado, significa que o marcador está fora da área de alcance ou ocorreu algum erro de transmissão-recepção.
- 4.Volta-se ao primeiro item e repete-se o processo indefinidamente.

A distância entre o robô móvel e um certo marcador m é calculada com base no tempo de atraso na recepção do código transmitido. O tempo total entre o final da transmissão do

código, enviado pelo robô, e a recepção do mesmo código, enviado pelo marcador, pode ser dado por

$$T = t_{is} + t_{pm} + t_{rs} + t_q + t_{pr}, \quad (\text{D.2})$$

onde t_{is} é o tempo que o sinal leva para sair do robô e chegar no marcador, t_{pm} é o tempo de processamento do sinal pelo marcador, t_{rs} é o tempo de retorno do sinal, t_q é o período do quadro de código e t_{pr} é o tempo de processamento do sinal de retorno recebido pelo robô móvel.

Sabendo-se que a velocidade do sinal de RF no ar pode ser aproximada por 0,3m/ns e considerando-se que a velocidade linear do robô é pequena de tal forma a poder considerar o deslocamento do robô como sendo zero durante o tempo T , então, a distância em metros entre o robô móvel e o marcador m pode ser dada por:

$$d = \frac{0,3(t_{is} + t_{rs})}{2}, \quad (\text{D.3})$$

onde t_{is} e t_{rs} são dados em ns.

O tempo T é calculado através de um temporizador de 64 bits do kit de desenvolvimento da Texas Instruments TMDXDSK6455 com microcontrolador DSP de ponto fixo TMS320C6455 que trabalha com frequência de clock de 1 GHz. Os tempos t_{pm} e t_{pr} são determinados empiricamente e $t_q = 71,68$ ms. Desta forma, a distância d pode ser dada por

$$d = \frac{0,3(T - t_{pm} - t_{rm} - t_q)}{2}. \quad (\text{D.4})$$

O Algoritmo 1 ilustra o método do cálculo da distância d através do ToF^2 do sinal de RF.

²Do inglês: *Time of Flight* - tempo de viagem do sinal.

Algoritmo 1 Método do cálculo da distância d através do ToF do sinal de RF.

Entrada: O robô móvel é inicializado com o número total de marcadores N_m e com seus respectivos códigos de endereços.

Saída: A distância entre o robô e o marcador m com base no tempo de atraso do sinal de RF.

while O sistema estiver ativo **do**

for $n = 1$ até $n = N_m$ **do**

 Transmitir um quadro-código para o marcador n ;

 Iniciar o temporizador para o cálculo do tempo T ;

if O sinal retornou .AND. $t < 100$ ms **then**

 Parar o contador e calcular o tempo T ;

 Calcular a distância $d = \frac{0,3(T-t_{pm}-t_{rm}-t_q)}{2}$;

else

 Ocorreu um erro na transmissão/recepção do sinal de RF;

 Tentar próximo marcador;

end if

end for

end while

FIM

Anexo A – Listagem da rotina de geração de trajetória

A rotina de geração de trajetória foi desenvolvida em linguagem *script* do MatLab[®]. Abaixo segue o programa fonte na sua totalidade.

```
%Função geradora de trajetória para o robô móvel não holonômico
%Entradas: Xtraj, Ytraj, R, ds
%(Xtraj(i), Ytraj(i)) indicam pontos da trajetoria
%R(i) é o raio de transições entre trajetórias
%ds(i) tamanho do deslocamento na trajetória

inicializacao_3

tj2 figure; hold on; for i=1:4
    plot(Xtraj(i),Ytraj(i),'go','markerfacecolor','g');
end

Xres = [];
Yres = [];
R_cur = [];
Temp = [];
P1 = [ 1, -1, -1];
P2 = [ 1,  1,  1];
P3 = [-1,  1,  1];
P4 = [-1, -1, -1];

Xi = Xtraj(1); X_inic = Xi; Yi = Ytraj(1); Y_inic = Yi;
```

```

%tet(1) = atan2(Ytraj(2) - Ytraj(1), Xtraj(2) - Xtraj(1));
for ii=1:3
    tet(ii) = atan2(Ytraj(ii+1) - Ytraj(ii), Xtraj(ii+1) - Xtraj(ii));
    if tet(ii) < 0
        tet(ii) = tet(ii) + 2*pi;
    end
end
%tet(2)= atan2(Ytraj(3) - Ytraj(2), Xtraj(3) - Xtraj(2));
%tet(3)= atan2(Ytraj(4) - Ytraj(3), Xtraj(4) - Xtraj(3));
tet2=[0 tet(1) tet(1) tet(2) tet(2) tet(3)];
tet2=tet2';
teti=tet(1);
Teta_inic = teti;
Xreta(1)=Xi;
Yreta(1)=Yi;
rtcv(1)=10000;
R_c = R;
cont1=2;
cont2=2;
hold on;

for nTraj = 1 : size(Xtraj, 2) - 1;

    % Ponto de intersecção das duas retas.
    Xe = Xtraj(nTraj+1);
    Ye = Ytraj(nTraj+1);

    % Cálculo dos centros das circunferencias 1, 2, 3, 4.
    tan_teti = tan(teti);
    sec_teti = sec(teti);
    tan2_teti = tan_teti^2;

    if nTraj < size(Xtraj, 2) - 1

        Xf = Xtraj(nTraj+2);

```

```

Yf = Ytraj(nTraj+2);
tetf = atan2(Ytraj(nTraj + 2) - Ytraj(nTraj + 1), ...
    Xtraj(nTraj + 2) - Xtraj(nTraj + 1));
raio = R(nTraj);

tan_tetf = tan(tetf);
sec_tetf = sec(tetf);
tan2_tetf = tan_tetf^2;

% Identifica as inclinações das trajetórias.
teta(1) = atan2(Yi - Ye, Xi - Xe);
teta(2) = atan2(Yf - Ye, Xf - Xe);
teta = sort(teta);

% Desenho das trajetórias.
XI = [Xi, Xe];
YI = Yi - (Xi - XI) * tan_teti;
XF = [Xf, Xe];
YF = Yf - (Xf - XF) * tan_tetf;
line(XI, YI);
line(XF, YF);

for i = 1 : 4

    if i==1; Va = P1; disp('solução No 1'); end
    if i==2; Va = P2; disp('solução No 2'); end
    if i==3; Va = P3; disp('solução No 3'); end
    if i==4; Va = P4; disp('solução No 4'); end

% Encontra o PontoTangente.
% Centro da Circunferencia: calculo da reta - coeficientes.
X(i) = (raio * (Va(1) * sec_tetf + Va(2) * sec_teti)...
    + Yi - Yf - Xi * tan_teti + Xf * tan_tetf)...
    / (tan_tetf - tan_teti);
Y(i) = Va(3) * raio * sec_teti + X(i) * tan_teti...

```

```

    + Yi - Xi * tan_teti;
a = 1 + tan2_teti;
b = 2 * Yi * tan_teti - 2 * X(i) - 2 * Y(i) * tan_teti...
    - 2 * Xi * tan2_teti;
c = Yi*Yi - 2*Yi*Y(i) - 2*Yi*Xi*tan_teti...
    + 2*Y(i)*Xi*tan_teti + Xi*Xi*tan2_teti...
    - raio^2 + X(i)*X(i) + Y(i)*Y(i) ;

% As raizes da equação do segundo grau, que são iguais,
%representam o X do ponto onde a reta tangencia
%a circunferência.
Delt1(i) = b^2 - 4 * a * c;
X1 = - b/(2*a);
AUX = (raio^2 - (X1 - X(i)) * (X1 - X(i)));
Yti(i) = Yi - (Xi - X1)*tan_teti;
Xti(i) = X1;

% Reta.
a = 1 + tan2_tetf;
b = 2 * (Yf * tan_tetf - (X(i) + Y(i) * tan_tetf...
    + Xf * tan2_tetf));
c = Yf^2 - 2* (Yf*Y(i) + Yf*Xf*tan_tetf -Y(i)*Xf*tan_tetf)...
    + (Xf*tan_tetf)^2 - raio^2 + X(i)^2 + Y(i)^2 ;
Delt2(i) = b^2 - 4 * a * c;
X1 = -b/(2*a);
AUX = ( raio^2 - (X1 - X(i)) * ( X1 - X(i)));
Ytf(i) = Yf - (Xf - X1)*tan_tetf;
Xtf(i) = X1;

% Saidas: Xti, Yti, Xtf, Ytf.
fprintf(1,'Delt1 = %2.2f \n',Delt1(i))
fprintf(1,'Delt2 = %2.2f \n',Delt2(i))
disp ('posição inicial')
fprintf(1,'Xti = %5.3f mm\n',Xti(i))
fprintf(1,'Yti = %5.3f mm\n',Yti(i))

```

```

disp ('posição final')
fprintf(1,'Xtf = %5.3f mm\n',Xtf(i))
fprintf(1,'Ytf = %5.3f mm\n\n',Ytf(i))

% Escolha da circunferencia certa para a mudança de trajetoria.
% Identifica a inclinação a partir do ponto de intersecção das
% trajetórias.
teta_circ = atan2(Y(i) - Ye, X(i) - Xe);

if teta(1)<teta_circ & teta_circ<teta(2)
    XC = [min(Xti(i), Xtf(i)): .1: max(Xti(i), Xtf(i))];
    YCp = sqrt(raio^2 - (XC - X(i)).^2) + Y(i);
    YCn = -sqrt(raio^2 - (XC - X(i)).^2) + Y(i);
    line(XC, YCp, 'Color','k','LineWidth',1);
    line(XC, YCn, 'Color','k','LineWidth',1);
    XC = [Xti(i), Xtf(i)];
    YC = [Yti(i), Ytf(i)];
    line(XC, YC, 'Color','Y','LineWidth',1);
    Xreta(cont1)=Xti(i);
    Yreta(cont1)=Yti(i);
    cont1=cont1+1;
    Xreta(cont1)=Xtf(i);
    Yreta(cont1)=Ytf(i);
    cont1=cont1+1;
    % Prepara a continuação da trajetoria (proxima reta)
    break;
end
end
else
    Xti(i) = Xe;
    Yti(i) = Ye;
end

% Define o comprimento dos dois segmentos (trajetória reta e trajetória
% curva).

```

```

% Calcula o comprimento da trajetória reta.
traj_reta = sqrt((Xti(i) - Xi)^2 + (Yti(i) - Yi)^2);
% Calcula o comprimento de cada passo de acordo com o número de passos.
n_passos_traj = traj_reta / ds(i);
% Comprimento total da trajetória.
dx = (Xti(i) - Xi) / n_passos_traj;

rtcv(cont2)=10000;
cont2=cont2+1;
for x = Xi : dx : Xti(i)
    Xres = [Xres x];
    Yres = [Yres Yi-(Xi-x)*tan_teti];
    R_cur = [R_cur 10000];      %reta
end

if nTraj < size(Xtraj, 2) - 1

    % Calcula o ângulo da parte da circunferencia utilizada
    %na trajetória.
    ang_circ_ti = atan2(Yti(i) - Y(i), Xti(i) - X(i));
    ang_circ_tf = atan2(Ytf(i) - Y(i), Xtf(i) - X(i));
    if ang_circ_tf < 0
        rr = -raio;
    else
        rr = raio;
    end
    rtcv(cont2)=rr;
    cont2=cont2+1;
    % Calcula comprimento do arco.
    perim = raio * abs(ang_circ_ti - ang_circ_tf);
    % Calcula o angulo necessário entre cada passo
    n_passos_traj_curva = perim/ds(i);

    dang = (ang_circ_tf - ang_circ_ti) / n_passos_traj_curva;
    for ang = ang_circ_ti+dang : dang : ang_circ_tf

```

```

        Xres = [Xres X(i)+raio*cos(ang)];
        Yres = [Yres Y(i)+raio*sin(ang)];
        R_cur = [R_cur rr];
    end
    Xi = Xtf(i);
    Yi = Ytf(i);
    teti = tetf;
end
end

Xreta(cont1)=Xtraj(size(Xtraj, 2));
Yreta(cont1)=Ytraj(size(Ytraj, 2));
traj1 = [Xreta' Yreta' rtcv'];
traj1(:,4)=tet2;

Temp = [1:size(R_cur,2)];
Temp = Temp ./ 100;
Temp2 = Temp';
R_cur = R_cur';
Vet_f = [Temp2 R_cur];

X_f = [Temp2 Xres'];
xx=[Temp; Xres];
save('eixo_X.mat','xx')

Y_f = [Temp2 Yres'];
yy=[Temp; Yres];
save('eixo_Y.mat','yy')

trr1=[Temp; Xres; Yres];
save('trajr1.mat','trr1')

% visualizaçao
for j = 1:20:size(Xres, 2)
    plot(Xres(j),Yres(j),'ro','markersize',4,'markerfacecolor','r');

```

```
    pause(0.05);  
end hold off;
```

Anexo B – Listagem da rotina de geração de movimentos

A rotina de geração de movimentos foi desenvolvida em linguagem *script* do *Embedded MatLab® Editor* para *Simulink®*. Abaixo segue o programa fonte na sua totalidade.

```
function [Vd, Ve, i2, Wref, TetaRef, Raio_c, jrb, tet]=
    V_rodas(Vlin, L, X, Y, traj1, Teta_antRef, i1, ja)
Delta_t2 = .01;
%A matriz trajetória (traj1) é composta de:
%na coluna 1 dos pontos inicial e final das retas
    e arcos ref. ao deslocamento X do robô
%na coluna 2 dos pontos inicial e final das retas
    e arcos ref. ao deslocamento Y do robô
%na coluna 3 dos raios de curvaturas nos segmentos
    de reta e nos arcos
%na coluna 4 das inclinações angulares (teta)
    ref. aos segmentos de reta
Raio_c = traj1(i1,3); tet=traj1(i1,4); Xrs =
    ceil(traj1(i1,1)*6)/6; Xri = floor(traj1(i1,1)*6)/6;
Yrs = ceil(traj1(i1,2)*6)/6; Yri = floor(traj1(i1,2)*6)/6;

if ((X < Xrs) && (X > Xri) && (Y < Yrs) && (Y > Yri))
    i2 = i1 + 1;
else
    i2 = i1;
end
```

```

%Velocidade linear da roda direita e esquerda
alfa = L / (2 * Raio_c); Vd = Vlin * (1 + alfa); Ve = Vlin * (1 - alfa);

%Wref = velocidade angular de referência do robô em rad/s
Wref = (Vd - Ve) / L;
%Teta = deslocamento angular do robo no tempo em rad
Delta_TetaRef = Wref * Delta_t2; TetaRef = Teta_antRef + Delta_TetaRef;

if ((abs(Wref) < 0.01) && (abs(tet-TetaRef) > 0.01))
    jb = ja + 1;
    Raio_c = (traj1(i1-1,3));
    alfa = L / (2 * Raio_c);
    Vd = Vlin * (1 + alfa);
    Ve = Vlin * (1 - alfa);
    Wref = (Vd - Ve) / L;
    Delta_TetaRef = Wref * Delta_t2;
    TetaRef = Teta_antRef + Delta_TetaRef;
else
    jb=ja;
end

```

Anexo C – Listagem da rotina do modelo cinemático do robô

A rotina de geração do modelo cinemático do robô móvel foi desenvolvida em linguagem *script* do MatLab® e tem como base matemática as equações desenvolvidas no no capítulo 2, nas seções 2.3.1 e 2.3.2. Abaixo segue o programa fonte na sua totalidade.

```
function [X,Y,V_rob0,R_c,Teta,W] = ...
    trajetoria(V_d,V_e,L,t,Teta_ant,X_ant,Y_ant)

% Monta os vetores X, Y e o Teta da trajetória dinâmica do robo móvel

Delta_t = .01;

if V_d == V_e      %trajetória reta
    V_rob0 = V_d;
    Teta = Teta_ant;
    %X = posição do robo no eixo X
    X = X_ant + V_rob0 * cos(Teta) * Delta_t;
    %Y = posição do robo no eixo Y
    Y = Y_ant + V_rob0 * sin(Teta) * Delta_t;
    R_c = inf;
    W=0;
else              %trajetória circular
    %W = velocidade angular do robô em rad/s
    W = (V_d - V_e) / L;
    %R_c = raio de curvatura da trajetória do robô
    R_c = (V_d + V_e) / (2 * W);
```

```

%Teta = deslocamento angular do robo no tempo em rad
Delta_Teta = W * Delta_t;
Teta = Teta_ant + Delta_Teta;
%Centro de curvatura instantânea
CCIx = X_ant - R_c * sin(Teta);
CCIy = Y_ant + R_c * cos(Teta);
%Velociade linear do robô
V_robo = ((V_d - V_e) * R_c) / L;
%Posição X e Y do robô
X = (X_ant - CCIx) * cos(Delta_Teta) - (Y_ant - CCIy)...
    * sin(Delta_Teta) + CCIx;
Y = (X_ant - CCIx) * sin(Delta_Teta) + (Y_ant - CCIy)...
    * cos(Delta_Teta) + CCIy;
end

```

Anexo D – Listagem da rotina de inicialização dos atuadores do robô

A rotina de inicialização dos atuadores, com todos os parâmetros e características de cada atuador utilizado no robô móvel, foi implementada em linguagem *script* do MatLab®. Abaixo segue o programa fonte na sua totalidade.

```
%Inicializa os parâmetros dos motore DC e do controlador PID

%Motor 1 --> Roda Direita
%Motor 2 --> Roda Esquerda

disp('Inicialização');

clc;
%close all;
%clear all;

%Raio das rodas de tração do robô
Rd = 0.04;    %m

%Comprimento do eixo das rodas de tração do robô
L = 0.387;    %m

%Posicionamento inicial do robô
%X_inic = 2;    %m
%Y_inic = 3;    %m
%Teta_inic = pi/2;    %rad
```

```

%Parametros dos motores
Ra = [2 2];           % Resistência da bobina em Ohm
La = [0.18 0.18];    % Indutância da bobina em Henry

%Constante de torque do motor
Kt = [0.00001 0.00001]; % N.m/A

%Constante da FCEM
Kb = [56.5 56.5];    % V. (rad/s)

%Momento de inercia do motor
Jm = [0.0001 0.0001]; % (Kgm^2)

%Amortecimento(atrito) viscoso do motor
Bm = [0.00005 0.00005]; % N.m/(rad/s)

%Relacao de transmissao (reductor)
N1 = [1 1]; %motor
N2 = [10 10]; %carga
N = N1 ./ N2;

%Velocidade angular máxima do motor (rotação)
Wmx = [300 300]; % rad/s

%Carga (momentos de inercia de massa)
Jl = [0.01 0.01]; % Kg.m^2

% Coeficientes de Atrito Viscoso (devido a carga)
Bl = [0.0115 0.0115]; % N.m/(rad/s)

%momento de inercia equivalente
Jeq = [Jm(1)+Jl(1)*N(1)^2 Jm(2)+Jl(2)*N(2)^2]; % Kg.m^2

%atrito viscoso equivalente

```

```

Beq = [Bm(1)+Bl(1)*N(1)^2 Bm(2)+Bl(2)*N(2)^2];      % N.m/(rad/s)

%Função de transf. do motor DC
disp('Função de transferência do motor DC') num_dc =...
    [0 0 0 Kt(1)];
den_dc = [La(1)*Jm(1) (Ra(1)*Jm(1))+(La(1)*Bm(1))...
    (Ra(1)*Bm(1))+(Kb(1)*Kt(1)) 0];

planta_motor = tf(num_dc,den_dc)

%Função de transf. do motor DC + redutor
disp('Função de transferência do sistema: Motor + redutor')...
    num_sis = num_dc;

den_sis = [La(1)*Jeq(1) (Ra(1)*Jeq(1))+(La(1)*Beq(1))...
    (Ra(1)*Beq(1))+(Kb(1)*Kt(1)) 0];

planta_sis = tf(num_sis,den_sis)

```

Anexo E – Publicações

MELO, L. F.; ROSÁRIO, J. M.; “A Hybrid Opened Architecture with Hardware Reconfigurable Control Applied in Nonholonomic Mobile Robots”, 19th International Congress of Mechanical Engineering-COBEM 2007. November 5-9, 2007, Brasília, DF.

MELO, L. F.; ROSÁRIO, J. M.; “A Proposal for a hybrid opened architecture with hardware reconfigurable control applied in mobile robots”, 12/2006, IEEE International Conference on Robotic and Bionemetics - ROBIO 2006, Vol. 1, pp. 1101-1106, Kunming, China, 2006.

MELO, L. F.; ROSARIO, J. M. “Hardware Reconfigurable Control With Opened Architecture for Mobile Robots”. 18th International Congress of Mechanical Engineering-COBEM 2005. Ouro Preto, MG. Proceedings of COBEM 2005. vol. 1, p. 1-8, novembro de 2005.

ROSARIO, J. M.; MELO, L. F.; LIMA, C. R. E.; SARAMAGO, M. A. P. “A Reconfigurable Control Architecture for Mobile Robots”. MuSMe 2005-2nd. Internacional Symposium on Mutibody and Mechatronics, Uberlândia, MG. Proceedings of MUSME 2005. NY : IEEE, 2005. v. 1. p. 1-8, março de 2005.

MELO, L. F.; LIMA, C. R. E.; ROSARIO, J. M. “Robô Móvel com Sistema de Arquitetura Aberta Utilizado na Realização de Tarefas de Inspeção das Paredes Internas e Externas de Dutos”. Congresso Internacional de Automação-ISA Show 2004, São Paulo, Revista InTech, ISSN 1518-6024, nr. 67, vol. 1, p. 77, novembro de 2004.

MELO, L. F.; LIMA, C. R. E.; ROSARIO, J. M. “Uma Aplicação da Robótica Móvel Utilizando Sistema de Arquitetura Aberta”. 3o. Congresso Temático de Dinâmica e Controle da SBMAC (Sociedade Brasileira de Matemática Aplicada e Computacional)-DINCON'2004. UNESP, Ilha Souteira, SP. vol. 1, p. 1369-1373, junho de 2004.

MELO, L. F.; LIMA, C. R. E.; ROSARIO, J. M. “Uma Utilização de Sistemas de Arquitetura Aberta Aplicada em Sistemas Robóticos”. Congresso Internacional de Automação-ISA Show 2003, São Paulo, Revista InTech, ISSN 1518-6024, nr. 66, vol. 1, p. 42-48, novembro de 2003