

Armando Jorge Miranda de Sousa

**Arquitecturas de Sistemas
Robóticos e Localização em Tempo
Real Através de Visão**

**Aplicações no Domínio do
Futebol Robótico**

Dissertação submetida para a obtenção do grau de
Doutor em Engenharia Electrotécnica e de Computadores
pela Faculdade de Engenharia da Universidade do Porto

Trabalho realizado sob a orientação do Professor Doutor

Adriano da Silva Carvalho

e do Professor Doutor

Paulo José Cerqueira Gomes da Costa

Universidade do Porto
Faculdade de Engenharia
Departamento de Engenharia Electrotécnica e de Computadores
2003

Apoios:

O autor foi,
durante a realização desta dissertação,
bolseiro do PRODEP III, N° 5/5.3./PRODEP 2000

Trabalho também parcialmente suportado pelo projecto
PORTUS
Uma Estrutura Genérica para Cooperação em Robótica Móvel,
financiado pela FCT com ref. N° POSI/SRI/41315/2001

Gostaria de agradecer:

Aos meus pais e à minha família

À minha querida Daniela

*Aos meus orientadores:
Paulo Costa e Adriano Carvalho*

*Aos meus colegas do Futebol Robótico:
António Paulo Moreira, Luís Paulo Reis,
Paulo Marques, Pedro Costa e Susana Gaio*

*Aos meus colegas da FEUP:
Aníbal Matos, Carlos João Ramos,
Inês Carvalho e Paulo Portugal*

*Aos meus amigos que ajudaram neste trabalho
Luís Ferreira
Filipe Pacheco*

A todos os outros que procuraram ajudar

*Muito, muito obrigado,
sem o vosso apoio tudo
seria muito muito mais complicado !*

*“Genius is one percent inspiration
and ninety-nine percent perspiration.”*

*“Gênio é um por cento de inspiração
e noventa e nove por cento transpiração”*

Thomas A. Edison, 1932

Resumo

A necessidade de autonomia dos sistemas robóticos incrementa a sua complexidade e o seu projecto cuidadoso torna-se assim ainda mais importante. Este trabalho apresenta o projecto de sistemas robóticos de forma abrangente, tirando partido da experiência de projecto de diversas plataformas para competições de futebol robótico da federação RoboCup.

A autonomia em robótica, associada à mobilidade, leva à necessidade de localização em Tempo Real. O conceito do codificador de janela deslizante é apresentado como sendo a base de um inovador sistema de localização, adequado a ambiente estruturado. Tal sistema é baseado em visão e permite a localização completa do robot em x, y, θ , conseguida em Tempo Real.

Os conceitos relativos à localização dinâmica de sistemas em ambiente estruturado são validados experimentalmente. Para tal, é utilizado um conjunto que inclui um sistema complementar de localização externa para confronto com o sistema integrado de auto-localização. É utilizado um Filtro de Kalman Extendido para a fusão de informação de auto-localização, que inclui medidas da hodiometria, do reconhecimento de linhas e postes e ainda do sistema baseado no codificador de janela deslizante. É feita a caracterização de cada medida e demonstrada a importância da fusão de informação para reunir toda a informação válida. Após a análise dos resultados obtidos, discute-se a aplicabilidade das metodologias propostas em aplicações reais.

Résumé

L'autonomie dans les systèmes robotiques augmente sa complexité et rend la conception soignée très importante. Ce travail présente une méthodologie de conception globale basée sur l'expérience acquise de la conception de plusieurs robots footballeurs pour des participations aux jeux de la Fédération RoboCup.

L'autonomie robotique, ainsi que la mobilité, a le besoin de localisation en Temps Réel. L'encodeur coulissant de fenêtre est la base d'une stratégie innovatrice de localisation, utile pour la localisation dans un environnement structurée. Ce système est basé uniquement dans la vision et permet la localisation complète du robot (x, y, θ) sous des contraintes de Temps Réel.

Les méthodologies présentées pour la localisation dynamique sont examinées dans des expériences réalistes. Les expérimentations utilise un système externe de localisation qui est employé pour établir la qualité des méthodologies intégrées de localisation dans le robot. Un *Extended Kalman Filter* est utilisé pour la fusion de données. Les données disponibles sont l'odométrie, les lignes du champ, les poteaux et le nouveau système qui est basé sur l'encodeur coulissant de fenêtre. Chaque mesure est caractérisée et l'importance de fusion de données est démontrée. L'applicabilité des idées présentées pour d'autres vraies applications est également discutée.

Abstract

The need for autonomy in robotic systems augments its complexity and makes careful design very important. This work presents an overall design methodology based on the experience gained from the design of several Robotic Soccer platforms for participation in the RoboCup Federation games.

Robotic autonomy, together with mobility, has the need for Real Time (RT) Localization. The “sliding window encoder” is the basis of an innovative localization strategy, useful for structured environment localization. This system is based uniquely in vision and allows for complete (x, y, θ) localization under RT constraints.

The presented methodologies for dynamic localization are tested in a realistic environment. The test suit involves an external localization system that is used to ascertain the quality of the integrated self localization methodologies. An Extended Kalman Filter (EKF) is used for data fusion. The available data is odometry, field line recognition, pole recognition and the new system that is based on the sliding window encoder. Each measurement is characterized and data fusion importance is demonstrated.

The applicability of the ideas presented for other real applications is also discussed.

Índice

Resumo.....	ix
Résumé.....	x
Abstract.....	xi
Índice.....	xii
Índice de Figuras.....	xviii
Índice de Tabelas.....	xxiv
Capítulo 1	
Introdução.....	1
1.1 Motivação.....	1
1.2 Contexto.....	2
1.3 Contribuições.....	3
1.4 Estrutura.....	3

Capítulo 2

Projecto de Sistemas Robóticos.....	5
2.1 Introdução.....	5
2.2 Requisitos.....	6
2.3 Áreas de Projecto.....	8
2.4 Projecto Global e Mecatrónica.....	9
2.5 Tipo de Robot e sua Locomoção.....	11
2.6 Projecto Mecânico.....	20
2.7 Sistema de Alimentação.....	24
2.8 Sistema Embebido.....	29
2.8.1 Sistemas Tempo Real.....	29
2.8.2 Classificação de Sensores.....	29
2.8.3 Sensores Comuns em Robótica.....	30
2.8.4 Actuadores.....	31
2.8.5 Implementação de Subsistemas.....	32
2.8.6 Comunicações Internas.....	33
2.9 Navegação.....	34
2.10 Decisão, Autonomia e Cooperação.....	35
2.10.1 Cooperação entre Sistemas Robóticos.....	38
2.10.2 Tentativa de Indiferença ao Hardware.....	39
2.10.3 Fusão de Informação.....	39
2.11 Interface e Comunicações com o Exterior.....	41
2.12 Ferramentas de Ajuda ao Desenvolvimento.....	42
2.13 Conclusões.....	43
Capítulo 3	
Sistemas de Localização.....	44
3.1 Enquadramento e Definições.....	44
3.2 Abordagens à Localização.....	45
3.2.1 Medidas Relativas.....	45
3.2.2 Medidas Face ao Ambiente.....	46
3.2.3 Metodologias Frequentemente Utilizadas.....	47
3.3 Soluções de Localização	49
3.3.1 Soluções Envolvendo LASER.....	49
3.3.2 Soluções Envolvendo RF.....	49
3.3.3 Sistemas Globais de Navegação por Satélite.....	51
3.3.4 Soluções Baseados na Rede de Telemóveis.....	57
3.3.5 Soluções Envolvendo Ondas Acústicas.....	58
3.3.6 Soluções Envolvendo Visão.....	59

3.4	Localização e Mapeamento Simultâneo.....	64
3.5	Conclusões.....	66
Capítulo 4		
	Sistemas de Visão.....	67
4.1	Cor e Imagem.....	67
4.2	Modelo para a Câmara.....	70
4.2.1	Parâmetros intrínsecos.....	71
4.2.2	Parâmetros Extrínsecos.....	74
4.2.3	Relação imagem-mundo (UVZ2XY).....	75
4.2.4	Calibração de Parâmetros.....	76
4.3	Conclusões.....	77
Capítulo 5		
	Processos Estocásticos e Filtros de Kalman.....	78
5.1	Introdução.....	78
5.2	Variáveis aleatórias.....	79
5.2.1	Fundamentos acerca de Probabilidades.....	79
5.2.2	Atributos de Variáveis Aleatórias.....	80
5.2.3	Propriedades dos Atributos.....	81
5.2.4	Expressões Aproximadas para Operações entre Atributos.....	81
5.2.5	Distribuições Gaussianas.....	82
5.3	Método dos Mínimos Quadrados.....	83
5.4	Filtro de Kalman.....	84
5.4.1	Filtro de Kalman Discreto.....	85
5.4.2	Filtro Kalman Extendido.....	87
5.4.3	Teste Estatístico.....	91
5.5	Conclusões.....	92
Capítulo 6		
	Codificador de Janela Deslizante.....	93
6.1	Introdução aos Códigos de Barras.....	93
6.2	Introdução aos Codificadores.....	94
6.3	Definições.....	95
6.4	Sequências Unívocas.....	96
6.5	Exemplo de Implementação.....	97
6.6	Implementações multidimensionais.....	98
6.7	Implementações com restrições adicionais.....	98

6.8 Geração de Sequências Unívocas.....	100
6.9 Conclusões.....	101
 Capítulo 7	
Implementações Relevantes para o Ambiente RoboCup.....	102
7.1 Introdução.....	102
7.2 Federação RoboCup.....	103
7.3 Robots futebolistas F180 (ano 2003).....	105
7.3.1 Requisitos.....	105
7.3.2 Descrição Geral.....	106
7.3.3 Sistema de Alimentação.....	106
7.3.4 Localização.....	107
7.3.5 Locomoção.....	107
7.3.6 Actuadores.....	107
7.3.7 Visão.....	108
7.3.8 Arquitectura global.....	108
7.3.9 Melhoramentos.....	109
7.4 Robots futebolistas F2000 (ano 2003) – plataforma experimental.....	109
7.4.1 Requisitos.....	109
7.4.2 Descrição geral.....	110
7.4.3 Sistema de alimentação.....	111
7.4.4 Locomoção.....	111
7.4.5 Actuadores.....	111
7.4.6 Comunicações com o exterior.....	112
7.4.7 Arquitectura.....	112
7.4.8 Visão.....	112
7.4.9 Decisão.....	114
7.4.10 Controlo.....	114
7.4.11 Detalhes de implementação.....	115
7.4.12 Melhoramentos.....	116
7.5 Conclusões.....	116
 Capítulo 8	
Validação Experimental.....	117
8.1 Procedimento experimental.....	117
8.2 Câmara Externa (VHC).....	119
8.2.1 Introdução.....	119
8.2.2 Determinação dos parâmetros.....	120

8.2.3	Localização Externa do Robot.....	122
8.2.4	Caracterização da medida.....	124
8.2.4.1	Estudo Analítico do Erro em Função da Distância à Câmara (VHC).....	126
8.2.4.2	Medição Experimental do Erro de Posicionamento.....	128
8.3	Calibração da Câmara do Robot.....	129
8.4	Auto-Localização por Código de Barras Unívocos.....	133
8.4.1	Modelo do erro na medida.....	135
8.4.2	Processamento de Imagem.....	135
8.4.3	Análise da medida.....	144
8.5	Auto-Localização por Linhas.....	147
8.5.1	Modelo do erro na medida.....	148
8.5.2	Implementação.....	148
8.5.3	Análise Resultados.....	151
8.6	Localização por Postes.....	155
8.6.1	Modelo do erro na medida.....	156
8.6.2	Filtro de Kalman Extendido.....	157
8.6.3	Implementação.....	158
8.6.4	Análise de Resultados.....	161
8.7	Hodometria.....	164
8.8	Ciclo de Fusão de Informação.....	164
8.9	Simulação da Auto-Localização.....	165
8.10	Testes Dinâmicos.....	171
8.10.1	Localização através da medida de Linhas.....	171
8.10.2	Localização através da medida de Postes.....	173
8.10.3	Localização completa.....	174
8.11	Conclusões.....	176

Capítulo 9

Conclusões.....	178
9.1 Trabalho futuro.....	180

Capítulo 10

Referências Bibliográficas.....	182
---------------------------------	-----

Anexo I

Acrónimos e Siglas.....	195
-------------------------	-----

Anexo 2	
Marcos Históricos da Robótica.....	197
Anexo 3	
Tabela de Algumas Sequências Unívocas.....	199
Anexo 4	
Ficha Técnica.....	202

Índice de Figuras

Figura 2.1 Ilustração dos constituintes do sistema robótico.....	6
Figura 2.2 Representação simplificada das áreas científico-tecnológicas envolvidas em projecto robótico.....	9
Figura 2.3 Conceitos de base na Mecatrónica, adaptado de [2].....	10
Figura 2.4 Dinâmica de projecto de acordo com os princípios da mecatrónica (adaptado de [2]).	11
Figura 2.5 Locomoção diferencial.....	12
Figura 2.6 Locomoção tipo triciclo.....	12
Figura 2.7 Locomoção tipo Ackerman.....	13
Figura 2.8 Locomoção do tipo “Synchronous Drive”.....	13
Figura 2.9 “nBot” Robot que se equilibra em duas rodas [7].....	14
Figura 2.10 Robot com múltiplos graus de liberdade [6].....	14
Figura 2.11 Robot “BloodHound” para busca e salvamento [8].....	14
Figura 2.12 “Soryu”: à esquerda fotografia do protótipo e à direita o protótipo “rebola” por umas escadas [9].....	15
Figura 2.13 Locomoção OmniDireccional.....	15
Figura 2.14 À esquerda está um robot “protótipo rápido” com rodas omni-direccionais e à direita uma roda omni-direccional ampliada [13].....	16
Figura 2.15 Micro-robot voador (uma “mosca”) [22].....	16
Figura 2.16 “Minhoca totalmente aderente” [18].....	16
Figura 2.17 Robot esférico [19].....	16
Figura 2.18 Robot “Spiral Track” [21].....	16
Figura 2.19 Robots “Alice”: à esquerda foto do robot (com escala); à direita estes robôs a jogarem futebol 3 contra 3 [24].....	17
Figura 2.20 Honda P3 e Asimo [27].....	17
Figura 2.21 Humanóide Sony SDR4x [28].....	17
Figura 2.22 Robot tubarão [43].....	18
Figura 2.23 Robot Lagosta [44].....	18

Figura 2.24 Movimento do robot do tipo serpentina (adaptado de [47]).....	19
Figura 2.25 Cão robótico “Aibo” modelo ERS220, da Sony [36].....	19
Figura 2.26 Gato robótico Necoro da Omron [37].....	19
Figura 2.27 Robot Isurus [52].....	20
Figura 2.28 Render do robot submarino Infante.....	20
Figura 2.29 Hélice dentro do propulsor do Infante.....	20
Figura 2.30 Propulsor do Infante.....	20
Figura 2.31 Benefícios provenientes da utilização de sistemas CAD.....	21
Figura 2.32 Modelo de CAD do conjunto do robot 5dpo, F180, ano 2003 (comparar com fig. 7.4).....	23
Figura 2.33 Distribuição de energia dentro do robot.....	25
Figura 2.34 Comparação da densidade energética de diversas soluções para armazenamento e produção de energia eléctrica – as linhas diagonais são para descarga em tempo igual para determinada carga – adaptado de [63]	28
Figura 2.35 Arquitectura típica de um sistema de navegação.....	35
Figura 2.36 Blocos de entradas, saídas e decisão dentro do robot e respectivo relacionamento com o ambiente exterior.....	36
Figura 2.37 Robots cooperando (Collective Robot [77]).....	38
Figura 3.1 Dois ângulos internos ao robot não localizam um robot (mostram-se 4 localizações possíveis de um número infinito de soluções).....	47
Figura 3.2 Não sabendo que medida diz respeito a que marcador, um robot com 2 medidas de ângulos e respectivas distâncias a marcadores fica com duas hipóteses de localização....	48
Figura 3.3 Intercepção de 3 calotas esféricas resulta nos 2 pontos assinalados com os marcadores de forma “bicuda”; à direita temos a versão translúcida da figura da esquerda onde as esferas pequenas representam os marcadores de localização (centros das outras esferas).....	48
Figura 3.4 GPS com e sem Selected Availability - adaptado de [114].....	53
Figura 3.5 Ilustração do funcionamento do GPS Diferencial.....	54
Figura 3.6 Diagrama de blocos do Galileo – adaptado de [121]	56
Figura 3.7 Princípio de Funcionamento de um PSD – adaptado de [143].....	60
Figura 3.8 Iluminação estruturada (adaptado de [145]).....	61
Figura 3.9 Esquema do espelho isométrico - de [149] -	61
Figura 3.10 Utilização de espelhos para obter duas imagens – adaptado de [151].....	62
Figura 3.11 Robot tentando localização absoluta utilizando padrão ternário [160].....	63
Figura 3.12 Método SLAM utilizando dados directos dos sensores.....	65
Figura 4.1 Sensores RGB nos Humanos (adaptado de [170]).....	67
Figura 4.2 Espectro de cor (adaptado de [173]).....	68
Figura 4.3 Sensibilidade a cada cor (adaptado de [173]).....	68
Figura 4.4 À esquerda, cores primárias e secundárias; à direita pode-se ver o cubo RGB.....	68
Figura 4.5 Ilusão óptica num tabuleiro em xadrez.....	69
Figura 4.6 Sensibilidade global da visão humana (adaptado de [173]).....	70
Figura 4.7 Casos frequentes em visão para robótica.....	71
Figura 4.8 Efeitos de uma projecção	71
Figura 4.9 Esquema visual de uma câmara mostrando o plano virtual da imagem.....	71
Figura 4.10 Sistema de eixos no modelo da câmara.....	72
Figura 4.11 Efeito de uma lente.....	72

Figura 4.12 Coordenadas da imagem.....	72
Figura 4.13 Ilustração da imagem com distorções do tipo barril e “pin-cushion” num exemplo apresentando a resolução de 384 por 288 pixels.....	73
Figura 4.14 Sistemas de eixos utilizados, representação em perspectiva isométrica.....	74
Figura 4.15 Sistemas de eixos em detalhe.....	75
Figura 6.1a) Diversos exemplos de códigos de barras; b) os dois códigos assinalados contêm a mesma informação.....	94
a) b) Figura 6.2 Discos de codificadores para medida angular: a) versão incremental; b) versão absoluta.....	95
Figura 6.3 Codificador de Janela utilizado pelo pescoço dos Robots da Equipa 5dpo-2000.....	98
Figura 6.4 Robot com codificador de janela deslizante.....	98
Figura 6.5 Sequência unívoca largura $j=2$ implementada com 2 dígitos na vertical, cada um deles com 4 níveis de sinal, $b=8$, dígito menos significativo em baixo. As cores seguem o código das resistências, (0=preto, 1= castanho, 2=vermelho, 3=laranja, etc.).....	98
Figura 6.6 Sequência unívoca Binária, Janela=4, sem restrições.....	99
Figura 6.7 Sequência unívoca Binária, Janela=4, sem sequências de 4 dígitos iguais consecutivos.....	99
Figura 6.8 Sequência unívoca Binária, Janela=4, sem sequências de 3 nem de 4 dígitos iguais consecutivos.....	99
Figura 7.1 Descrição geral do robot desenvolvido para liga F180.....	106
Figura 7.2 Imagem do CAD para projecto do sistema de chuto.....	107
Figura 7.3 Diagrama do sistema 5dpo para a liga F180.....	108
Figura 7.4 Robots da equipa 5dpo, liga F180 (Foto tirada no RoboCup 2003, Pádua, Itália). ..	109
Figura 7.5 Descrição geral do robot desenvolvido para liga F2000.....	111
Figura 7.6 Arquitectura de Software para a equipa 5dpo-2000.....	112
Figura 7.7 Robot 5dpo-2000 versão modificada para movimento omni-direccional a ser configurado (foto no RoboCup 2003, Pádua, Itália).....	113
Figura 7.8 Ilustração do funcionamento da equipa 5dpo-2000 para o RoboCup 2003, Pádua, Itália.....	114
Figura 8.1 Manchas RGB para 3 conjuntos de cores dentro do cubo RGB (cores “laranja”, “azul” e “amarelo”).....	118
Figura 8.2 Ampliação da imagem obtida com uma vulgar CamCorder tal como a utilizada para a localização externa.....	118
Figura 8.3 Robot sendo localizado exteriormente.....	120
Figura 8.4 Pontos da imagem (U,V) para VHC.....	120
Figura 8.5 Pontos da imagem (Xw,Yw) para VHC.....	121
Figura 8.6 Mapeamento VHC.....	121
Figura 8.7 Imagens de todas as aplicações VHC – em cima imagem e classificação de cor; em baixo manchas de cor de facto transmitidas por rede e resultado visual do robot no campo; fundo da aplicação e cor não classificada representadas a branco.....	122
Figura 8.8 Classificação de cor em zonas estreitas resulta em poucos pixels correctamente classificados; cor não classificada representada a branco.....	123
Figura 8.9 Tipos de “chapéu” para localização externa do robot.....	123
Figura 8.10.....	124
Figura 8.11 Histogramas bidimensionais para as manchas de cor referidas nas figura 8.10.....	124

Figura 8.12 Histogramas dos dados enviados após localização por VHC.....	126
Figura 8.13 Desvio padrão da localização VHC (todas as grandezas em metros).....	127
Figura 8.14 Desvio padrão para outros valores de localização VHC (todas as grandezas em metros).....	127
Figura 8.15 Gráfico do desvio padrão dos resultados experimentais a diferentes distâncias à câmara.....	128
Figura 8.16 Imagem da câmara dentro do robot durante a sua calibração (capturada através de um utilitário de desktop remoto).....	129
Figura 8.17 Posições na imagem (u,v) dos pontos de calibração	130
Figura 8.18 Mapeamento imagem mundo dentro do robot com os parâmetros mostrados em (8.9).....	130
Figura 8.19 Mapeamento imagem mundo dentro do robot com os parâmetros mostrados em (8.10).....	131
Figura 8.20 Correção do efeito de barril	132
Figura 8.21 Pontos da imagem transformados de acordo com a correção do efeito de barril remapeados nos seus lugares iniciais através do parâmetro kc.....	132
Figura 8.22 Esquema para localização por BC.....	133
Figura 8.23 Detalhe dos ângulos envolvidos na localização por BC.....	133
Figura 8.24 Configurações do Processamento de Imagem para localização BC.....	136
Figura 8.25 Imagem no robot à esquerda e derivada na luminância à direita (imagem capturada através de desktop remoto).....	137
Figura 8.26 Figura 8.25 tratada, porção invertida e ampliada para melhor legibilidade (por imagem invertida entenda-se transformar cores escuras em claras e vice versa).....	137
Figura 8.27 Comparação dos 4 modos de cálculo de derivadas: Modo 1 – cima esquerda ; Modo 2 – cima direita; Modo 3 – baixo esquerda; Modo 4 – baixo direita; os números dentro das caixas representam, da esquerda para a direita: número de pontos seleccionados, número de segmentos horizontais, número de regiões rectangulares e número final de segmentos.....	138
Figura 8.28 Evolução do processamento de imagem da situação da figura 8.25, pontos seleccionados por a derivada superar dado limiar.....	138
Figura 8.29 Evolução do processamento de imagem da situação da figura 8.25, mostrando segmentos e agrupamentos de pontos em zonas rectangulares.....	139
Figura 8.30 Evolução do processamento de imagem da situação da figura 8.25, ajuste de segmentos.....	139
Figura 8.31 Segmentos recebidos pela aplicação DEC.....	140
Figura 8.32 Determinação das 2 “horizontais” do código de barras.....	140
Figura 8.33 Classificação de segmentos válidos: Teste 3 – ajuste sobre uma linha de ângulos e diferença de ângulos.....	141
Figura 8.34 Resultados do teste do ajuste cego sobre a parede.....	142
Figura 8.35 A localização BC utiliza os pontos nos centros dos segmentos relativos ao padrão regular (por baixo) e ao código unívoco (por cima).....	143
Figura 8.36 A localização BC também pode utilizar as extremidades dos segmentos ajustados para a localização	143
Figura 8.37 À esquerda alguns dados da aplicação de processamento de imagem, mostrando o ajuste das rectas conseguido e à direita a localização BC.....	144
Figura 8.38 Histogramas para localização por BC (todas as distâncias em metros).....	145

Figura 8.39 Histograma 2D da localização BC (todas as distâncias em metros).....	145
Figura 8.40 Histogramas para localização por BC na situação bi-modal (todas as distâncias em metros).....	145
Figura 8.41 Histogramas para localização por BC na situação bi-modal (todas as distâncias em metros).....	146
Figura 8.42 Disposição das medidas BC no campo experimental.....	146
Figura 8.43 Esquema da localização por linhas.....	147
Figura 8.44.....	149
Figura 8.45 Transições entre cores detectadas ao longo das rectas e linhas fronteira detectadas (uma é incorrectamente assinalada).....	149
Figura 8.46 Uma linha que não consegue ser detectada.....	150
Figura 8.47 Histogramas na medida de linhas (todas as distâncias em metros).....	151
Figura 8.48 Histograma bidimensional da dispersão na medida do ângulo à linha.....	152
Figura 8.49 Histograma bidimensional da dispersão na medida da distância à linha.....	152
Figura 8.50 Dispersão na medida de linhas.....	153
Figura 8.51 Disposição no campo das medidas de linhas.....	153
Figura 8.52 Esquema da localização por marcadores redondos/postes.....	155
Figura 8.53 Ilustração do robot em movimento.....	157
Figura 8.54 Encenação para classificação de cor relativamente a 4 postes em frente da baliza amarela (imagem rodada daquela que o robot realmente visualiza).....	158
Figura 8.55 Imagem de situação encenada em que são visíveis: à esquerda a imagem com quatro postes; ao centro a classificação de cor conseguida (o verde do campo não está calibrado); à direita, os dados recebidos pela decisão.....	159
Figura 8.56 Resumo das dimensões associadas à medida de postes.....	159
Figura 8.57 Distância mínima para um poste ser visível.....	161
Figura 8.58 Dispersão na medida de Postes em forma de gráfico.....	162
Figura 8.59 Diagrama da dispersão do ângulo na medida de postes.....	162
Figura 8.60 Diagrama da dispersão da distância na medida de postes.....	162
Figura 8.61 Medidas de postes representadas no campo.....	163
Figura 8.62 Detalhe dos robots real e estimado, representados em $x=0, y=0$,	165
Figura 8.63 Traçados de vr e	166
Figura 8.64 EKF sem nenhuma informação de localização.....	167
Figura 8.65 Simulação do robot localizando-se através do poste 1 (assinalado por uma estrela de 5 pontas).....	167
Figura 8.66 Simulação do robot localizando-se através do poste 2 (assinalado por uma estrela de 6 pontas).....	168
Figura 8.67 Simulação do robot localizando-se através dos postes 1 e 2.....	168
Figura 8.68 Simulação do robot localizando-se através da linha horizontal.....	169
Figura 8.69 Simulação do robot localizando-se através da linha vertical.....	169
Figura 8.70 Simulação do robot localizando-se através de um CBU.....	170
Figura 8.71 Simulação do robot localizando-se através da combinação de: 2 postes, 2 linhas e CBU.....	170
Figura 8.72 Teste dinâmico na localização por Linhas (tempos em segundos).....	171
Figura 8.73 Teste dinâmico na localização por Postes (tempos em segundos).....	173
Figura 8.74 Localização recorrendo a BC, Postes e Linhas (tempos em segundos).....	175

Índice de Tabelas

Tabela 2.1 Comparação de Baterias, adaptado a partir de [59], [60] e [61].....	27
Tabela 3.1 Sumário das várias fontes de erro presente no GPS, valor típico do erro em metros, por satélite (tabela compilada a partir de [86], [116] e [115]).....	55
Tabela 3.2 Precisão de sistemas de localização com base na rede celular de telemóveis.....	58
Tabela 3.3 Resumo comparativo e pontos fracos da utilização de diversos elementos a seguir [161].....	64
Tabela 5.1 Resumo da operação do Filtro de Kalman.....	87
Tabela 5.2 Resumo da operação do EKF.....	91
Tabela 8.1 Tabela do erro médio na medida VHC face à pose real (cada experiência com 256 medidas).....	128
Tabela 8.2 Resumo das experiências relativas à localização por BC.....	146
Tabela 8.3 Resumo das experiências relativas à medição de Linhas.....	154
Tabela 8.4 Resumo das experiências relativas à medição de Postes.....	163
Tabela 8.5 Valores finais da sequência da figura 8.74.....	176

Capítulo 1

Introdução

1.1 Motivação

A humanidade há muito que sonha com os benefícios dos sistemas automáticos. O filósofo grego Aristóteles cerca do ano 332 AC refere «se uma ferramenta quando ordenada ou mesmo por sua própria vontade pudesse fazer o seu trabalho... não seriam necessários aprendizes para os mestres nem escravos para os seus senhores...». Este filósofo previu a importância que a automatização viria a ter ao antever ferramentas que ajudariam o mestre. Tais equipamentos teriam até «vontade própria», o que conduz à noção de autonomia, isto é, a «ferramenta» que não depende permanentemente do ser humano.

Em 1921 Karel Čapek introduziu a palavra “Robot” na novela “*R.U.R. - Rossum's Universal Robots*”. A palavra vem de “*robota*”, que significa trabalho repetitivo/entediante na língua checa e assim os robots executariam tarefas repetitivas e/ou entediantes para o ser humano (consultar o Anexo 2 Marcos Históricos da Robótica, pág 198).

As ferramentas automáticas com autonomia culminam nos dias actuais nos **sistemas robóticos**, isto é, sistemas dotados de liberdade de movimentos para cumprirem tarefas com um certo nível de autonomia de decisão. O sistema robótico inclui o robot e os seus sistemas periféricos. O robot pode ser

considerado como a fachada visível de um sistema complexo, que pode ser, em parte, dependente de alguns sistemas exteriores. A autonomia do sistema robótico significa que ele deve ser capaz de tomar algumas decisões próprias, decisões essas necessárias para executar as suas tarefas com eficiência, precisão e segurança.

A mobilidade de um robot é uma questão relevante para o seu nível de autonomia. Um robot com a capacidade de se deslocar livremente num dado ambiente pode realizar de forma eficiente muitas tarefas que de outra forma lhe estariam vedadas. A localização aparece então como essencial para a autonomia dos sistemas robóticos.

A visão aparece como solução fácil para a localização já que as câmaras são sensores intrinsecamente seguros e de custo acessível, especialmente porque este género de sensor pode ser partilhado para diversas funcionalidades. A multiplicidade de informações que se podem retirar dos sistemas de visão coloca-os assim como sendo sensores de importância primordial para a construção de sistemas robóticos. O processamento dos dados provenientes da visão deve, no entanto, obedecer a limites temporais bem definidos de acordo com a dinâmica do sistema e do ambiente.

O estado do sistema deve ser resultado da fusão de toda a informação disponível. O sistema de fusão de informação deve tirar o máximo proveito de toda a informação válida disponível, entrando em consideração com a caracterização prévia de cada tipo de medida.

1.2 Contexto

Este trabalho surge na sequência do grande envolvimento do autor na área da robótica móvel. Destaca-se a participação regular do autor em duas ligas de futebol robótico da Federação RoboCup. As competições do RoboCup e as ligas de futebol robótico em particular são uma banca de ensaios padrão a nível mundial para o teste de sistemas robóticos. Ao caracterizar-se adequadamente um dado problema contribui-se para a focalização dos investigadores e desta forma incentiva-se o desenvolvimento científico e tecnológico. Acresce ainda que o ambiente do futebol robótico é adequado à validação experimental pois conduz a conclusões realistas que podem facilmente ser portadas para outras aplicações.

Diversas aplicações da sociedade civil beneficiam do desenvolvimento dos sistemas robóticos. Aplicações na área da Automação Industrial tal como os sistemas de produção e os sistemas de armazenamento automático, beneficiam de avanços nesta área. Outras aplicações civis emergirão ainda com os

desenvolvimentos nestas tecnologias tal como por exemplo os sistemas de entregas, de limpeza e de vigilância, etc.

1.3 Contribuições

Esta dissertação apresenta as seguintes contribuições de relevo:

- Apresentação abrangente da metodologia de projecto de sistemas robóticos;
- Análise de dois projectos reais em que o autor participou;
- Definição formal do codificador de janela deslizante;
- Algoritmo de auto-localização através de códigos de barras;
- Construção prática de um sistema de fusão de informação;
- Demonstração da viabilidade das soluções apresentadas para localização robótica em ambientes estruturados.

1.4 Estrutura

Este documento aborda em primeiro lugar o projecto de sistemas robóticos com elevada autonomia.

De seguida aborda-se o problema da localização, incluindo sistemas e métodos associados.

Segue-se a teoria necessária aos trabalhos apresentados. É abordada a modelização e calibração dos sistemas de visão e apresenta-se o Filtro de Kalman incluindo a sua versão Extendida.

Após breve contextualização relativa a códigos de barras e a codificadores, é realizada a apresentação formal do codificador de janela deslizante e do correspondente código de barras de janela unívoca.

Os dois projectos robóticos em que o autor participou mais recentemente são discutidos de seguida. São plataformas para duas ligas do futebol robótico do RoboCup, a liga F180 e a liga F2000. A validação experimental decorre na plataforma descrita para a liga F2000.

A nível experimental, apresenta-se uma série de algoritmos para extracção de características interessantes da imagem. As medidas obtidas são caracterizadas e

é construído um sistema de fusão de informação baseado num filtro de Kalman Extendido. O sistema de fusão de informação é simulado e depois validado em vários testes dinâmicos reais.

São depois apresentadas conclusões e propostas de melhoramentos futuros.

Como anexo inclui-se um dicionário de acrónimos para facilidade de consulta.

Capítulo 2

Projecto de Sistemas Robóticos

2.1 Introdução

Um **sistema robótico** é um sistema com alguma liberdade de actuação e autonomia de decisão. Estes sistemas incluem uma porção mais interventiva com o ambiente, o **robot** e podem também incluir diversos sistemas periféricos (figura 2.1). A palavra robot hoje em dia é, no entanto, utilizada em linguagem comum com significado alargado e os sistemas robóticos executam trabalhos que os seres humanos:

- não querem fazer – trabalhos sujos, repetitivos, etc.;
- não podem fazer – ambientes nocivos, radioactivos, etc.;
- fariam com pior qualidade, mais devagar ou mais caro.

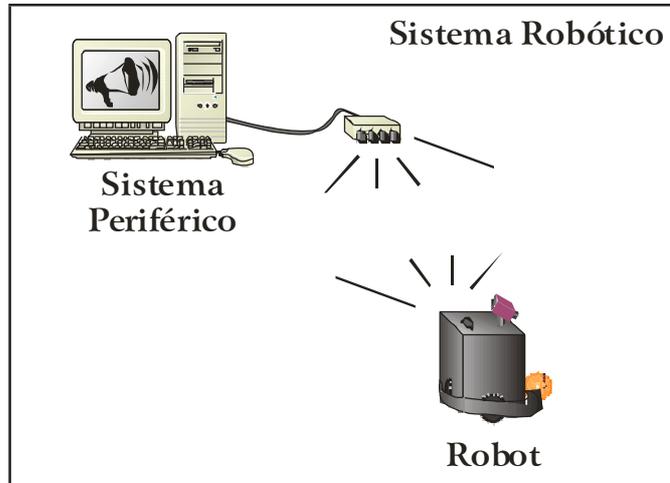


Figura 2.1 Ilustração dos constituintes do sistema robótico

Os sistemas robóticos assumem assim elevada utilidade para a humanidade e são motivo de grande trabalho de investigação científico-tecnológico. Estes sistemas assumem variadíssimas formas conforme a sua finalidade e o ambiente onde se inserem. Alguns exemplos comuns são:

- Um robot de produção fabril pode ter capacidade de transportar cargas de toneladas mas apenas segue linhas no chão;
- Um robot para cartografia submarina terá uma forma adequada ao meio onde se desloca;
- Um robot futebolista pode ser pequeno e rápido mas as suas “missões” têm curta duração;
- Um robot explorador do planeta Marte deve resistir às condições ambientais deste planeta e ainda poder deslocar-se autonomamente sobre a superfície do planeta.

Os exemplos apresentados levantam questões a diversos níveis pelo que é pertinente o levantamento dos requisitos do sistema que se pretende projectar.

2.2 Requisitos

Os requisitos a identificar para o projecto robótico devem incluir:

- Restrições de forma/tamanho/aparência exterior;
- Tipo de ambiente onde vai operar (mobilidade, dinâmica, nível de estrutura);
- Necessidades específicas às tarefas a realizar (carga, manipulação, etc.);
- Tempo e características da autonomia necessária;

- Capacidades de decisão autónoma/cooperação;
- Tipo de supervisão/dependência de elementos (informáticos ou humanos) exteriores;
- Canais de comunicações disponíveis e tolerância a falhas das comunicações;
- Questões de disponibilidade, de robustez, de tolerância a falhas (*fault tolerance*) e avaria segura (*fail safe*) – pode implicar algum grau de reconfigurabilidade interna;
- Questões de segurança para o ambiente que o rodeia (incluindo seres humanos) – tecnologias admissíveis para sensores e actuadores;
- Manutenção necessária;
- Decisões tecnológicas essenciais;
- Custo e duração de projecto.

Porventura a primeira questão de projecto é a forma do robot, que deve ser adequada à sua finalidade e ao meio que o rodeia. O meio ambiente não deve poder danificar facilmente o robot, nem o robot deve danificar de forma relevante aquilo que o rodeia. O robot pode também ter a necessidade de conviver em segurança com seres humanos.

A falha do robot pode ser motivo de projecto cuidadoso: um submarino autónomo não pode deixar-se afundar descontroladamente no oceano. Também a falha de uma parte do robot pode ser motivo de estudo cuidado. A falha segura do robot por vezes necessita de suporte a nível do projecto global do robot. Um detalhe importante é impedir da propagação das falhas, isto é, impedir que uma avaria num elemento do robot avarie outros componentes. Adicionalmente, o robot pode tomar medidas para limitar os estragos próprios e/ou tentar alguma forma auto diagnóstico da falha. Perante uma falha parcial, o robot pode ainda tentar contornar o problema e assim continuar os seus propósitos, ainda que de forma limitada ou diferente do plano inicial.

Como qualquer outro equipamento, convém considerar as intervenções necessárias a nível de manutenção. Será necessário identificar cedo na fase de projecto os consumíveis necessários e admissíveis para o funcionamento normal.

Será também importante considerar quais são os requisitos no que toca à disponibilidade¹ esperada do sistema. Não devem assim ser descuidadas questões relativas ao comportamento sob avaria. Questões tal como identificação da avaria, facilidade de reparação e disponibilidade de peças sobressalentes são assim relevantes durante todo o projecto.

A operação do sistema e a gestão das fronteiras do robot deve ser ponderado

¹ Disponibilidade= $(MTTF)/(MTTF+MTTR)$ – *MTTF Mean time to Failure* – tempo médio para avaria e *MTTR Mean Time to Repair* – tempo médio para a reparação

cuidadosamente incluindo o estudo de quais são as técnicas admissíveis para os sensores e actuadores assim como devem ser estudadas as necessidades de comunicações com o supervisor (humano) e os sistemas envolventes.

Como qualquer outro projecto de engenharia, terá certamente que se encontrar compromissos que permitam cumprir os orçamentos. O custo da engenharia de pôr todo o conjunto a funcionar deve então ser tido em linha de conta, quer a nível de custo, quer a nível de tempo de projecto.

2.3 Áreas de Projecto

Após o levantamento de todos os requisitos anteriores, será de admitir que a robótica é uma disciplina abrangendo diversas áreas científico-tecnológicas que atravessam a Mecânica, a Electrotecnia e a Informática. Procurando detalhar as áreas mais importantes para a robótica, pode-se identificar:

- Mecânica:
 - Sistemas de tracção;
 - Projecto mecânico:
 - *Chassis* e disposição dos subsistemas;
 - Espaço de carga (*payload*);
 - Arrumação e acessibilidade aos sistemas internos;
 - Sensores, Actuadores e manipuladores mecânicos;
- Electrotecnia e Informática:
 - Sistemas de alimentação (ex: baterias);
 - Sensores, actuadores e manipuladores eléctricos;
 - Electrónica:
 - Analógica (ex: condicionamento de sinal);
 - Digital (ex: sistemas de interface);
 - De potência (ex: comando de actuadores);
 - Comunicações:
 - Internas (ex: redes de sensores e actuadores);
 - Com o exterior (ex: redes sem fios);
 - Plataformas de processamento:
 - Sistemas Tempo Real (ex: comando de motor por microcontrolador);

- Outros sistemas de processamento computacional (ex: processadores);
- Hardware dedicado (ex: FPGAs);
- Teoria dos sistemas e controlo de sistemas dinâmicos;
- Processamento de sinal e de imagem;
- Sistemas Operativos e ferramentas informáticas;
- Decisão, Controlo e Inteligência Artificial

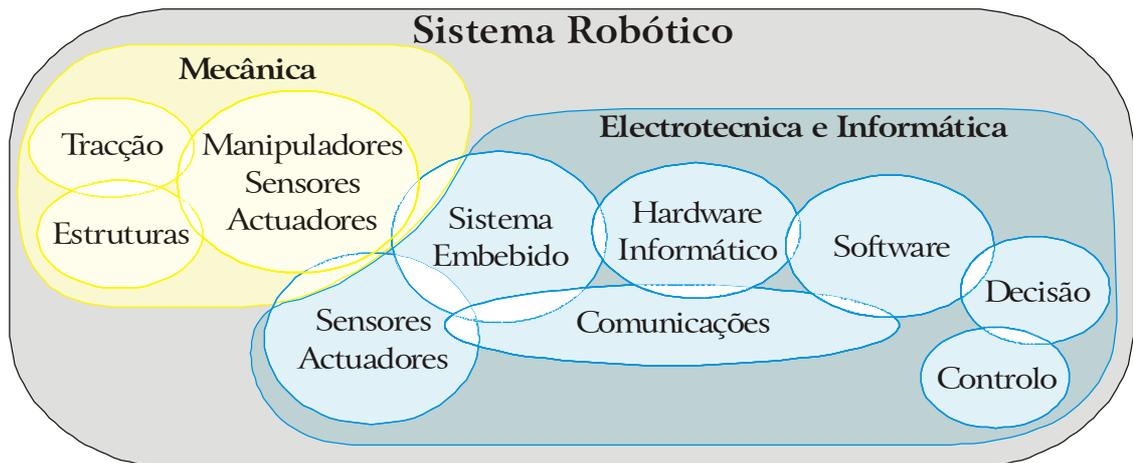


Figura 2.2 Representação simplificada das áreas científico-tecnológicas envolvidas em projecto robótico

Algumas áreas pertencem claramente à mecânica ou à electrotecnia ou à informática. Outras áreas porém, atravessam diversos domínios e poderão ser incluídas como questões onde o projecto deve ser partilhado de tal forma que a melhor solução para o conjunto seja atingida. Estas interacções são representadas de forma simplificada na 2.2.

A filosofia de projecto deste tipo de sistema é abordada de seguida.

2.4 Projecto Global e Mecatrónica

Uma das principais dificuldades do projecto robótico actual é a diversidade de áreas a abordar. É necessário portanto que a equipa de projecto implemente uma abordagem de conjunto. Desta forma problemas que surjam a dado nível podem ser resolvidos eficientemente noutra nível, com uma abordagem diferente. Estes conceitos gerais deram origem à Mecatrónica [1] que lida com o projecto de sistemas complexos. Um exemplo citado que segue estes conceitos é o do avião *Lockheed Martin/Boing F-22 Raptor* [1]. Um avançado sistema aviónico deste tipo beneficia de um projecto onde [1] «a abordagem aos problemas é realizada de forma concorrente (e não sequencial) resultando assim em produtos com mais sinergias». Generalizando, pode então dizer-se que o projecto de sistemas multi-

disciplinares beneficia da abordagem mecatrónica onde é dada elevada importância ao projecto integrado e às sinergias entre diferentes abordagens. Exemplos frequentemente referidos de áreas de aplicação deste conceito são as áreas da indústria automóvel e da aeronáutica. O projecto de um sistema robótico beneficia também desta abordagem globalizante.

A mecatrónica é geralmente descrita [1], [2] como abrangendo os seguintes campos todos em comum com os Sistemas Robóticos:

- Mecânica;
- Electrónica;
- Controlo;
- Computadores.

Cada um destes campos chave inclui uma série de questões pertinentes (ver figura 2.3).

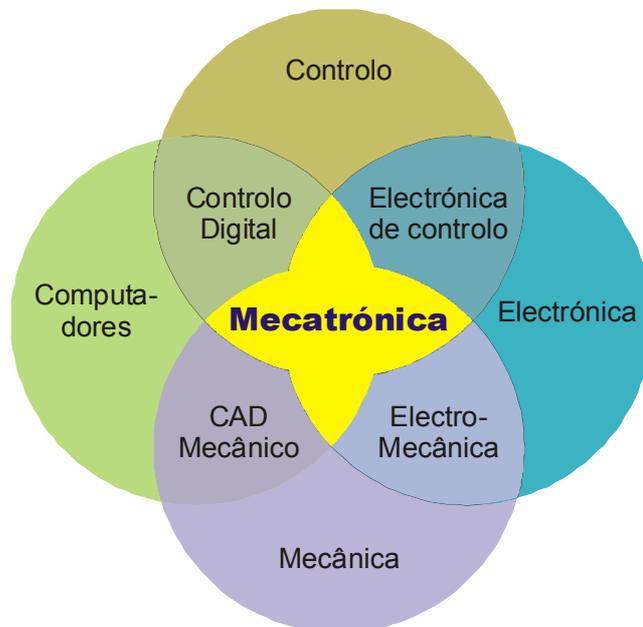


Figura 2.3 Conceitos de base na Mecatrónica, adaptado de [2]

Dentro das áreas dos computadores, do controlo e da electrónica são essenciais conhecimentos de modelização, processamento e análise de sinais, métodos numéricos, controlo de processos dinâmicos e ainda de comunicações e electrónica [2], [1].

A área da mecânica diz respeito à respostas dos materiais sobre o efeito de forças [1]. Inclui a mecânica dos corpos rígidos e deformáveis, leis relativas à mecânica dos fluídos e às leis do movimento.

A abordagem mecatrónica menciona a simulação com partes de hardware no *loop* de controlo, simulação *Hardware-in-the-Loop* [1]. Este tipo de teste valida a simulação e permite o teste por blocos do sistema em fases precoces do projecto.

Este tipo de simulações que compreendem sinais de várias naturezas procuram identificar problemas com a variabilidade de parâmetros entre subsistemas. Desta forma pode estudado e melhorado o grau de degradação de desempenho do conjunto, face à degradação de desempenho de uma das partes.

De acordo com as ideias preconizadas pela mecatrónica, a figura 2.4 resume o trajecto que um projecto dinâmico deve ter [2]. Utilizam-se modelos matemáticos e físicos para fazer o projecto evoluir antes da construção e teste do protótipo. A avaliação de cada simulação e cada teste leva a melhoramentos que depois são incorporados no projecto até que a avaliação do desempenho seja satisfatória e o projecto seja dado como concluído.

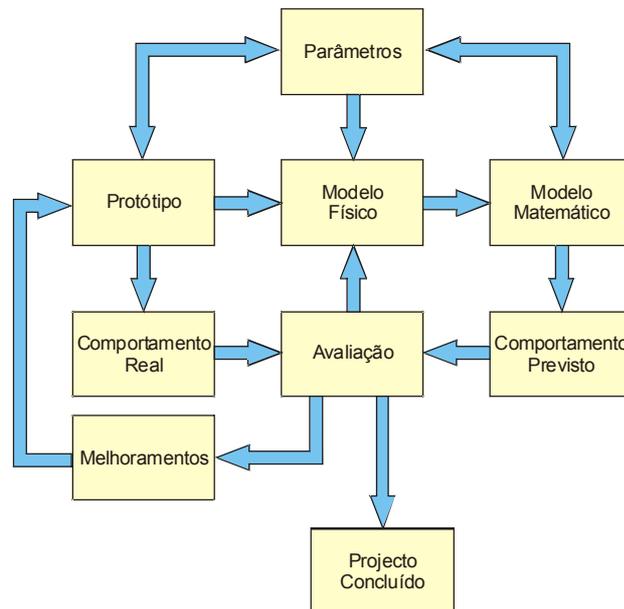


Figura 2.4 Dinâmica de projecto de acordo com os princípios da mecatrónica (adaptado de [2])

Uma vez estabelecida a filosofia que o projecto deve ter, estuda-se de seguida vários tipos de configurações para sistemas robóticos.

2.5 Tipo de Robot e sua Locomoção

Existem diversos tipos de robots com elevada variedade de funcionalidades. Uma questão essencial é a forma como o robot se deslocará, isto é, a configuração do seu sistema de locomoção incluindo motricidade e direcção.

Os veículos movidos por rodas convencionais são frequentemente utilizados em sistemas robóticos. As configurações de locomoção sobre rodas mais comuns são:

- Locomoção diferencial [3], [4] - figura 2.5:

- Duas rodas motorizadas independentemente;
- Um terceiro ponto de apoio dá estabilidade (pode ser uma roda ou apenas um apoio);
- Permite que o robot rode sobre si próprio (em torno do ponto C);

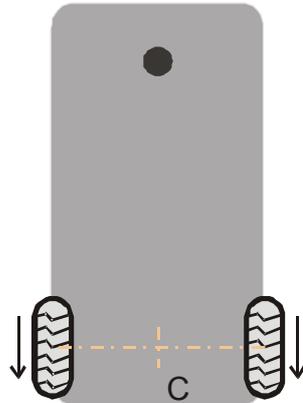


Figura 2.5 Locomoção diferencial

- Locomoção tipo triciclo [3] - figura 2.6:
 - A mesma roda dá direcção e motoriza o veículo;
 - Duas rodas livres atrás;
 - Curvatura em torno do ponto C;

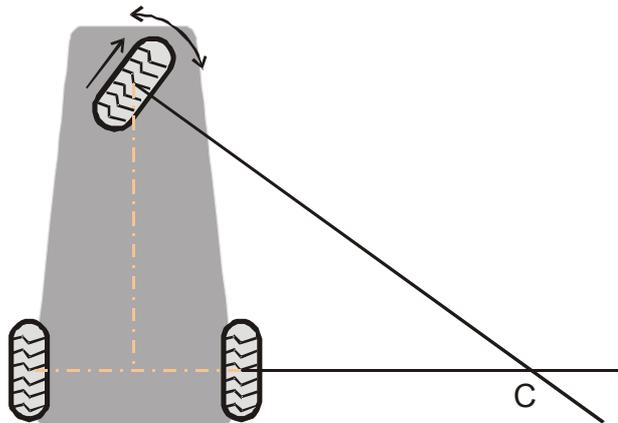


Figura 2.6 Locomoção tipo triciclo

- Locomoção Ackerman² [3] - figura 2.7:
 - Duas rodas guiam a direcção do carro;
 - Muitas vezes tracção nas rodas traseiras com diferencial mecânico;
 - Pode também ter tracção à frente nas rodas de direcção;
 - Permite apenas raio de curvatura limitado ao robot;

² Este é o tipo de direcção utilizada nos automóveis ligeiros comuns

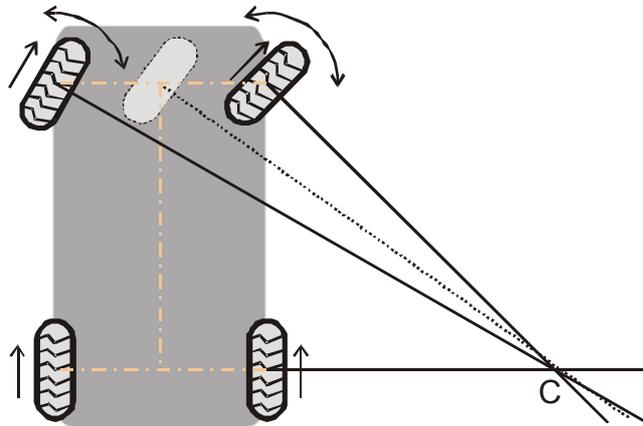


Figura 2.7 Locomoção tipo Ackerman

- Locomoção *Synchronous Drive* [3], [5] - figura 2.8:
 - Todas as rodas são motoras e direccionais;
 - Um conjunto de correias faz girar todas as rodas sincronizadamente;
 - Um outro conjunto de correias dá tracção a todas as rodas;
 - Pode mover-se em qualquer direcção após rotação das rodas.

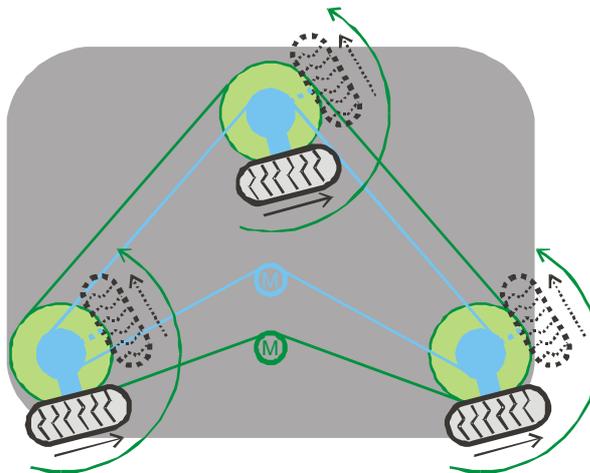


Figura 2.8 Locomoção do tipo "Synchronous Drive"

Existem ainda sistemas compostos por vários segmentos, segmentos esses que movimentam de forma independente apesar de ligados entre si. Estes veículos denominam-se com múltiplos graus de liberdade [3], [6] (fig. 2.10). Uma das principais vantagens destes veículos é que a sua configuração é adequada para transportar grandes cargas. A desvantagem é a maior complexidade no controlo dos vários segmentos do corpo com respectivas restrições mecânicas associadas.



Figura 2.9 “nBot” Robot que se equilibra em duas rodas [7]



Figura 2.10 Robot com múltiplos graus de liberdade [6]

Um robot diferente mas que utiliza rodas é o “nBot” [7] que pode ser visto na figura 2.9. Utilizando o princípio do “pêndulo invertido”, este robot consegue deslocar-se mantendo o equilíbrio em 2 rodas apenas, sem um ponto de apoio adicional.

Outros veículos movimentam-se através de lagartas [3], [8] (figura 2.11). Esta forma de locomoção permite deslocamentos sobre terrenos difíceis o que pode ser interessante por exemplo para aplicações de busca e salvamento.



Figura 2.11 Robot “BloodHound” para busca e salvamento [8]

Um outro veículo chamado *soryu* [9] tem lagartas e o corpo dividido em segmentos. Pode movimentar-se de diversas formas, arrastando-se ou “rebolando” o que lhe permite lidar com terrenos difíceis (ver fig 2.12). O robot *soryu* é também projectado para missões de busca a salvamento.

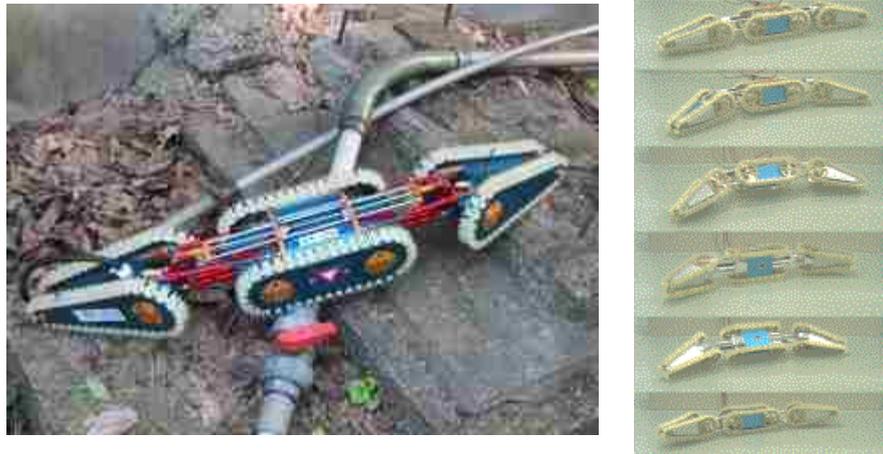


Figura 2.12 “Soryu”: à esquerda fotografia do protótipo e à direita o protótipo “rebola” por umas escadas [9]

Muitos sistemas recentes tem vindo a tirar partido da utilização de rodas omni-direccionais para permitir movimento omni-direccional. Cada roda deste tipo tem “rodinhas” na sua periferia desliza numa direcção e tracciona na direcção perpendicular. Utilizando 3 (ou mais) rodas deste tipo o veículo consegue movimentar-se em todas as direcções e mesmo rodar sobre si próprio simultaneamente [10], [11], [12]. Esta técnica tem tido muita aplicação recente em termos de competições de futebol robótico pela flexibilidade acrescida na movimentação [13], [14], [15], [16], [17]. Em termos esquemáticos, consulte-se a figura 2.13; nesta figura se as duas rodas superiores estiverem a ser motorizadas com igual velocidade, a terceira roda desliza parada e o robot movimenta-se tal como indicado (as rodas de tracção também deslizam enquanto traccionam).

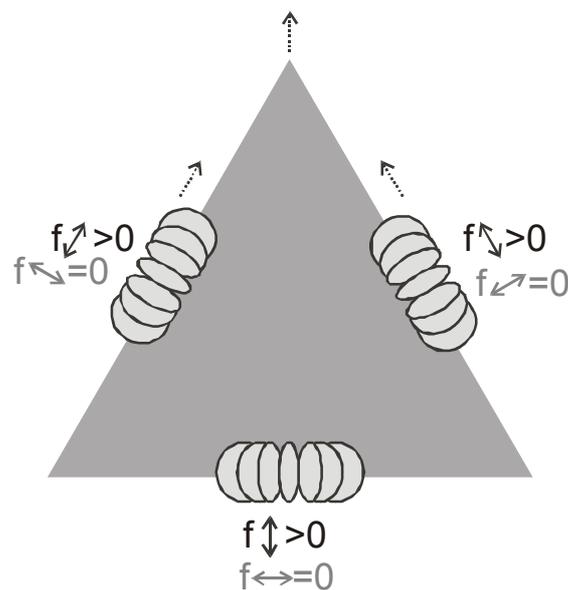


Figura 2.13 Locomoção OmniDireccional

Exemplos de robots com mobilidade omni-direccional podem ser vistos na figura

2.14 [13], na figura 7.4 pág. 110 [16] e na figura 7.7 pág. 114 [17].



Figura 2.14 À esquerda está um robot “protótipo rápido” com rodas omni-direccionais e à direita uma roda omni-direccional ampliada [13]

Há também outros tipos de veículos menos convencionais, que incluem formas estranhas e meios especiais de locomoção:

- “Minhoca totalmente aderente” - [18];
- “Esférico” - [19] [20] - ver figura 2.17;
- “Spiral Track” - [21] - figura 2.18.



Figura 2.15 Micro-robot voador (uma “mosca”) [22]



Figura 2.16 “Minhoca totalmente aderente” [18]



Figura 2.17 Robot esférico [19]



Figura 2.18 Robot “Spiral Track” [21]

Outros robots são diferentes porque pertencem ao mundo dos Robots Micromecânicos – MEMS – *Micro Electro Mechanical Systems*

- Versão Voadora [23], [22] - figura 2.15;
- Versão com pernas [23].

O também pequeníssimo robot “Alice” consegue jogar futebol [24]. Na verdade 3 contra 3 destes robots jogam numa folha A4 com uma bola de 2 cm (ver figura 2.19).



Figura 2.19 Robots “Alice”: à esquerda foto do robot (com escala); à direita estes robôs a jogarem futebol 3 contra 3 [24]

Existem muitas tentativas para investigar tipos de robots inspirados na biologia animal – Robots “Biomiméticos”. Existem diversas tentativas para imitar o andar com pernas e em especial o andar do ser humano, em duas pernas. A locomoção de Robots Humanóides e outros Bípedes é também alvo de complexo trabalho de modelização e investigação [25], [26]. Este tipo de movimento é complexo e existem relativamente poucos robots bípedes reais em movimento. É a investigação neste campo que se pretende incrementar com a liga de humanóides da Federação *RoboCup* (ver secção 7.2, pág 104).



Figura 2.20 Honda P3 e Asimo [27]



Figura 2.21 Humanóide Sony SDR4x [28]

Alguns humanóides conhecidos são:

- O humanóide da EPFL, Suíça [29];
- O Honda P3 e o ASIMO [27] - figura 2.20;
- O Sony SDR4x [28] - figura 2.21;
- No MIT, EUA o robot M2 [30].

Há também uma grande série de robots inspirados na biologia animal [31]:

- Robots “Com pernas” - *Legged*:
 - Escorpião [32];
 - Salamandra [33];
 - Dinossauro - *Troody* [34];
 - Cão [35], *Sony Aibo* [36] (figura 2.25);
 - Gato *Omron NeCoRo* [37] (figura 2.26);
- Formiga³ [38] e colónia de formigas [39];
- Cobra [40];
- Macaco (saltando entre barras no tecto) [41];
- Robots “Sub Aquáticos”:
 - Peixe genérico [42];
 - Tubarão [43] – figura 2.22
 - Lagosta [44] - figura 2.23;
 - Atum [45];
 - Lampreia [46];

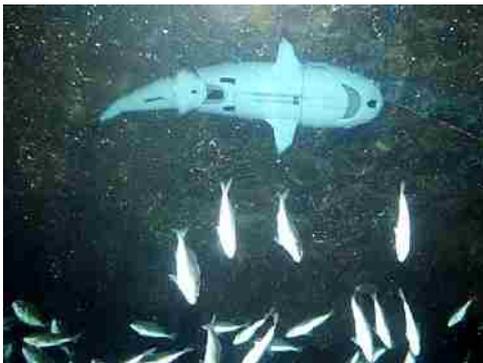


Figura 2.22 Robot tubarão [43]

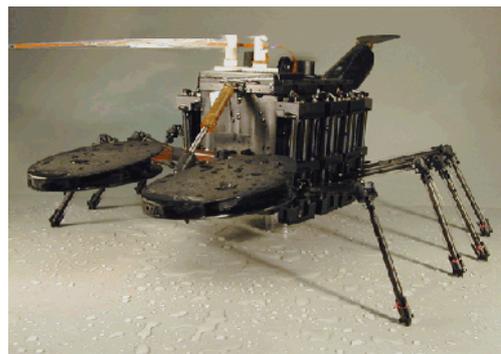


Figura 2.23 Robot Lagosta [44]

Um outro tipo de robot é a serpentina [47] [48]. Este robot pode locomover-se de diversas formas [47] (ver figura 2.24):

- *Twisting mode* – encurvamentos laterais fazem avançar o robot (tal como uma cobra!);
- Locomoção com rodas – as torções laterais fazem com que as rodas livres avancem tangencialmente às curvas do corpo do robot;
- Locomoção em ponte – o robot eleva-se quase na vertical e dobra-se na direcção para onde quer avançar de tal maneira que possa apoiar a outra

3 Esta “formiga” desloca-se por lagartas

extremidade mais à frente;

- Locomoção em anel – o anel deforma-se fazendo com que o centro de gravidade do anel mude e o anel avance rolando;
- *Inching mode* – movimento ondulatório na vertical que se propaga ao longo do corpo.

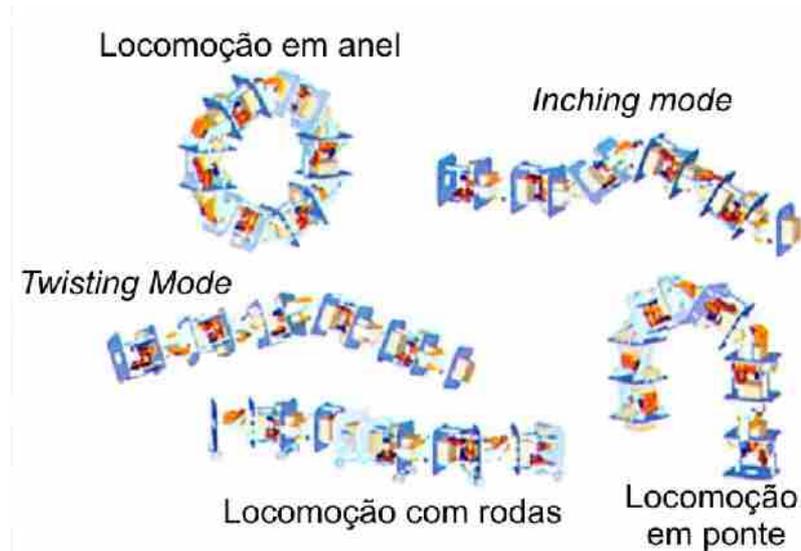


Figura 2.24 Movimento do robot do tipo serpentina (adaptado de [47])

A robótica é também motivo de interesse para a indústria dos brinquedos pelo que aparecem já versões comerciais de animais de estimação robóticos “*Robot Pets*” tais como os já referidos *Sony Aibo* [36] (figura 2.25) e o Gato *Omron NeCoRo* [37] (figura 2.26).



Figura 2.25 Cão robótico “Aibo” modelo ERS220, da Sony [36]

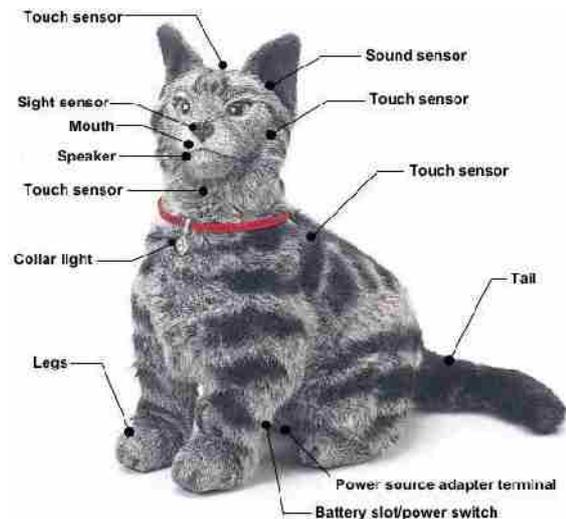


Figura 2.26 Gato robótico Necoro da Omron [37]

Os robots submarinos podem ter vários aspectos menos inspirados na biologia. Exemplo disso mesmo é o robot “*Infante*” [49] e uma imagem virtual deste robot

submarino está na figura 2.28. A propulsão destes sistemas é frequentemente conseguida através de hélices adequadas, tais como as do *Infante*, mostradas nas figuras 2.30 e 2.29 [50], [51]. Na figura 2.27 pode-se ver o robot *Isurus* [52] que possui também apenas uma hélice propulsora. Neste robot, a direcção de movimento é alterada através da orientação de duas barbatanas horizontais e duas barbatanas verticais situadas na popa do robot.



Figura 2.27 Robot Isurus [52]



Figura 2.28 Render do robot submarino Infante



Figura 2.29 Hélice dentro do propulsor do Infante



Figura 2.30 Propulsor do Infante

Sabendo qual a tipologia adequada para a aplicação em causa, pode-se dar início ao projecto mecânico propriamente dito.

2.6 Projecto Mecânico

O projecto da parte mecânica do sistema robótico é fundamental para o bom funcionamento global de conjunto. Devem ser considerados diversos factores relativos às necessidades específicas de cada sensor, cada actuador e cada manipulador. Também a disposição mecânica das partes dentro do robot e eventualmente da sua carga devem ser considerados. O projecto das várias partes elementares do robot não deve perder de vista as restrições dimensionais existentes do conjunto bem assim como o peso das partes e do conjunto. A

acessibilidade de cada parte do robot é também um facto a ter em consideração, visto que pode ter impacto no tempo efectivo de funcionamento conforme a dificuldade de diagnóstico e reparação de avarias.

A utilização de ferramentas de projecto assistido por computador denominadas de ferramentas CAD – *Computer Aided Design*, são essenciais para um eficiente projecto mecânico. As ferramentas do tipo CAD 3D possibilitam o projecto de formas e espaços gerando uma simulação do resultado sob diversas perspectivas de uma forma facilmente perceptível para toda a equipa de projecto. Também uma alteração dimensional não exige que o projecto de conjunto seja refeito. A definição de peças elementares é também possível permitindo assim representações hierárquicas de peças. Estes sistemas facilitam também muito a geração de cotas para fabrico. Os melhores sistemas CAD podem ser ligados de forma automática a sistemas CAM *Computer Aided Manufacturing*, para fabrico automatizado. Por vezes o conjunto deste tipo de sistemas é referido como um sistema CAD/CAE/CAM⁴, conjunto de ferramentas informáticas para ajuda ao projecto, à engenharia de fabrico (simulação de fabrico, etc.) e ao fabrico automatizado [53]. As metodologias CAD/CAE/CAM permitem a prototipagem rápida, isto é, permitem a construção de um protótipo mecânico funcional em pouco tempo.

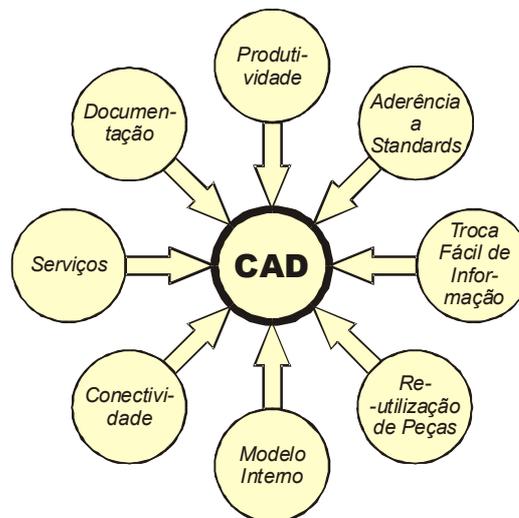


Figura 2.31 Benefícios provenientes da utilização de sistemas CAD

Tal como ilustrado na fig. 2.31, os sistemas CAD promovem uma série benefícios [54]:

- Produtividade aumentada (redução de custos);
- Facilidade de visualização dos modelos e promovem a sua completa e correcta documentação;

⁴ CAD/CAE/CAM – Computer Aided Design/Computer Aided Engineering/Computer Aided Manufacturing

- Facilita o cumprimento de *standards*, o que mais tarde se reflecte em facilidade de interpretação e melhor documentação do projecto;
- Permite a utilização de serviços tais como, por exemplo, a geração automática de dados, validação do modelo, listagem de peças, etc.;
- Permitir a ligação a sistemas de fabrico simulado e/ou geração de programas de fabrico para sistemas de maquinagem de comando por computador (também designadas por máquinas de comando numérico).

Este tipo de ferramentas de auxílio ao projecto leva a que seja também fácil implementar conceitos de facilidade/adequação à montagem e fabrico das peças elementares do sistema sob projecto.

Os sistemas CAD 3D utilizam diversos tipos de estruturas internas para guardar a informação [55], [54]:

- Representação das periferias dos elementos a modelizar (ex: planos 3D, métodos analíticos, etc.);
- Partição do espaço em volumes elementares (*voxels*) preenchidos ou não;
- *Ray-Tracing* – registar modificações em cada raio elementar que atravessa o volume do universo;
- Modelação Sólida (intersecção e reunião de elementos geométricos 3D básicos).

Uma das mais divulgadas ferramentas CAD foi utilizada pelo autor desta dissertação para o trabalho mostrado na secção 7.3, página 106. Esta ferramenta é o *PTC Pro Desktop* e é de utilização totalmente livre na versão utilizada [56]. Esta ferramenta, em concreto, é caracterizada como sendo uma ferramenta CAD de sólidos 3D, paramétrica, *feature based* e permite, por exemplo:

- Modelizar cada peça do sistema de forma paramétrica: introduzindo medidas, definindo intersecções, etc.;
- Efectuar o ajuste mecânico das várias peças e elementos no conjunto de acordo com conceitos tal como: distâncias, ângulos, alinhamentos, etc.;
- Utilizar peças sob vários níveis de hierarquia;
- Obter a listagem de todas as peças utilizadas;
- Fazer cotagens para fabrico de forma muito facilitada (incluindo planos de cortes, vistas de detalhe, etc.);
- Testar configurações de peças em várias posições e peças deslizantes;
- Localizar interferências (sólidos sobrepostos);
- Calcular peso e centro de massa;
- Construir animações.

Alguns exemplos do trabalho realizado com esta ferramenta podem ser encontrados nas figuras 2.32, 3.3, 4.14, 4.15 e 7.2.

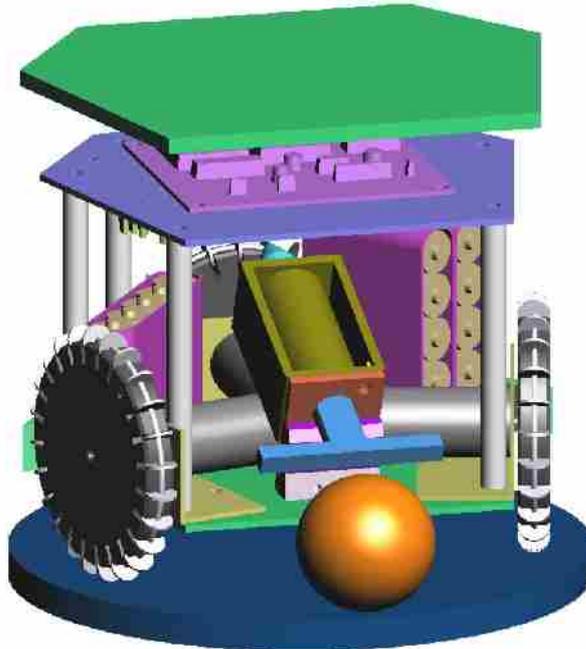


Figura 2.32 Modelo de CAD do conjunto do robot 5dpo, F180, ano 2003 (comparar com fig. 7.4)

O estudo dos espaços no robot é ainda mais valioso se existirem restrições dimensionais apertadas.

Após o projecto mecânico, estudam-se de seguida as fontes de energia disponíveis para um sistema robótico.

2.7 Sistema de Alimentação

Alguns sistemas robóticos podem conviver com limitações de mobilidade inerentes à sua fonte de energia ser externa. Se for viável o robot depender de energia exterior, então esta energia necessita apenas de ser condicionada e distribuída. Os robots de maior autonomia não podem, porém, ter cabos nem tubos que os liguem permanentemente ao exterior. O robot deve então ter algum nível de independência em termos energéticos: ele deve ser capaz de armazenar ou produzir energia a partir do seu interior ou a partir do ambiente. Em termos tecnológicos, é complexo transmitir energia para o robot se movimentar sem fios, em segurança e com rendimento aceitável. O trabalho [57] faz uma tentativa interessante de transmitir energia sem fios em ambiente industrial mas não refere qualquer aplicabilidade em robótica. É então incontornável, actualmente, que o robot seja tenha algum sistema de armazenamento de energia ou que a obtenha a partir do ambiente circundante.

A escolha das fontes de energia primárias devem ter em conta vários factores que são independentes da natureza da energia em causa:

- Características de carga e descarga (tempos e intensidades);
- Características de descarga própria, de envelhecimento e de deterioração;
- Questões de manutenção e armazenamento;
- Rendimento das transformações envolvidas;
- Quantidade de energia acumulada e questões volumétricas (densidade, disponibilidade em forma adequadas, etc.);
- Preço inicial e durabilidade (número de ciclos, envelhecimento, etc.);
- Restrições mecânicas (peso, dimensões e questões de arrumação);
- Segurança e restrições ambientais;
- Restrições quanto à recarga da energia a acumular;
- Interferência e facilidade de condução e adaptação da energia.

A alimentação eléctrica para um robot dos dias de hoje é incontornável. É, no entanto, possível a inclusão de mais que um tipo de energia primária. Para além da energia eléctrica, o ar comprimido é muitas vezes utilizado para accionamentos mecânicos em robótica [58]. É necessário, no entanto, suportar o acréscimo de restrições mecânicas envolvidas e a complexidade adicional para recarregar o sistema com formas de energia adicionais. Uma vantagem possível pode ser a adequação da fonte energética a determinadas partes do sistema (por exemplo determinados actuadores).

A distribuição de energia dentro do robot segue os conceitos resumidos na figura

2.33. Nesta figura pode-se identificar acumuladores e geradores internos ao robot. Estes acumuladores internos necessitam de ser recarregados de alguma forma enquanto que os geradores transformam energia do exterior ou de fontes de armazenamento intermédias internas ao robot. Qualquer conjunto de acumuladores e geradores pode ligar a qualquer conversor de energia. Alguns subsistemas podem também aceitar energia directa de geradores e/ou acumuladores. Alguns conversores e subsistemas podem utilizar diferentes conjuntos de energia provenientes de diferentes conjuntos de acumuladores e geradores.

Um exemplo comum de um acumulador de energia eléctrica é uma bateria. Como exemplo de um gerador que “retira” energia do ambiente refira-se um painel foto-voltaico que transforma luz ambiente em energia eléctrica. Uma célula de combustível é um exemplo de um gerador que transforma matérias primas em energia eléctrica, produzindo como subproduto água. Qualquer gerador interno pode também reabastecer uma outra forma de energia interna, por exemplo um painel foto voltaico pode também carregar baterias.

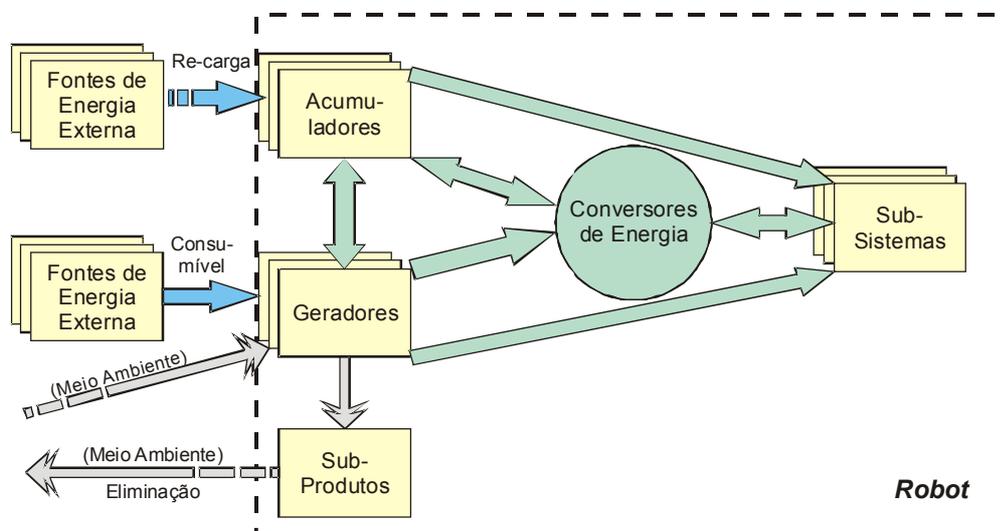


Figura 2.33 Distribuição de energia dentro do robot

Os sistemas computacionais do robot necessitam de energia eléctrica com características adequadas. A solução mais frequente é a de obter uma determinada tensão nominal e depois utilizar conversores estáticos para adequar a energia eléctrica para distribuição pelos diversos subsistemas. Alguns exemplos podem ser consultados na secção 7.3, pág. 106 e na secção 7.4, pág. 110. Valores de tensão utilizados frequentemente são 12V e 24V. A escolha do(s) valor(es) exacto(s) a utilizar é tecnológica. Uma outra solução possível é transformar a tensão de alimentação das baterias na tensão standard da rede para depois obter todas as tensões necessárias internamente ao robot. Tal procedimento acarreta uma elevação de tensão para depois ser posteriormente reduzida, com conseqüente redução no rendimento eléctrico do conjunto. Esta opção é justificável se se considerar que as perdas energéticas são aceitáveis face

a outro tipo de ganhos possíveis pela utilização de equipamentos disponíveis de forma generalizada ao consumidor – COTS (*Components-Off-The-Shelf*). A utilização de tensões elevadas dentro do robot poderá criar problemas de segurança eléctrica e de isolamento.

Existem várias soluções de armazenamento de energia eléctrica sob a forma química, para utilização em unidades móveis [59], [60], [61]. Na tabela 2.1 estão resumidos as principais características das baterias mais utilizadas. Existem ainda outros tipos de baterias adaptados a partir de pilhas não recarregáveis que são pouco utilizadas para sistemas autónomos [59].

A utilização de motores de explosão enfrenta graves limitações quer a nível de segurança quer a nível da poluição. Principalmente por estes motivos são pouco interessantes para aplicações em veículos autónomos, ainda mais se estes alguma vez entrarem em ambientes fechados.

Tabela 2.1 Comparação de Baterias, adaptado a partir de [59], [60] e [61]

	<i>NiCd</i>	<i>NiMH</i>	<i>Chumbo/Ácida</i>	<i>Iões de Lítio</i>	<i>Iões de Lítio/Polímero</i>
Densidade Energética (Wh/Kg)	45-80	60-120	30-50	110-160	100-130
Resistência Interna (mΩ) - incluindo circuitos periféricos (a determinada tensão)	100-200 6V	200 - 300 6V	<100 12V	150 - 250 7.2V	200 - 300 7.2V
Ciclo de Vida (80% cap. Inicial)	1500	300-500	200-300	500-1000	300-500
Carga Rápida	1h	2-4h	8-16h	2-4h	2-4h
Tolerância à sobrecarga	Moderada	Baixa	Alta	Muito Baixa	Baixa
Tensão Nominal da Célula	1.25V	1.25V	2V	3.6V	3.5
Corrente de Carga Rápida	20C ⁵	5C	5C	>2C	>2C
Corrente de Carga Lenta	1	0.5	0.2	<1C	<1C
Temperatura de Descarga (°C)	-40 – 60	-20-60	-20-60	-20-60	0 - 60
Utilização Comercial Desde	1950	1990	1970	1991	1999
Toxicidade	Elevada	Menor	Elevada	Menor	Menor
Reciclagem	Obrigatória	Anconselhável	Obrigatória	Anconselhável	Anconselhável
Questões Adicionais	Efeito de Memória ⁶	Efeito de Memória Menor	Pode libertar Hidrogénio	Envelhece sem utilização	Envelhece sem utilização
Algumas Utilizações Típicas	Ferramentas sem fios, aplicações médicas		UPSs ⁷ , Auto-móveis	PCs Portáteis, Telemóveis, Câmaras Vídeo	

Um outro sistema de armazenamento eléctrico são os condensadores, que acumulam carga eléctrica. São muito utilizados como sistemas intermédios pois permitem elevadas correntes apesar da sua capacidade de armazenamento de energia ser limitada quando comparada com as baterias (ver Fig. 2.34). As características da utilização de vários tipos de condensadores pode ser consultado em [62].

5 O valor C é a corrente numericamente igual à capacidade da bateria, ex: capacidade=1Ah => C=1A

6 Diminuição precoce da capacidade de acumulação de energia; este efeito é reversível após descarga completa e carga completa

7 UPS – *Uninterruptible Power Supply* - Fonte de Alimentação Não Interruptível

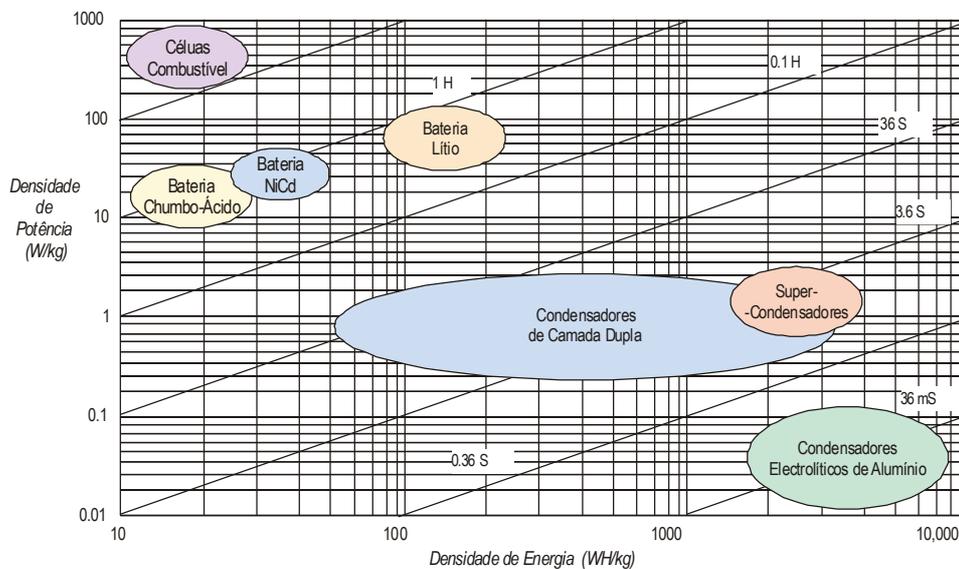


Figura 2.34 Comparação da densidade energética de diversas soluções para armazenamento e produção de energia eléctrica – as linhas diagonais são para descarga em tempo igual para determinada carga – adaptado de [63]

As células de combustível são uma alternativa recente às baterias para veículos autónomos. Oferecem maiores densidades energéticas (ver Fig. 2.34) à custa de consumo de um combustível: estes equipamentos não armazenam energia eléctrica, produzem-na. O subproduto da reacção química inerente à célula de combustível é a água. A procura de veículos com emissões nocivas nulas leva a que já existam carros e autocarros utilizando estas tecnologias. Existe por exemplo uma frota experimental de autocarros utilizando células de combustível e consumindo hidrogénio [64], [65] onde a propulsão é conseguida através de motor eléctrico. A utilização de células de combustível enfrenta os seguintes obstáculos [64],[65],[66]:

- Consumo de um combustível;
- Produção de água;
- Podem existir problemas de segurança;
- Problemas de reciclagem visto utilizarem-se substâncias muito especiais e por vezes muito nocivas na construção destes equipamentos.

Apesar destas questões, as células de combustível são interessantes pois permitem densidades de energia superiores às das baterias (ver figura 2.34), sendo portanto relevantes para área dos sistemas robóticos.

Após o estudo do projecto mecânico e do sistema que fornece energia ao sistema robótico, estuda-se todo o equipamento embarcado dentro do robot.

2.8 Sistema Embebido

Designa-se por **sistema embebido** o conjunto de sistemas embarcado no robot que lida com as interfaces robot-ambiente e processamento dos respectivos sinais. Este sistema lida com a conversão de energia de e para o exterior incluindo sensores, actuadores e respectivos circuitos de condicionamento. A aquisição de sinal dos sensores faz parte deste sistema bem assim como o seu pré-processamento, eventual conversão e eventual disponibilização adequada, por exemplo, numa rede interna ao robot. Também a geração de sinais adequados aos actuadores é responsabilidade deste importante subsistema do robot.

2.8.1 Sistemas Tempo Real

Um sistema **tempo real** (*Real Time* – RT) é um sistema em que a correcção do resultado não depende apenas da correcção lógica da computação mas depende também do tempo em que essa computação foi completada. Se o limite temporal (*dead line*) não foi cumprido então diz-se que ocorreu uma **falha** do sistema [67], [68].

O sistema embebido é um sistema que está sujeito a restrições de Tempo Real visto que o atraso de processamento na zona de interface robot-mundo cria atrasos na execução de ordens e/ou atrasos na percepção do ambiente e compromete ainda o ciclo de decisão e controlo.

2.8.2 Classificação de Sensores

Os sensores podem ser classificados quanto à forma como interagem com o ambiente [55], [69]:

- **Sensores passivos** não adicionam energia ao ambiente circundante durante o seu funcionamento podendo, no entanto, de alguma forma retirar energia do ambiente;
- **Sensores activos** enviam energia para o meio e medem essa interacção com o ambiente.

Um exemplo de um sensor activo é o radar que envia energia para o meio e mede o tempo que decorre até à chegada do respectivo eco. Uma câmara é um sensor passivo no sentido em que a sua medida é baseada em energia que não é emitida por si.

Outra classificação para os sensores diz respeito à proveniência dos dados medidos. A este respeito, utiliza-se a classificação proveniente dos sensores biológicos [55]:

- **Sensores exteroceptivos** – lidam com o mundo externo;

- **Sensores proprioceptivos** – medem o estado interno do sistema.

Uma câmara que visualiza o exterior do robot é um sensor exteroceptivo porque traz dados do meio ambiente externo ao robot. Um codificador ligado ao veio do motor do robot é considerado um sensor proprioceptivo porque mede dados internos ao robot.

Os **sensores** ditos **inteligentes** podem ter um determinado nível de auto-calibração e/ou podem adaptar-se automaticamente ao meio circundante. É esperado que cumpram a sua função da melhor forma possível e que os seus resultados sejam mais interessantes que os fornecidos pelo sensor elementar (não “inteligente”). Um exemplo de um sensor inteligente seria um sensor de distância com base em Ultra-Sons que apresenta na saída uma distância já corrigida tendo em conta a variação da velocidade do som com a temperatura ambiente.

2.8.3 Sensores Comuns em Robótica

Os subsistemas de sensorização utilizados em sistemas robóticos muitas vezes são compostos por sensores e sistemas adicionais. Alguns subsistemas e respectivos mecanismos adicionais e/ou variantes frequentemente utilizados são:

- Sistema de Visão – câmaras:
 - Matriz ou linear;
 - Preto e branco ou a cores;
 - Sistemas de orientação (*Pan & Tilt*);
 - Sistemas de ampliação óptica (*zoom* motorizado comandável);
 - Sistemas utilizando iluminação estruturada;
- Ultra-Sons:
 - Sistemas de orientação;
 - Medida de efeito *Doppler*;
- Sensores ópticos de Infra Vermelhos (IR - *Infra Red*):
 - Medida de intensidade reflectida;
 - PSD⁸ (*Position Sensitive Device*) – sensor de iluminação IR estruturada em que a distância ao sensor é obtida por triangulação;
- Radares - *RADAR* - *RA*dio *DE*tectio *AN*d *RAN*ging – medida de distância através da contagem de tempo entre a emissão e o retorno do eco (*time of flight*); é também efectuada a medida de velocidade do alvo através do efeito de *Doppler*;

8 Ver página 60 e figura 3.7

- LIDAR – *LIght Detection and Ranging* – análogo ao RADAR mas utilizando *Laser* [70];
- Sensores de contacto – microinterruptores por exemplo aplicados em pára choques – *bumpers*;
- Sensores de torção [13];
- Acelerómetros;
- Bússolas;
- Inclínómetros e pêndulos;
- Giroscópios;
- Codificadores (*encoders*) absolutos ou relativos.

Determinados subsistemas de sensorização podem fazer uso de múltiplos princípios elementares de medida, procurando assim uma medida com melhores características que a medida isolada. Podem até receber comandos de configuração para que os resultados do subsistema tenham o máximo de informação útil. Exemplos deste tipo de sistemas são os sensores configuráveis, que podem ser considerados subsistemas de sensorização. Um exemplo de um sensor deste tipo é uma câmara com um sistema de orientação.

2.8.4 Actuadores

Os actuadores eléctricos e os actuadores pneumáticos são muito frequentes em sistemas robóticos. Os actuadores eléctricos são muito utilizados principalmente devido à sua facilidade de comando e flexibilidade. Os actuadores pneumáticos permitem, no entanto, forças muito elevadas, superiores às que se conseguem com facilidade utilizando actuadores eléctricos.

Os motores eléctricos rotativos são quase omni-presentes para motorização de veículos móveis. Existem também motores eléctricos lineares que podem ser utilizados para actuadores robóticos, embora a sua utilização seja menos frequente. Os motores mais frequentemente utilizados são motores de corrente contínua e os motores de passo motores de passo-a-passo (*Stepper Motors*). Estes últimos são mais utilizados para posicionamento de equipamentos [17] mas podem também ser utilizados para tracção [71], beneficiando assim do seu menor custo. A desvantagem dos motores de passo-a-passo é a necessidade de cuidados adicionais para manter a estabilidade durante o seu funcionamento. Os motores eléctricos de corrente contínua são a solução mais frequente para tracção de veículos móveis. Devido às características próprias destes actuadores de alta rotatividade, muitas vezes utilizam-se caixas desmultiplicadoras para obter maiores binários. É também possível utilizar motores de corrente alternada mas a sua utilização em sistemas robóticos é menos frequente devido à complexidade acrescida do seu comando e funcionamento.

Frequentemente são também utilizados actuadores electromagnéticos. Em [15], [16] e [17] são utilizados solenóides como dispositivos para chutar a bola em competições de futebol robótico. Para obter forças elevadas, utilizam-se condensadores para armazenar carga eléctrica com elevada tensão e assim permitir elevadas correntes de descarga durante um tempo limitado mas suficiente para impulsionar a bola. Para a mesma aplicação são também utilizados actuadores pneumáticos [58] e sistemas mecânicos à base de energia acumulada em molas [5], [72].

2.8.5 Implementação de Subsistemas

Cada subsistema do robot deve traduzir uma repartição em blocos coerentes. Todos os sistemas devem possuir uma interface bem documentada com o exterior e com as outras partes do robot.

Na implementação prática destes subsistemas existe muitas vezes discussão acerca da possibilidade de executar determinadas tarefas por electrónica dedicada ou por software [73]. Esta é uma decisão tecnológica importante que acarreta consequências a nível dos tempos de resposta que o sistema terá com implicações ao nível do Tempo Real. Realizar tarefas por software, por exemplo dentro de um microcontrolador (MCU), pode ser uma solução flexível e de baixo custo. A alternativa é a implementação do sistema em hardware dedicado que permitirá tempos de processamentos muito menores.

A nível de hardware dedicado, as principais soluções tecnológicas são:

- FPGA – *Field Programmable Gate Array* – sistema digital baseado em ligações e blocos reconfiguráveis;
- FPGAA - *Field Programmable Gate Analog Array* – análogo ao anterior mas permitindo blocos analógicos;
- ASIC – *Application Specific Integrated Circuit* - integrado (não programável) projectado especificamente para a aplicação.

As soluções utilizando sistemas de processamento via software são:

- MCU – *Micro-Controller Unit* - processador com periféricos destinados a controlo tempo real e blocos programáveis;
- MPU – *Micro-Processor Unit* - processador destinado a processamento genérico de dados;
- DSP – *Digital Signal Processor* – processador geralmente contendo barramento de dados e de endereços de largura diferente, destinado a processamento de dados intensivo contendo alguns periféricos programáveis.

Existe ainda a possibilidade de integrar o hardware e software:

- SoC – *System on a Chip* – FPGA + MCU/DSP.

Independentemente de determinada escolha tecnológica para um certo bloco do sistema, uma boa prática a adoptar é tomar medidas para que mesmo na presença de ordens incoerentes (do software de alto nível) ou na ausência de ordens, os sistemas garantam sempre a sua própria integridade, isto é, garantir a avaria segura do sistema.

A implementação de blocos com requisitos Tempo Real utiliza frequentemente um processador dedicado ao bloco em causa. Tal solução faz com que os tempos envolvidos em cada operação sejam fáceis de determinar e caracterizar e dessa forma garantir o funcionamento temporalmente correcto.

Por vezes os subsistemas menos exigentes acabam por crescer em complexidade e pode ser proveitoso a utilização de um Sistema Operativo Tempo Real adequado à plataforma onde decorre a implementação.

2.8.6 Comunicações Internas

A comunicação mais frequente dentro de sistemas é, possivelmente, RS232(C). Este tipo de ligação é apenas ponto a ponto mas é amplamente divulgado para comunicações dentro de sistemas e é virtualmente omni-presente. Existem diversas variantes desta norma, sendo talvez a mais utilizada a norma RS485 que permite comunicação em rede.

Os mais recentes MCUs oferecem, no entanto, diversas possibilidades de comunicação dentro do mesmo integrado. Muitos incluem mesmo suporte para protocolos de rede algo complexos. Estas **redes denominadas como redes de sensores e actuadores** permitem distribuir sinais de forma simples pelo interior de um sistema. Estas redes apresentam algum nível de suporte para funcionamento sob restrições de Tempo Real. Alguns exemplos de redes de sensores e actuadores integradas em MCUs são:

- CAN – *Controller Area Network* – ex: Siemens SAB80C167C;
- I2C ou TWI – *Two Wire Interface* – ex: Atmel ATMega8;
- USB/USB2 – *Universal Serial Bus* – ex: Atmel AT89C5131;
- *Ethernet*⁹ – ex: Dallas DS80C400

Após o estudo físico do sistema, o estudo seguinte diz respeito à estratégia que o robot utiliza para cumprir os seus objectivos, em particular, ser capaz de chegar ao seu destino.

⁹ Ethernet não foi inicialmente pensada para esta finalidade mas pode, ainda assim, ser uma solução muito interessante para esta classe de aplicação

2.9 Navegação

As perguntas fundamentais da **navegação** de um sistema robótico móvel são:

“Onde estou?”, “Para onde vou?” e “Como chegar lá?”

Estas questões não são de resposta simples quando se pretende elevada autonomia. “Onde estou?” implica medidas face ao ambiente exterior que pode ser dinâmico. Saber para onde se deve deslocar o robot para melhor cumprir a sua tarefa pode também ter solução não trivial. Seguir numa certa direcção pode não garantir chegar a certo ponto mesmo que ele fique nessa direcção pois o caminho pode estar bloqueado. Complexidade adicional aparece se se tiver em atenção considerações de segurança tal como quando o robot se desloca em ambiente dinâmico. Especial cuidado é necessário se o espaço é partilhado com seres humanos.

A navegação envolve diversas capacidades como por exemplo o contorno de obstáculos e actualização dinâmica do mapa actual. As técnicas mais utilizadas para navegação são [3], [74], [75]:

- Utilização de “mapas de estradas” de segmentos utilizáveis;
- Decomposição em sub-blocos do mapa actualizando blocos com informação ocupado/livre mais recente;
- Campos de Atracção/Repulsão (obstáculos repelem o robot que é atraído pelos pontos onde pretende chegar).

A navegação pode ser considerada como a plataforma intermédia que faz cumprir a ordem de chegar a um certo destino. O nível de autonomia que se pretende dar a esta camada intermédia de decisão é muito variável conforme a aplicação. Pode acontecer que um dado problema de navegação não tenha solução perante dado nível de autonomia concedida a este sistema. Neste caso novas decisões de nível superior são necessárias para que o robot atinja os seus objectivos.

Considerando que é possível uma localização satisfatória para o problema em causa, o problema seguinte prende-se em fazer com que o robot se desloque para cumprir os seus objectivos. É então necessário saber onde o robot está, para onde deve seguir e como o conseguir, isto é, é necessário que o robot navegue através do caminho escolhido até ao seu destino.

A arquitectura típica de um sistema de navegação pode ser decomposta tal como mostra a figura 2.35. No nível superior de decisão encontra-se geralmente o planeamento de tarefas que gera comandos para o sistema de planeamento de trajectórias. O sistema de contorno de obstáculos actua muitas vezes sobre o nível de controlo de movimento. Sistemas mais complexos podem fornecer a informação de obstáculos aos níveis superiores de decisão. A gestão do desvio de

obstáculos complicados em sistemas mais evoluídos pode exigir o traçado de novas trajectórias.

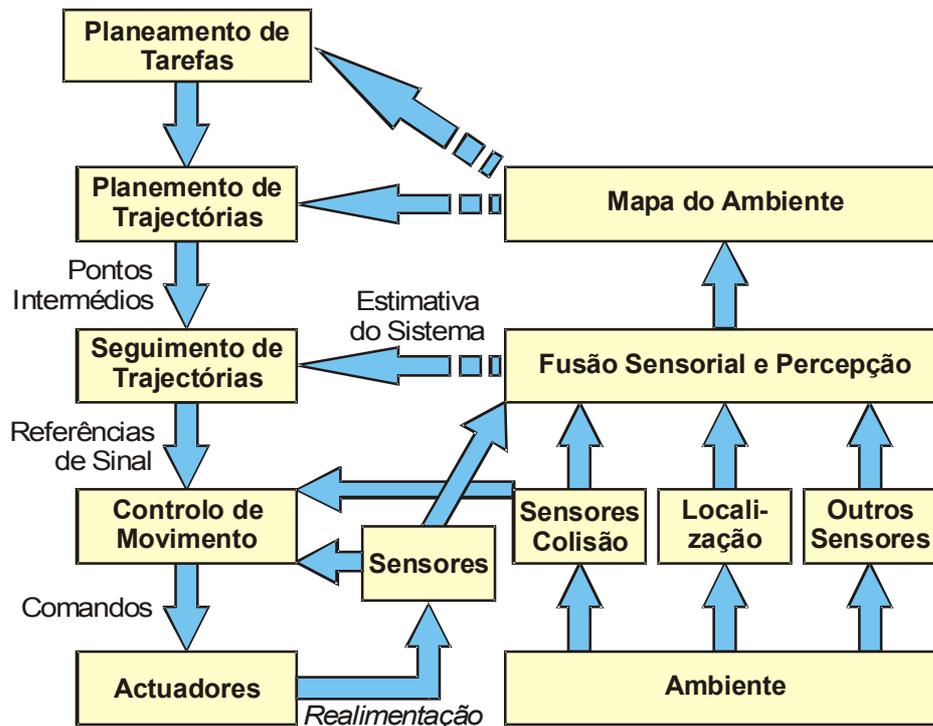


Figura 2.35 Arquitectura típica de um sistema de navegação

Sistemas com este tipo de arquitectura permitem fechar a malha de comando a vários níveis o que permite por sua vez:

- Contorno e desvio de obstáculos – a baixo nível;
- Fusão de dados para criar uma estimativa global do sistema – a nível intermédio;
- Actualização do mapa corrente – a nível superior.

Tipicamente a navegação acaba por ser abordada como sendo o seguimento de referências intermédias até alcançar destinos finais onde são atribuídas novas tarefas.

A camada de software da navegação recebe ordens do nível hierárquico superior, que é apresentado de seguida.

2.10 Decisão, Autonomia e Cooperação

A arquitectura do software de nível mais elevado é crítica para o desempenho satisfatório do robot que se pretende projectar. Chame-se ao software com mais elevado nível de abstracção o software de **decisão**. Tal como ilustrado na figura

2.36, o módulo de **decisão** toma as suas decisões com base em informações do mundo real fornecidas pelo bloco de entradas do robot e muda esse mesmo estado do mundo por intermédio das saídas do sistema. No bloco das entradas cabem subsistemas de sensores e respectivo tratamento de sinais enquanto que ao bloco das saídas dizem respeito os actuadores e respectivo condicionamento.

O bloco de entradas deve ser entendido como um bloco que reúne a geração de todos os dados necessários à produção de decisões pelo respectivo bloco e não apenas como um conjunto de sensores. Este bloco incluirá possivelmente informação relativa à fusão de informação de sensores e do estado estimado do robot.

O bloco de saídas reúne todos os sub-sistemas que implementam as decisões do módulo superior. Este bloco incluirá possivelmente malhas de controlo em tempo contínuo e a geração de sinais de referência e comando sobre os actuadores.

O funcionamento dos vários blocos do robot é claramente paralelo e síncrono com a realidade, havendo assim diversas restrições de tempo real.

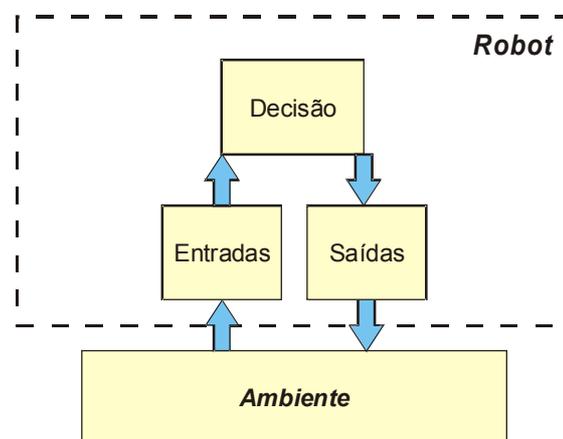


Figura 2.36 Blocos de entradas, saídas e decisão dentro do robot e respectivo relacionamento com o ambiente exterior

O software de decisão deve então basear-se em diversos outros blocos de software de nível inferior para criar o comportamento de alto nível desejado para o robot [76].

Diversas considerações são importantes relativamente à arquitectura do software de alto nível do robot para que este seja útil e adequado ao problema em causa [76]:

- Divisão de tarefas/modularidade/suporte para paralelismo;
- Adequação tecnológica à plataforma real (ao nível dos sensores, sensores configuráveis, actuadores, plataformas computacionais, comunicações, etc.);
- Adequação ao meio envolvente e às funcionalidade pretendidas;

- Robustez do software (mesmo em situação de avaria parcial);
- Qualidade e tipo de ferramentas existentes para a implementação;
- Capacidades de configuração durante a execução;
- Avaliação do desempenho – as metas temporais relativas ao comportamento em Tempo Real são atingíveis? Os dados para a tomada eficaz de decisões estão disponíveis? A estrutura permite flexibilidade para todas as tarefas a realizar serem “bem realizadas”¹⁰?

As considerações mais importantes que o software de decisão necessita de ter em conta são [76]:

- Como guardar informação acerca do seu próprio estado;
- Como guardar informação acerca do ambiente exterior;
- Como tomar decisões.

As abordagens mais frequentes à decisão em sistemas robóticos são [76]:

- Puramente reactiva – sem planeamento a prazo, para cada condição há uma reacção programada, sem estado interno do sistema;
- Hierárquico – controlo e planeamento utilizando diferentes níveis de abstracção; a informação do mundo é utilizada para gerar sequências de comandos a cumprir;
- Controlo híbrido – combina as duas abordagens anteriores num sistema hierárquico onde o nível inferior é reactivo e o nível superior produz sequências de acções;
- Comportamental – incluem partes reactivas mas utilizam representações complexas do mundo; utiliza-se por vezes coordenação, competição e concorrência para decidir a lista de acções a tomar.

Em termos de implementação, uma questão relevante é como incluir na arquitectura do robot a existência de várias tarefas para diversos subsistemas do robot pois pode existir a necessidade de gerir a coordenação/conflito entre diversos subsistemas. Como exemplo, considere-se um robot autónomo de transporte onde um braço do robot deve depositar um objecto numa caixa vazia. Este sistema hipotético utilizaria uma câmara que deve ser orientada para o local a depositar o objecto. Se este local não estiver vazio, o robot e a câmara devem receber ordens de alto nível para o robot se deslocar e visualizar a caixa seguinte até aparecer um local vazio. A visão deve depois fornecer dados relativamente à manobra de descarga do objecto que o braço do robot vai realizar.

O suporte para acções em caso de falha de um elemento do robot deve também ser previsto a nível de arquitectura de software. Poderá ser importante que o

¹⁰ Segundo [76] “Bem realizadas” pode ter métricas tal como o tempo que cada tarefa demora a ser completada, consumo de energia, nível de cooperação atingida, etc.

próprio robot tome medidas para limitar os danos e pode ser interessante ele conseguir algum tipo de auto diagnóstico da sua falha.

A nível de implementação, grande parte dos robots utiliza PCs embarcados no robot para implementação das camadas superiores de decisão em sistemas robóticos [17], [72], [15], [58]. São amplamente utilizados os Sistemas Operativos genéricos Windows e Linux. Menor número de aplicações utiliza sistemas operativos mais adequados a desempenho em Tempo Real tal como Tornado, QNX, LinxOS, VxWorks, Aperios e ainda diversas variantes Tempo Real do Linux.

Projectos que necessitem de segurança adicional podem ainda utilizar autómatos programáveis para tarefas críticas [5], à custa de um preço acrescido.

2.10.1 Cooperação entre Sistemas Robóticos

A cooperação entre diversos sistemas robóticos é uma tarefa complexa. A figura



Figura 2.37 Robots cooperando (Collective Robot [77])

Os aspectos positivos da utilização de sistemas de vários robots independentes mas cooperantes são [76]:

- Quando é possível decompor em sub-tarefas, o paralelismo é vantajoso e melhora o desempenho global do sistema;
- Algumas tarefas só são possíveis em conjunto (ex: 2 robots a transportarem um viga comprida);
- Sensorização e acção distribuída;
- Tolerância a falhas individuais por substituição do elemento em falha.

As desvantagens de utilizar uma equipa de robots são:

- Interferências, colisões;

- Custo de comunicação e robustez – a falha de comunicações leva a que a equipa não coopere e interfira com prejuízo para o conjunto;
- Acréscimo de complexidade para lidar com a incerteza da intenção de outro robot devido a falta de conhecimento ou falha de comunicação,
- Custo do sistema: dois robots poderão custar mais que um só. Se for possível utilizar vários robots mais simples para desenvolver a tarefa então o custo global pode ser menor, caso contrário maneira o custo será uma desvantagem.

Para exemplos de arquitecturas de controlo, consulte-se ainda a secção 7.2, pág. 104 e a secção 7.4.11, pág. 116.

2.10.2 Tentativa de Indiferença ao Hardware

O software de alto nível em robótica deve caminhar em direcção à independência ao hardware. O encapsulamento de tarefas deve ser efectuado de tal modo que esta afirmação seja levada em consideração. Dado o elevado ritmo do progresso tecnológico na área dos sistemas robóticos, é, em geral, interessante criar algum tipo abstracção ao hardware, criando assim uma camada de abstracção ao hardware – *Hardware Abstraction Layer (HAL)*. Desta forma, a troca de hardware por outro equivalente pode ser conseguida sem grande esforço. Dado o elevado nível de interdependência das tecnologias envolvidas na robótica tal pode não ser conseguido na totalidade mas o conceito, mesmo tomado de forma parcial, é ainda interessante.

Uma solução possível é modularizar o software em duas grandes partes. Uma parte inclui a sensorização e a actuação, enquanto que a outra parte compreende a decisão [16],[17]. O exemplo discutido na secção 7.3, pág. 106, apresenta mesmo as duas aplicações a funcionar em sistemas operativos diferentes, comunicando por rede. A troca de dados entre as partes deve utilizar sempre interfaces e linguagens bem definidas. Sempre que possível, a troca de dados deve ser conseguida de forma independente da implementação. Pretende-se assim limitar as dependências do hardware ao módulo da sensorização e actuação. A modularização proposta apresenta as vantagens adicionais de passar a existir um ponto importante de *debugging* e de permitir que as metades do sistema sejam simulados separadamente (simulação com hardware no *loop*, tal como advogado na secção 2.4, pág. 9).

2.10.3 Fusão de Informação

Uma possível definição para Fusão de Informação” [78] é:

«**Fusão de Informação** é um quadro formal de trabalho onde se emprega meios e ferramentas para a junção de dados de origens diversas com o objectivo de obter

informação de qualidade superior. A definição exacta de qualidade superior depende da aplicação em causa».

A fusão de informação pode ser classificada em 3 casos [55], [79]:

- Sensores Complementares – sem conflito;
- Sensores Competitivos – com o objectivo de obter redundância;
- Sensores Cooperativos – cooperam para gerar informação, por ex: visão stereo.

A fusão de informação é um processo segundo o qual a utilização de vários sensores permite obter elevados níveis de qualidade da informação. Podem ser utilizados vários sensores COTS menos caros e obter redundância adicional na informação. Pode também ser essencial recolher dados espalhados no espaço. A existência de múltiplos sensores é a única maneira de assegurar decisões correctas na presença de dados errados, desde que existam dados em quantidade suficiente. A nível de exemplo deve ser referido que no caso de ter 3 relógios para saber as horas, uma média não é uma boa medida de fazer a fusão de informação. Para uma melhor fusão de informação é essencial a caracterização de cada uma das medidas e em particular a incerteza relativamente a cada uma delas.

A caracterização de cada um dos sensores é então essencial:

- Precisão, Exactidão e Sensibilidade;
- Resposta dinâmica;
- Caracterização da eventualidade de leituras erradas;
- Caracterização do comportamento sob avarias.

A fusão de informação é utilizada em muitas áreas entre as quais [55]:

- Aeronáutica (controlo tráfego aéreo, IFF¹¹, ILS¹²);
- Tecnologias de fabrico industrial (ex: robótica e veículos móveis);
- Sensorização remota (ex: sensorização climatérica);
- Controlo de tráfego (automóvel, marítimo, etc.);
- Navegação autónoma/semi-autónoma;
- Aplicações médicas e ambientais;
- Aplicações militares (vigilância, defesa, etc.).

As mais comuns ferramentas para a fusão de informação são [55], [79]:

- Limites explícitos de confiança e decomposição em Árvore de regiões;

¹¹ *Identify Friend our Foe* – sistema militar automático de combate que detecta se outra aeronave é hostil ou aliada

¹² *Instrument Landing System* – Aterragem (assistida) por instrumentos

- Probabilidades e *Dempster – Shafer* (Métodos *Bayesianos*);
- Métodos Estatísticos/Estocásticos – Filtros de Kalman;
- Lógica Difusa¹³ - *Fuzzy Logic*;
- Métodos heurísticos e outros métodos no âmbito da Inteligência Artificial.

Este trabalho utiliza a técnica dos Filtros de Kalman (KF - *Kalman Filter*) para a fusão de informação. Esta é uma ferramenta com sólidos fundamentos teóricos na teoria das probabilidades e nos métodos estocásticos. Tais fundações permitem a modelização de conhecimento parcial do sistema sem perder a solidez do tratamento matemático (ver capítulo 5.4, pág. 84). O KF utiliza a noção matemática da covariância para representar a qualidade da representação do estado do sistema.

Existe uma variante dos KF designada de filtro de informação. Este filtro utiliza os mesmos princípios do KF numa versão numericamente mais adequada a fusão distribuída de dados, também interessante quando o estado inicial do sistema é totalmente desconhecido. Este filtro utiliza internamente os inversos das covariâncias, o que traz benefícios para representação de covariâncias infinitas durante o tratamento de ausência de informação [80].

Até ao momento fez-se o estudo da parte física do robot e a estrutura de tomada de decisões. Mencionou-se também que é necessário fazer comunicar as duas partes e referiu-se como. De seguida é tratado o problema da interface com o mundo exterior.

2.11 Interface e Comunicações com o Exterior

O interesse de ter uma comunicação permanente com o robot é manter a supervisão sobre a plataforma autónoma. Comandos podem ser enviados e recebidos de e para o robot, por exemplo, o robot pode enviar informações para coordenação com outros meios ou pode relatar eventos importantes.

Os recentes avanços nas tecnologias das redes sem fios permitem manter de forma económica comunicações com plataformas móveis.

Os standards mais utilizados para comunicações no âmbito da RF (Rádio-Frequência) com plataformas autónomas são:

- 802.11b – 11 Mb/s @ 2.4 GHz¹⁴;
- 802.11g – 54 Mb/s @ 2.4 GHz;

¹³ Por vezes também designado por lógica imprecisa

¹⁴ Banda dos 2.4-GHz é apelidada de *ISM - Industrial, Scientific and Medical*

- 802.11a – 11 Mb/s @ 4.9 GHz ~ 5 GHz¹⁵;
- *Bluetooth* – 723kb/s @ 2.4 GHz;
- Protocolos proprietários nos 151-173MHz, 418MHz, 433MHz, 868-870MHz e 902-920MHz¹⁶.

Outra técnica de comunicação sem fios é a utilização de IR [71] nomeadamente respeitando a norma IRDA – *Infra Red Data Association*. Este tipo de comunicações é de baixo custo mas sofre de interferências graves em contacto com barreiras físicas e tem por essa razão domínios de aplicação próprios.

É também possível descarregar parte do processamento para um computador não embarcado¹⁷. Neste computador não embebido, todo o poder de cálculo dos mais recentes processadores estará disponível¹⁸ uma vez que não existem limitações de consumo eléctrico. Este ganho no poder computacional aparece à custa de um peso superior nas comunicações e atrasos na chegada dos dados com acrescidos problemas para cumprir metas temporais no âmbito do desempenho em Tempo Real. Se o robot estiver permanentemente dependente de comunicações, interferências neste sistema levam a que o robot fique temporariamente inoperacional.

As comunicações são essenciais tanto para o funcionamento do sistema como o seu desenvolvimento. O desenvolvimento do sistema é discutido de seguida.

2.12 Ferramentas de Ajuda ao Desenvolvimento

A utilização de um cabo de comunicações especial para ajuda ao desenvolvimento era muito frequente no passado. Este cabo de comunicações denominado de “cordão umbilical” permite descarregar uma grande quantidade de informação para fora do robot para melhorar o desenvolvimento. Nesta situação, os dados seriam recolhidos e armazenados ao longo da missão e depois descarregados através do dito cordão umbilical. Com o aumento da largura de banda disponível nos modernos sistemas de comunicações sem fios, esta noção deixou de ser utilizada tão frequentemente.

Um simulador para o sistema é uma ferramenta muito importante para poder

¹⁵ Esta banda é apelidada de *UNII - Unlicensed National Information Infrastructure*; No Japão a frequência é de 4.9 GHz

¹⁶ A legislação varia de país para país e varia assim o espectro legal – possivelmente estas frequências não serão utilizáveis todas em simultâneo em país algum do mundo

¹⁷ Os robots da equipa 5dpo-2000 adoptaram esta filosofia nos anos de 2000 a 2002

¹⁸ Um processador Intel Pentium 4 a cerca de 3 GHz consome uma potência superior a 80W, o que pode ser difícil de sustentar em sistemas móveis alimentados por acumuladores eléctricos

testar a programação de alto nível antes de embarcar o software a testar. É também importante uma maneira de registar dados importantes para o sistema que está a ser desenvolvido e idealmente o ficheiro de registos (*log-file*) deveria poder ser executado de novo até que o ciclo de desenvolvimento esteja concluído.

Para além das ferramentas de ajuda ao projecto do sistema, será interessante prever ferramentas de auxílio ao *debugging* já a níveis funcionais e a diversos níveis de abstracção. Os simuladores de conjunto do sistema são ajudas preciosas ao desenvolvimento. Também os sistemas de armazenamento dos históricos são importantes para consulta dos detalhes dos motivos que levaram tal movimento a acontecer. A possibilidade de consulta remota deste tipo de dados é útil mas muitas vezes impraticável por motivos tecnológicos da quantidade de dados a transferir através de comunicações sem fios. Esses dados podem ser transferidos facilmente pelo “cordão umbilical”.

2.13 Conclusões

Apresentou-se neste capítulo os principais elementos do projecto de um sistema robótico. Este tipo de projecto é complexo e envolve diversas áreas científico-tecnológicas que incluem a mecânica, a electrotecnia e a informática.

A filosofia de projecto da Mecatrónica foi admitida como adequada ao projecto de sistemas robóticos pois o seu ciclo de desenvolvimento é apropriado ao projecto de produtos complexos.

A nível mecânico, mostraram-se diversos tipos de robots, diversos tipos de locomoção de robots e foram fornecidas indicações para um projecto mecânico saudável.

Os principais elementos eléctricos, electrónicos e informáticos foram ainda referidos. A escolha de soluções foi discutida para todos os elementos chave do projecto.

Os sistemas de fusão de informação foram ainda referidos como sendo capazes de melhorar a qualidade da informação que o sistema possui.

Capítulo 3

Sistemas de Localização

3.1 Enquadramento e Definições

A mobilidade é necessária para elevar o nível de autonomia de um sistema robótico. Dando liberdade de movimento ao robot, será então necessário localiza-lo, isto é, conhecer a sua posição no mundo. Pode assim dizer-se que localizar o robot significa conhecer todas as posições e ângulos relevantes ao sistema de eixos do mundo. Exactamente quais os parâmetros a determinar depende da aplicação em causa.

Considerando o caso típico de um robot móvel que se desloca num ambiente bidimensional então localizar o robot significa conhecer a sua posição cartesiana e orientação¹⁹ $[x \ y \ \theta]^T$ em coordenadas do mundo. Noutras aplicações localizar o robot pode ter outros significados. Por exemplo localizar um submarino autónomo pode significar encontrar a posição 3D e conhecer os 3 respectivos ângulos a cada eixo do mundo (chamados ângulos de *Euler*).

Encontrar sistemas de localização baratos e eficientes depende dos sensores disponíveis no robot, do poder de cálculo disponível e do nível de estrutura do ambiente onde o robot ser deslocará. O funcionamento “dentro de portas”

¹⁹ Orientação e atitude são sinónimos e dizem respeito ao ângulo de alguma parte do robot face ao exterior, tipicamente a sua direcção de movimento privilegiada, se existir alguma.

(*indoor*) ou “ao ar livre” (*outdoor*) é determinante para o conjunto de tecnologias disponíveis para a localização. Todos estes factores contribuem para que esta seja uma área altamente tecnológica onde as evoluções surgem a elevado ritmo.

Sendo verdade que a localização depende de sensores é interessante referir algumas técnicas genéricas e mencionar os sensores e respectivos princípios físicos utilizados. Algumas questões aqui abordadas relativamente a diversos sensores são complementares às considerações generalistas referidas na secção 2.8.2, pág. 29.

A abordagem que os seres humanos utilizam para a localização é a da visão: reconhecem pontos de referência (marcadores visuais) e movem-se em função do conhecimento da sua posição no mundo. Outros seres vivos utilizam sensores magnéticos ou Ultra-Sons para se moverem nos seus *habitats* naturais.

A medida remota a elevada distância debate-se com o complexo problema da elevada quantidade de energia a emitir ser segura para todos os elementos do ambiente em torno do robot e em particular segura para o ser humano. A utilização de sistemas laser e sistemas Rádio Frequência (RF) vem assim limitada por questões de segurança pois muitas vezes o princípio físico que permitiria fazer uma medida fácil envolve quantidades de energia não seguras.

Quando os sistemas dentro do robot permitem a sua **auto-localização**, ele fica a saber directamente a sua localização no mundo. Por vezes é o ambiente que fica a saber a posição do robot móvel a localizar e depois transmite essa informação, por exemplo através de comunicações sem fios. Neste caso está-se perante um **sistema de localização externo**.

3.2 Abordagens à Localização

3.2.1 Medidas Relativas

Os robots utilizam frequentemente sensores *proprioceptivos*²⁰ para fazer medida internas. Estas medidas são relativas ao estado anterior do sistema e assim a integração deste tipo de medidas acumula sempre um erro pequeno que cresce ao longo do tempo, inviabilizando a utilização prolongada desta técnica. Mesmo que o erro acumulado seja pequeno, haverá sempre perturbações externas que inviabilizam a utilização prolongada. É assim necessário recorrer a sensores que fazem medidas externas, isto é, relativas ao ambiente assim localizam o robot no mundo.

Um exemplo concreto de uma medida relativa frequentemente disponível é a utilização de codificadores ligados ao veio do motor. O objectivo é determinar

²⁰ Ver secção 2.8.2, pág. 29

deslocamentos do robot mas enfrenta-se o problema do escorregamento das rodas no pavimento. O sistema de medida de distancias percorridas pelo robot denomina-se por hodiometria²¹. Um refinamento também frequente é a utilização de uma roda livre suplementar para a medir o deslocamento no chão²² e não no veio do motor mas esta solução apresenta um acréscimo de complexidade mecânica permite ainda assim a existência de erros devido ao escorregamento da roda de medida sobre o pavimento. Mesmo perante uma excelente calibração destas medidas, ambos os sistemas resistem mal a pavimentos irregulares e a movimentos bruscos do robot tal como os resultantes de choques.

Sistemas mais complexos utilizam unidades inerciais que incluem bússola giroscópica e acelerómetros de qualidade para conseguirem manter fazer medidas fiáveis face ao estado anterior do sistema. Enquanto que uma bússola convencional mede o campo magnético terrestre, uma bússola giroscópica mantém uma qualquer direcção e é imune a interferências electromagnéticas [82]. A ideia por trás dos conjuntos inerciais é que a bússola giroscópica tem elevada precisão na presença de acelerações baixas e os acelerómetros complementam com uma elevada precisão perante acelerações elevadas [82].

3.2.2 Medidas Face ao Ambiente

Será então necessário obter medidas face ao ambiente de tal maneira que seja possível obter informação de localização não afectada de erros de integração inerentes a medidas relativas. É admissível que a medida “absoluta” de localização esteja apenas acessível de forma assíncrona pois nos instantes intermédios de tempo pode-se fazer a integração de medidas relativas para permitir que o sistema robótico continue a funcionar com uma degradação de características aceitável para a aplicação. É conveniente, porém, utilizar estratégias de fusão de informação para gerir os dois tipos complementares de informação que chegam em instantes que podem ser imprevisíveis.

Medir face ao ambiente significa encontrar algum ponto de referência conhecido, chamado de marcador (*landmark*). Os marcadores podem ser “naturais”, já existindo no ambiente ou “artificiais” no sentido em que foram colocados no ambiente para o propósito da localização do robot. O nível de estrutura do ambiente, isto é, se o ambiente tem marcadores ou não e a sua dinâmica, isto é, se o ambiente muda ou não ao longo do tempo, são questões relevantes pois o robot tem reconhecer um marcador do mundo e deve localizar-se relativamente à posição no mundo desse marcador que se presume conhecida com exactidão. Um erro na posição desse marcador não é porém relativa e não tem efeitos acumulativos para outras localizações desde que todas as medidas sejam efectivamente feitas em coordenadas do mundo. Uma outra questão importante é a identificação de qual o marcador avistado: pode ser utilizada

21 Hodiometria é a arte de medir as distâncias percorridas [81]

22 Esta é uma medida relativa mas que não provém de dentro do robot

informação anterior para presumir qual o marcador avistado ou a medida deve conter essa informação.

3.2.3 Metodologias Frequentemente Utilizadas

O princípio físico da detecção de marcadores pode variar entre técnicas ópticas, técnicas envolvendo ondas mecânicas, tecnologias RF ou combinações destas técnicas.

As medidas obtidas podem conter maior ou menor grau de informação. Alguns sistemas identificam o marcador em causa e outros não. A caracterização da medida é essencial para o algoritmo de fusão de informação que reúne os dados de todos os sensores presentes no sistema.

Diversos autores utilizam técnicas baseadas na medida de ângulos e/ou distâncias a marcadores cuja posição no mundo é conhecida. Se se utilizam apenas ângulos, o método é conhecido por **triangulação** [83], [84], [3], [75]. Se se utiliza ângulos e distâncias, o método é genericamente designado por **trilateralização** [83], [84], [3], [85], [86]. Como curiosidade refira-se que a triangulação de marcadores já é utilizada desde o século 12, por exemplo medindo o ângulo da estrela polar com o horizonte permite calcular a latitude do sitio onde é feita a medida (considerando a terra esférica e a estrela polar no eixo de rotação terrestre).

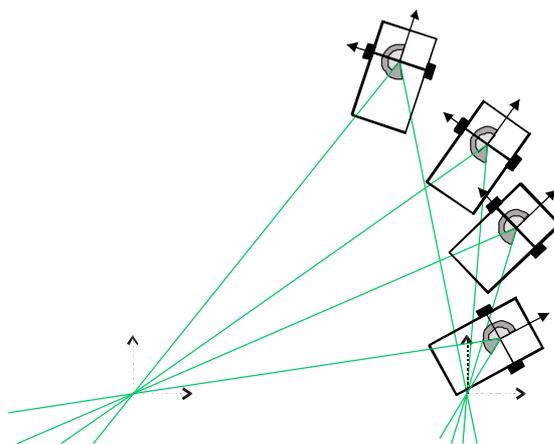


Figura 3.1 Dois ângulos internos ao robot não localizam um robot (mostram-se 4 localizações possíveis de um número infinito de soluções)

Devido ao facto de a localização ser um problema inverso, obter duas medidas angulares (dentro do robot) não o localizam, tal como ilustrado na fig. 3.1. Tendo 2 conjuntos de medidas ângulo e respectiva distância a um dado marcador localizam o robot num de dois pontos (fig.3.2). Se se souber que medida corresponde a que poste, a localização é conseguida sem ambiguidade.

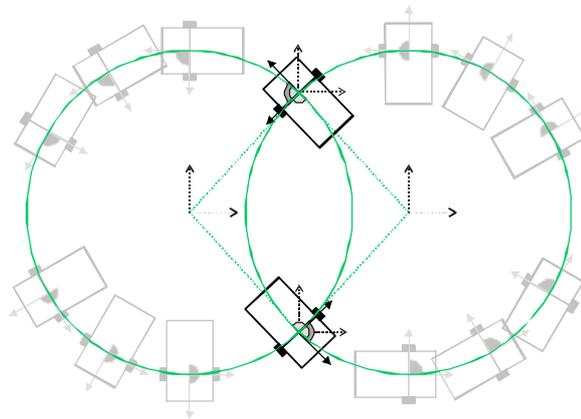


Figura 3.2 Não sabendo que medida diz respeito a que marcador, um robot com 2 medidas de ângulos e respectivas distâncias a marcadores fica com duas hipóteses de localização

Outra possibilidade é a utilização de várias distâncias a marcadores conhecidos. Tal acontece por exemplo nos sistemas de localização por satélite. Para este caso genérico, localizar em N dimensões, N medidas de distância resultam em duas soluções; $N+1$ medidas resultam numa solução única. Um exemplo tridimensional está na figura 3.3. Nesta figura pode-se identificar os pontos a uma distância conhecida de um marcador estando dispostos na calota esférica. A intersecção das 3 calotas esferas resulta em 2 pontos assinalados.

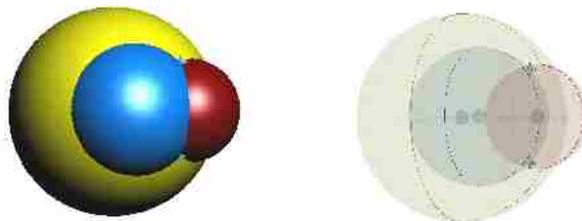


Figura 3.3 Intercepção de 3 calotas esféricas resulta nos 2 pontos assinalados com os marcadores de forma “bicuda”; à direita temos a versão translúcida da figura da esquerda onde as esferas pequenas representam os marcadores de localização (centros das outras esferas)

Para os casos em que é necessário “escolher” entre várias soluções qual é a interessante pode utilizar-se outras medidas ou conhecimentos anteriores acerca da localização.

A qualidade da localização depende essencialmente da precisão das medidas elementares necessárias ao seu cálculo. Problemas de erros de medida e de resolução limitada nas medidas elementares reflectir-se-ão necessariamente na qualidade de estimativa da localização.

3.3 Soluções de Localização

3.3.1 Soluções Envolvendo LASER

Muitos sistemas robóticos utilizam técnicas baseadas em LASER para medida do ambiente em torno do robot. Um fabricante reclama a interessante precisão de milímetro num alcance de 240 metros [87]. Muitas vezes estes sistemas aparecem associados em forma de *scanner*, isto é, sistemas que fazem medidas numa dada gama de direcções em torno do equipamento de medida. Outro produto comercial [88] menciona a precisão de 15 mm em 80 metros de alcance máximo ao longo de 180° numa resolução de 0.25°.

A utilização de marcadores naturais tais como paredes e janelas ou portas parece uma ideia interessante para conseguir a localização. Estes elementos podem ser caracterizados e identificados através do scanner laser. Após a aquisição de elementos geométricos, os dados recolhidos devem ser confrontados com o mapa conhecido [89], [90], [91], [92] para conseguir a localização.

O trabalho [93] apresenta também um método imaginativo de detectar o ângulo a um marcador exterior. Um laser é rodado até embater num espelho cuidadosamente desenhado de tal maneira que a reflexão incida de novo no robot. É utilizada triangulação para calcular a posição actual. Refere-se como resultados um erro menor do que 1 centímetro e menor que 0.8° em localização num espaço de 3m por 3.5m.

O sistema comercial “*LazerWay*” para utilização industrial no guiamento AGVs – *Automated Guided Vehicle* utiliza marcadores retro-reflectores que são triangulados através da incidência de um feixe laser [94], [95], [96]. Este método parece muito robusto, adequado para ambiente industrial desde que não exista excessiva poeira no ar.

3.3.2 Soluções Envolvendo RF

O sistema denominado *Active badge* [97] emite identificação Infra Vermelho todos os 10 segundos. Sensores colocados no meio ambiente detectam a localização do “*Active Badge*” (por sala). A escolha de Infra Vermelhos é apropriada porque acompanha topologia do ambiente (paredes e salas). Admite-se que o objectivo deste sistema é identificar a sala onde está o *active badge* desejado. Como exemplo de aplicação fala-se em transferir uma chamada telefónica para a sala certa dentro de um edifício.

O sistema VOR (VHF *Omni-directional Radio Ranges*) [98] é utilizado na aeronáutica. Este sistema permite triangulações utilizando RF. É possível determinar ângulo a uma estação VOR graças ao facto de esta estação emitir com uma modulação diferente para cada ângulo (emissão altamente direccional, cada grau sexagesimal tem valor diferente). Como existe uma emissão de

referência e a outra emissão é um sinal que varia ao longo de 360°, a diferença de fase entre as duas sinusóides recebidas é o ângulo do avião face ao VOR no mundo. Com ângulos para 2 VORs e um mapa é possível obter a localização em 2 D. Tal deve-se ao facto de a medida do VOR ser de um ângulo em coordenadas do mundo em não em coordenadas relativas (comparar com a situação da figura 3.1). Métodos direccionais tal como este são mais utilizáveis ao ar livre visto não existirem efeitos de reflexões múltiplas.

Outro sistema comercial é o “Pinpoint 3d-id” [99]. Este sistema é apropriado para seguir muitas *tags* em ambiente entro de portas (*indoor*). As *Tags* emitem a sua identificação e a infraestrutura sabe onde as *tags* estão. A estrutura emite pedido de localização para um dado dispositivo e a *tag* indicada responde passado tempo fixo com o seu identificador. A estrutura consegue determinar o tempo de voo subtraindo os tempos fixos conhecidos. A medida de tempo é sempre feita na estrutura (excepto os tempos “pequenos” na *tag*) o que resolve a questão da sincronização de vários relógios. A emissão da infraestrutura é a 2.4GHz e a *tag* responde a 5.8 GHz. Afirma-se conseguir uma precisão entre 1 e 3 metros.

Outro sistema chama-se “*Spot-on*” [100]. Este sistema baseia-se na intensidade do sinal RF recebido e utiliza *tags* já existentes no mercado. A medida da intensidade do sinal é feita através de um integrador/detector de pico no *transceiver* TR1000 da *RFMonolithics*, que funciona a 916.5MHz. A referência afirma que a localização 3D é conseguida dentro de um cubo de 1 m³.

Outro sistema baseia-se na medida da intensidade do sinal RF recebido num módulo standard *Radiometrix* [98]. Através da utilização do pino RSSI (*Received Strength Signal Indicator*) de módulos emissores da *Radiometrix* (neste caso de 418 MHz) a referência afirma ser possível estabelecer a lei da atenuação ao longo do espaço. Tendo vários *beacons* (antenas emissoras) a emitir separadamente no tempo e medindo a atenuação a cada emissor (que transmite a sua identidade), então é possível a auto-localização do receptor. Sistema viável ao ar livre – as ondas reflectidas causam excessivas interferência se existirem barreiras físicas. O erro de posicionamento obtido numa experiência com a área de posicionamento de 10 m por 10 m com *beacons* nos 4 cantos afirma-se ter distribuição Normal com média 1.83 metros e desvio padrão 1.07 metros sendo o erro máximo de cerca de 4 metros.

Existem também tentativas de modelizar a atenuação do sinal recebido em placas *Ethernet* sem fios. Identificando os endereços de hardware (*MAC Address*) das placas de rede emissoras torna-se possível a auto-localização. Este standard utiliza a frequência dos 2.4 GHz que é livre no mundo todo mas sofre de atenuações difíceis de prever pois nestas frequências os sinais facilmente assumem caminhos múltiplos. O trabalho [101] menciona erro de localização é menor que 1.7 metros com probabilidade de 0.77 numa experiência ao longo de um corredor com aproximadamente 9 metros de comprimento. Outro trabalho

denominado “RADAR” [102], [103] recorre a uma modelização empírica da propagação e atenuação da intensidade do sinal RF ao longo do sinal no espaço para conseguir a localização. Utilizam-se placas de rede sem fios genéricas, no caso *Lucent WaveLan*, que respeitam o protocolo IEEE 802.11b (2 Mbps). É referido um erro de cerca de 3 metros num mapa de 23 m por 42 m. Um dos objectivos declarados deste trabalho é o de criar componentes informáticos conscientes da sua localização dentro de um espaço estruturado onde exista uma rede sem fios. Estes sistemas estão, actualmente, a generalizar-se a elevado ritmo. Tomando como exemplo um *campus* universitário dotado de uma rede sem fios tal como preconizado pelo governo português [104], [105] seria então possível por exemplo imprimir para a impressora mais próxima do local em que o utilizador se encontre.

Uma interessante tecnologia para localização é o *Ultra Wide Band* (UWB) [106], [107]. O funcionamento baseia-se na transmissão de pequeníssimos impulsos RF, sem utilização de qualquer portadora. A duração dos impulsos é na ordem das dezenas de pico-segundos até alguns nano-segundos [107], [108]. É utilizado espectro espalhado na transmissão o que permite receptores pequenos e um certo grau de imunidade a percursos múltiplos, possibilitando assim a utilização desta tecnologia dentro de portas (*indoor*). O sistema comercial *Aether* [109] afirma permitir, dentro da legislação actual dos EUA, a localização com precisão de centímetro dentro de uma zona de um 1km. Este sistema mede o tempo de voo do emissor para o receptor e regresso através de integrações de pulsos de várias frequências. A deriva entre relógios é parte da informação transmitida para que a localização dependa do melhor relógio a uma distância conhecida [108].

3.3.3 Sistemas Globais de Navegação por Satélite

Os sistemas GNSS *Global Navigation Satellite System* são sistemas que envolvem tecnologias de satélites (com elevados custos). Para o sistema americano GPS, o custo aproximado de pôr o sistema em funcionamento foi de cerca de 12 mil milhões de dólares americanos. Existem, porém, receptores civis a custos interessantes para aplicações na área do posicionamento. Note-se que o posicionamento que os sistemas GNSS permitem têm imensas aplicações civis e militares. Esta questão é pertinente para várias aplicações com diversos graus de autonomia tal como a ajuda a deficientes e a localização de pessoas e bens.

A maior limitação dos GNSS é que o seu funcionamento está fortemente limitado a sistemas ao ar livre e sem barreiras naturais. É fortemente aconselhado ter linha de vista para vários satélites – tipicamente 4 satélites para uma localização básica.

Os GNSS baseiam-se na trilateralização das distâncias do receptor aos satélites [86]. As distâncias são medidas através do tempo de voo das mensagens emitidas

pelos satélites. O receptor estará então na intersecção viável das 3 calotes esféricas que representam a distância a cada satélite, que se presume conhecida. Uma medida adicional de tempo é necessária para eliminar a dessincronização entre os relógios dos satélites e do receptor. As distâncias da terra aos satélites são mantidas actualizadas com elevada precisão através de medidas em estações terrestres, depois difundidas pelo próprio satélite.

O sistema GPS *Global Positioning System* [86] [110] é o sistema mais conhecido. Utiliza 24 Satélites do Departamento de Defesa Norte-Americano e o início de operacionalidade ocorreu em 1993 mas a operacionalidade total data de 27 de Abril 1995. Este sistema fornece tempo actual e posicionamento 3D de um dado receptor. Para uma análise comparativa dos vários *chip sets* de recepção de GPS consultar [111]. Este sistema é de origem militar e assim existe a possibilidade de a onda emitida conter uma degradação propositada da qualidade da localização para fins civis chamado *Selected Availability* (SA). No comunicado da Casa Branca [112] é referida a data de 2006²³ para o fim da degradação propositada do sinal de GPS. Entretanto a pressão da instalação do sistema concorrente Europeu, o *Galileo*, fez com que a encriptação *Selected Availability* fosse desligada desde 1 de Maio de 2001 [113]. A precisão máxima do sistema (anteriormente classificada de “militar”) está agora acessível a aplicações civis e a qualquer receptor, enquanto os EUA assim o desejarem²⁴. O sistema civil GPS sofria anteriormente de um erro propositado de cerca de 30 m com média não nula ao longo do tempo, aplicado directamente na emissão do satélite (ver figura 3.4).

²³ 2006 é a data prevista de operacionalidade do sistema Europeu *Galileo*

²⁴ Os EUA afirmam que não têm intenção de voltar a activar globalmente e permanentemente o sistema SA mas admitem também a possibilidade de o activar regionalmente, conforme o seu próprio critério (que pode ser um critério militar)

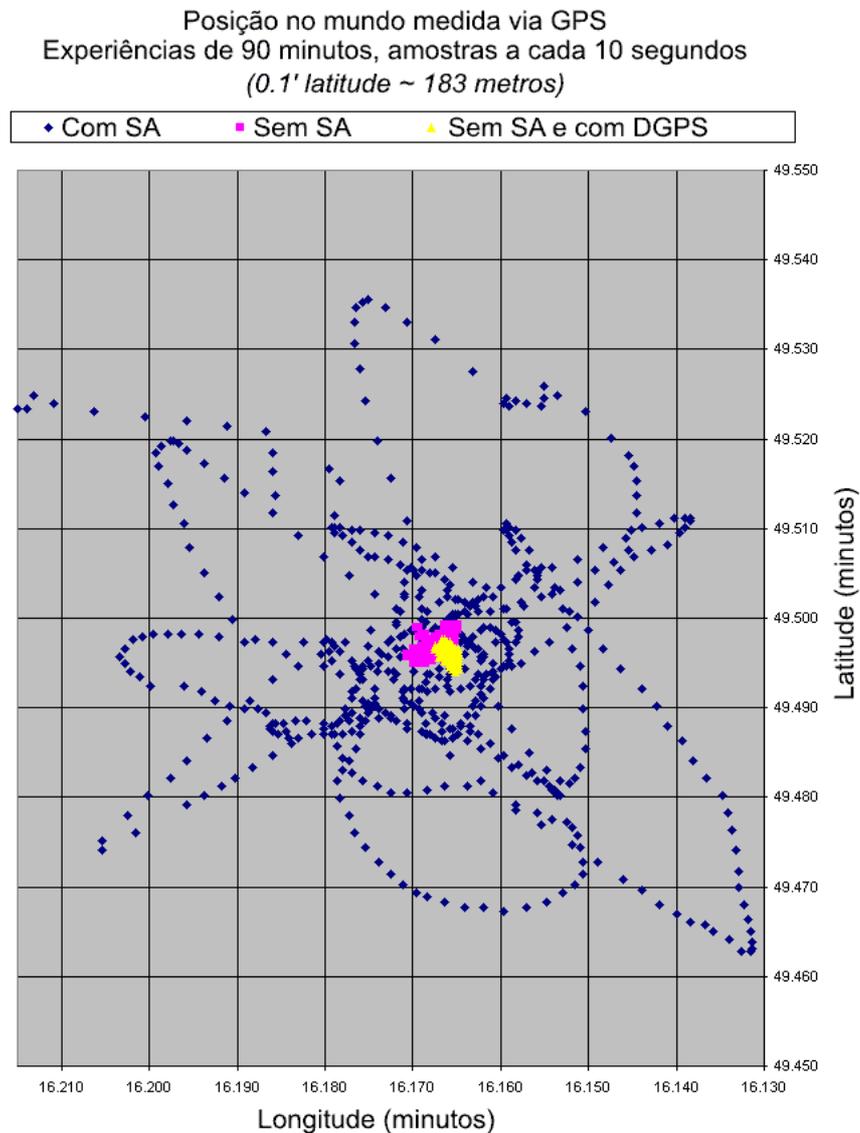


Figura 3.4 GPS com e sem Selected Availability - adaptado de [114]

O sistema GPS tem a seguinte precisão nominal (consultar também a tabela 3.1):

- Componente Horizontal:
 - 100 m , 95% do tempo;
 - 300 m, 99.99% do tempo;
- Componente Vertical:
 - 156 m, 95% do tempo;
 - 500 m, 99.99% do tempo;
- Componente temporal:
 - 340 ns, 95% do tempo.

Precisão comum da versão civil, para um receptor avançado mas que não

recolhe informação da fase do sinal recebido:

- Horizontal 50 m;
- Vertical 78 m;
- 3D 2.8m;
- 200ns.

Para contornar o sistema SA e alguns erros devidos à atmosfera que se podem considerar análogos para uma dada região surgiu o DGPS - GPS diferencial [115], [111]. Estações fixas geograficamente próximas do receptor transmitem correcções para localização fornecida pelo GPS (ver figura 3.5). Tal é conseguido através de um canal adicional de comunicações na expectativa de que os erros no receptor e na estação fixa sejam aproximadamente iguais. A tabela 3.1 lista a contribuição de cada fonte de erro. Afirma-se que a precisão obtida é:

- Horizontal: 1.3m;
- Vertical: 2.0m;
- 3D 2.8m;
- Tempo: 340ns.

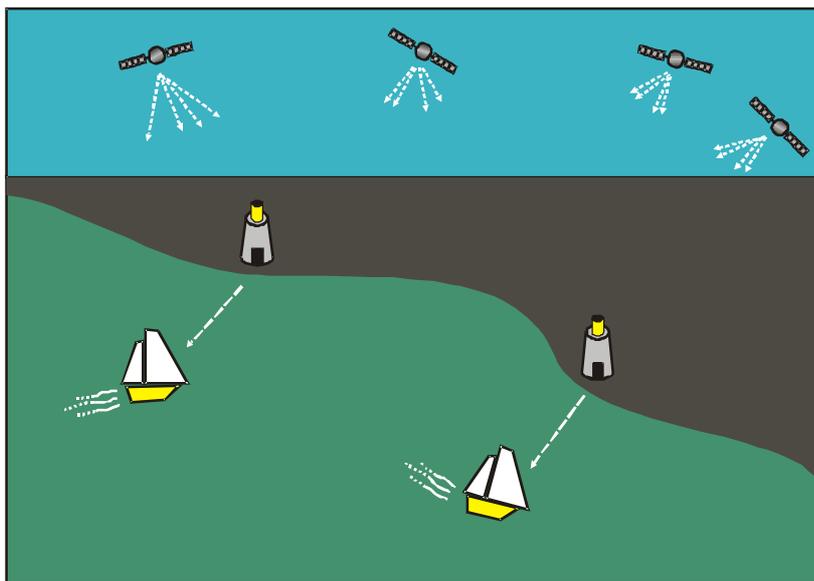


Figura 3.5 Ilustração do funcionamento do GPS Diferencial

Tabela 3.1 Sumário das várias fontes de erro presente no GPS, valor típico do erro em metros, por satélite (tabela compilada a partir de [86], [116] e [115])

<i>Fonte do Erro</i>	<i>GPS Standard</i>	<i>GPS Diferencial</i>
Relógio do Satélite	1.5 m	0
Erro na Posição Orbital	2.5 m	0
Atravessamento da Ionosfera	5.0 m	0.4 m
Atravessamento da Troposfera	0.5 m	0.2 m
Ruído no Receptor	0.3 m	0.3 m
Reflexões Múltiplas	0.6 m	0.6 m
Degradação propositada para fins civis (SA)	~ 30 m (Desactivada)	

O sistema GNSS de origem soviético chama-se *Glonass - Global'naya Navigatinnaya Sputnikovaya* mas aparece também referido como *Global Navigation Satellite System*. Este sistema baseia-se em 21 satélites enquanto que o GPS tem 24. Existe também a possibilidade de degradação propositada do sinal deixando a precisão em cerca de 100 metros. A precisão militar é de 10 a 20 metros:

- Precisão Horizontal: 57-70 metros (99.7% do tempo) [117];
- Precisão Vertical: 70 metros (99.7% do tempo) [117].

O início do lançamento de satélites foi em 1982 e finalização da colocação dos satélites em órbita ocorreu em finais de 1995. O sistema *Glonass* foi formalmente declarado operacional em 24 Setembro de 1993.

O produto comercial [118] em declara que integrando GPS+*Glonass* é possível obter em Tempo Real a precisão de centímetros. Tal é conseguido através da observação de um total de 6 ou mais satélites de qualquer das redes GPS e *Glonass* e também através da utilização de informação da fase das ondas recebidas. A mesma referência declara ainda que tal precisão é ainda possível em 100% do tempo mesmo existindo uma obstrução de 45° sobre a horizontal, à custa de um tempo que ronda 5 a 15 minutos. A referência [119], ligada ao mesmo produto comercial [118] refere que durante testes de voo a conjugação GPS+*Glonass* resultou em localizações com erro menor que 16 metros.

Há menções a que o sistema *Glonass* não tem demonstrado recentemente um elevado nível de operacionalidade [120], [121] pelo que a utilização de sistemas baseados apenas neste GNSS poderá enfrentar limitações.

O interesse da aeronáutica pelos sistemas GPS é compreensível para a melhor gestão do sempre crescente tráfego aéreo. Acontece que o GPS original não cumpre os requisitos da aeronáutica civil por exemplo a nível de segurança do sinal.

Aparecem assim diversas iniciativas tal como o “*Satellite-Based Augmentation System (MSAS)*” [122], [123], [124]. Este é um projecto de cooperação entre os EUA e o Japão para gerir cooperativamente redes de DGPS. Outro projecto similar dentro dos EUA é o WAAS - *Wide Area Augmentation System* [123]. Este último tem por objectivo o auxílio à manobra de aterragem em aeronáutica civil.

Outro GNSS será o sistema Europeu *Galileo* [121], [125], previsto para estar activo na totalidade em 2008 [126]. Este sistema é uma medida politico/estratégica para eliminar a dependência do GPS que é um sistema militar dos EUA. Está projectada uma constelação constituída por 27 satélites activos e 3 de reserva, em órbita média, que é a solução que oferece mais fiabilidade para o utilizador do sistema [127]. A precisão esperada será de cerca de 5 metros. O sistema *Galileo* terá diversas funcionalidades em acréscimo às do GPS (ver figura 3.6) tal como a capacidade de validação das outras redes GPS/Glanoss e fornece garantia de serviço a utilizadores civis. Outro sistema importante por exemplo para a aeronáutica é o sistema de monitorização da integridade do sinal: o utilizador será informado no caso de o sistema não poder ser confiado a 100% (GIPA *Galileo Integrity and Performance Assessment* [128] [121]. Actualmente existe já o sistema EGNOS - *European Geostationary Navigation Overlay System*. Este sistema de DGPS europeu será mais tarde será incorporado no *Galileo*. O objectivo é permitir a utilização generalizada para fins civis tal como por exemplo aplicações na área da aeronáutica. Decorre também um estudo de viabilidade acerca de um sistema de portagens pan-europeias em que o percurso é determinado através de GNSS e em particular será utilizado quando disponível, o sistema *Galileo* [126].

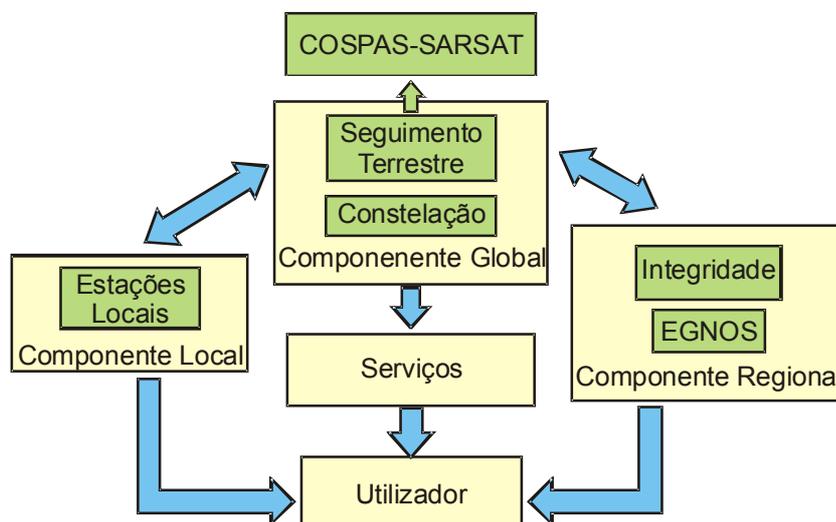


Figura 3.6 Diagrama de blocos do Galileo – adaptado de [121]

3.3.4 Soluções Baseados na Rede de Telemóveis

A rede de telemóveis celulares é actualmente amplamente disseminada e quase omnipresente. A utilização desta estrutura para a localização pode ser interessante para aplicações várias, não só no âmbito dos sistemas robóticos. Na verdade, a localização do chamador para o número de emergência [129], [106], [130] é um assunto de relevância para o público em geral e permite também a localização de equipamentos. Uma tal classe de sistemas poderão no futuro integrar informação proveniente das redes UMTS, GSM 900 e 1800 MHz, bem assim como informação proveniente de outras tecnologias diferentes tal como GPS.

O método mais simples de localização é conhecer a célula de origem da chamada. Esta informação já existe na rede. A precisão pode ser melhorada utilizando cálculos baseados em diferenças de tempos de chegada de pacotes emitidos pelo telefone a várias antenas (3 antenas). Exige no entanto que esteja a decorrer uma chamada. A implementação seria fácil por parte do operador da rede de telemóveis visto a informação já existir na rede. Basta utilizar o mecanismo de entrega de chamadas entre antenas (*hand-off*) e os pacotes vêm marcados com tempos diferentes (ver resumo na tabela 3.2).

Outra técnica diferente seria introduzir modificações às antenas tornando possível saber o ângulo de incidência do sinal que o telefone emite. Duas antenas com este sistema conseguem saber a localização do emissor. Este sistema seria adequado a zonas rurais com poucas antenas.

Seria ainda possível a auto localização do telefone celular que recebe 3 ou mais sinais de diferentes antenas e mede o tempo de chegada de cada um dos pacotes destinados a medir a potência do sinal.

É ainda possível implementar GPS diferencial (DGPS) baseado na informação recolhida pela localização obtida a partir da rede de telemóveis (chame-se a este sistema AGPS – GPS Assistido pela rede celular). Este sistema permitiria localizar com precisão o local a partir do qual é feita uma chamada para o número de emergência desde que o telefone inclua um receptor de GPS (ver resumo comparativo na tabela 3.2).

Tabela 3.2 Precisão de sistemas de localização com base na rede celular de telemóveis

Precisão	Técnica	Estimativa do Custo
200 m – 2.5 km	Célula de Origem	Menor de todos
30 m – 550 m	Tempo de chegada às antenas	Muito Baixo
20 m – 125 m	Ângulo de chegada às antenas	Elevado
50 m	Auto-Localização do receptor via diferenças de tempos de sinais emitidos por várias antenas	Baixo
5 m – 20 m	GPS assistido pela rede celular	Médio

3.3.5 Soluções Envolvendo Ondas Acústicas

Diversos autores trabalham com ondas acústicas para conseguir a localização. É um método muito utilizado pois o som desloca-se a uma velocidade que torna fácil medir o tempo entre a onda emitida e a chegada do seu eco. Infelizmente a velocidade exacta com que o som se propaga varia com as condições atmosféricas e estas ondas mecânicas são também susceptíveis de serem reflectidas ou totalmente absorvidas causando assim medidas falsas e ou ausência de medidas.

O trabalho [131] utiliza então a medida de distâncias com Ultra-Sons para descobrir correspondências entre o mapa conhecido e segmentos rectos nas medidas recebidas do sonar rotativo. As correspondências locais interessantes são guardadas e a validade global das correspondências é testada achando assim a localização e atitude do robot. Para fiabilidade acrescida é feito um teste que verifica que as medidas do sonar na actual pose do robot não ignoram nenhum objecto sólido conhecido. Outros trabalhos utilizam a triangulação e/ou trilateralização com dados provenientes de Ultra-Sons em configurações afins da referida [132], [133], [134], [135], [136] e [137].

O sistema “*The Bat System*” - Sistema dos Morcegos [138] utiliza Rádio Frequência (RF) para comunicar num sentido e Ultra-Sons para comunicar no outro sentido. A medida do tempo de voo é feita por subtracção de tempos fixos medidos. É possível estender o processo até determinar posição e rotação 3D à custa de maior número de medidas.

Outro trabalho utiliza audição *stereo*: utilizando 2 microfones omni-direccionais e utilizando a diferença entre tempo de chegadas de ondas sonoras é possível saber

a direcção proveniente do som e assim conseguir a localização [139].

3.3.6 Soluções Envolvendo Visão

Os sensores ópticos são muito utilizados na robótica. São conhecido robots que seguem linhas utilizando sensores ópticos elementares tal como o foto-díodo. Este tipo de sensores podem até conseguir dar informação da velocidade angular a que o robot se desvia do seu percurso, numa configuração muito simples apresentada em [140].

As técnicas de localização baseadas em visão são objecto de intensa investigação. A generalização da utilização de câmaras e melhoria da sua qualidade a baixo custo tornam este sensor interessante para múltiplas aplicações. De facto uma imagem é rica em informação e permite retirar diversas informações. Este processamento beneficia dos recentes avanços em termos dos poder de cálculo disponível. Acresce ainda que a recolha de imagens através de câmaras COTS fornece já qualidade na imagem a uma resolução interessante bem assim como um regime dinâmico interessante: as *webcams* actuais de baixo custo conseguem resoluções na ordem dos 640x480 a 30 imagens a cores por segundo. A utilização de sistemas de orientação (*Pan & Tilt*) de câmaras oferecem ainda uma direccionalidade que pode ser interessante. Adicionalmente pode ser controlada a resolução da aquisição quer através de subamostragem quer através da utilização de *zoom* (óptico) o que permite obter informação com nível variado de detalhe. Dada uma colocação da câmara pode ser interessante adaptar a resolução efectiva à cena a visualizar, adequando assim a taxa de amostragem à densidade da informação na zona em causa da imagem [141]. Todas estas técnicas resultam no entanto numa sequência de dados não tridimensionais. O sistema de processamento da imagem será aprofundado na secção 4.2, pág. 70.

É também possível utilizar um sistema óptico que inclua espelhos para modificar o mapeamento dos pontos da cena para a imagem. Esta técnica é utilizada frequentemente pois é mais conveniente em termos mecânicos um espelho construído de raiz para a aplicação do que o sistema equivalente em componentes ópticos transparentes.

Sendo que a câmara é um sensor passivo, utiliza a iluminação já existente no meio para recolher dados. Por vezes e para a obtenção de resultados óptimos é necessário iluminação cuidada. Tal é frequente em sistemas de visão na indústria. Existem diversas técnicas de iluminação de imagem [142].

- Iluminação Frontal:
 - Difusa (uniforme);
 - Direccional, ex: rasante à cena para aumentar contraste dos contornos e dos baixo-relevos na peça;
 - Polarizada – para evitar reflexos;

- Circular (em torno da câmara ou da cena);
- Iluminação estruturada – a estrutura da iluminação dá informação tridimensional – ver exemplo na Fig 3.8 onde é utilizada uma linha produzida por um laser;
- Iluminação na retaguarda do objecto (*back light*):
 - Difusa – realçar contornos e furos;
 - Direcional realça alguns contornos;
 - Polarizada – permite contornar algumas transparências ou reflexos permitindo por exemplo realçar defeitos internos a uma peça transparente.

Diversas técnicas são utilizadas para introduzir informação tridimensional na imagem obtida.

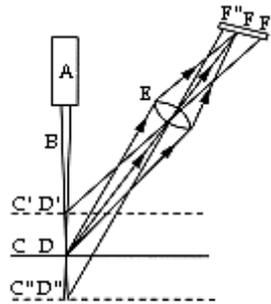


Figura 3.7 Princípio de Funcionamento de um PSD – adaptado de [143]

A iluminação estruturada introduz na imagem informação tridimensional codificada por exemplo nas intensidades luminosas [144]. Os dispositivos PSD são exemplo simples disso mesmo (ver figura 3.7). Esta técnica é também muito utilizada em sistemas industriais para medidas ao longo do corpo de um objecto, tal como ilustrado pela Fig 3.8.

Considerando em concreto o caso da visão embarcada num robot, a câmara obtém normalmente uma visão limitada da cena em torno do robot. Para obter mais dados, utilizam-se sistemas de orientação da câmara e ou lentes e espelhos para abranger uma maior porção de espaço na mesma imagem.

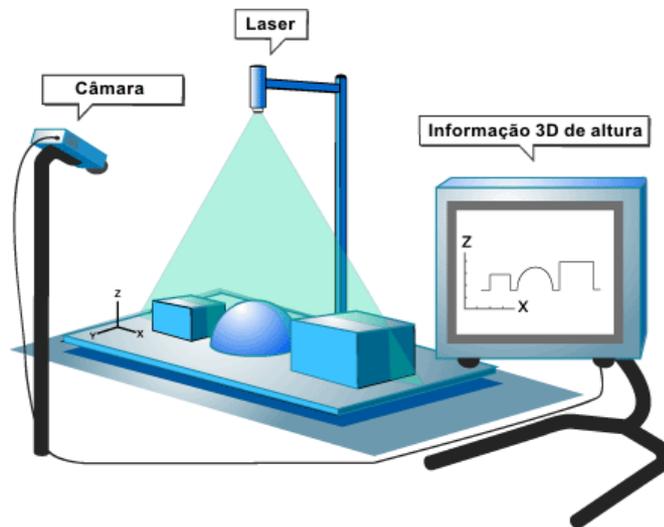


Figura 3.8 Iluminação estruturada (adaptado de [145])

Uma técnica muito utilizada é a de apontar a câmara para um espelho cónico (ou outra forma) conseguindo assim observar 360° em torno do robot [58]. A utilização deste tipo de espelhos ditos omni-direccionais [146] [147] [148] é muito útil porque todas as direcções são amostradas simultaneamente. As imagens obtidas são assim panorâmicas da cena em torno do robot. Para obter o mesmo tipo de dados num sistema com câmara orientável seria necessário rodar a câmara e a amostragem não seria conseguida no mesmo instante introduzindo assim complexidade acrescida no processamento da imagem e algum erro inerente às dinâmicas do robot e do ambiente. Os espelhos omni-direccionais distorcem assim o mapeamento espaço-imagem: a forma exacta do espelho permite ajustar esse mapeamento. É possível construir um espelho tal que as rectas no espaço sejam mapeadas em rectas na imagem o que facilita a reconstrução da cena envolvente ao robot - estes espelhos são denominados de isométricos (ver fig 3.9) [147], [149]. A análise da imagem poderá nessa situação encontrar mais facilmente os pontos de interesse .

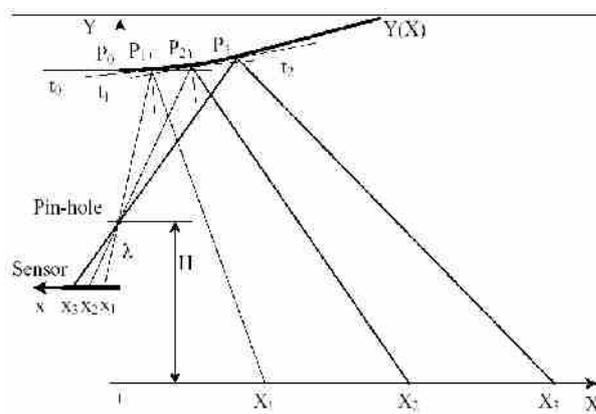


Figura 3.9 Esquema do espelho isométrico - de [149] -

$$x_{i+1} - x_i = k(X_{i+1} - X_i) \forall i$$

Uma opção interessante é utilizar a forma do espelho para adequar o mapeamento do espaço-imagem aos requisitos do problema em causa. Este tipo de espelho será então constituído por várias partes que permitem conciliar requisitos tal como precisão ao perto e mesmo assim conseguir ver longe (com menor precisão) [147], [150], [151], [152].

Uma técnica que os seres humanos utilizam para aferir a 3ª dimensão é a visão *stereo*. Utilizam-se não um mas dois sensores de imagem (câmaras ou os olhos humanos) e é feita uma correspondência entre os objectos visualizados nas duas imagens; sendo as duas imagens adquiridas em posições conhecidas diferentes no espaço, é possível determinar a distância ao objecto visto mesmo na presença de uma cena complexa [153]. A referência [154] tenta contornar a visão parcial da cena utilizando lentes de grande abertura (do tipo “olho-de-peixe”) para depois fazer processamento das imagens *stereo* obtidas.

Dentro desta técnica é também possível utilizar uma diversidade de configurações de espelhos para conseguir as duas imagens utilizando espelhos em várias configurações [151], [155], [156], algumas configurações são apresentadas na figura 3.10.

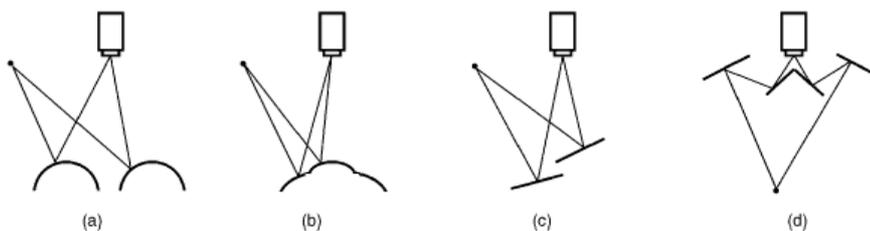


Figura 3.10 Utilização de espelhos para obter duas imagens – adaptado de [151]

Diversos autores trabalham a interessante técnica de utilizar as lâmpadas do tecto como marcadores naturais para localização e navegação [157] [158]. Este sistema tem a vantagem de já estar instalado e de ser facilmente reconhecível com uma câmara apontada para o tecto. A desvantagem é que há marcadores repetidos pelo que é necessário utilizar informação acerca da localização anterior para calcular a posição actual. Tal como referido anteriormente, neste tipo de marcadores naturais é necessário haver um mapa com a sua localização real. Um erro na medida da posição de um destes marcadores não é cumulativo. Este sistema acaba por escalar relativamente bem visto que é frequente haver muitas fontes luminosas artificiais nos ambientes dentro de portas onde circulam normalmente os robots. A dificuldade passa então a ser lidar com as fontes de luz em falha e com a iluminação natural. Como exemplo de resultados obtidos com esta técnica, [159] segue lâmpadas no tecto com forma rectangular separadas de uma distância de cerca de 2.5 metros conseguindo uma localização caracterizada por:

- Erro segundo o eixo dos xx de $\sim N(2.4\text{mm}, 8.8 \text{ mm})$, erro máximo de 15.2mm;

- Erro nos $yy \sim N(3.2\text{mm}, 32\text{mm})$, erro máximo de 32.3mm;
- Erro na orientação de $\sim N(0.04 \text{ graus}, 0.5 \text{ graus})$, erro máximo de 4.5 graus.

No trabalho [160] são utilizadas marcações ópticas para gerar um padrão que é único e que, sendo reconhecido, permite encontrar uma localização global (ver figura 3.11). O padrão dos dados é ternário e a imagem recolhida a Preto e Branco. Em torno da marca com nível de cinzento intermédio estão os dados que fornecem o tal padrão único. São também lançados elementos para uma outra forma de padrão que utiliza apenas Preto e Branco puros.



Figura 3.11 Robot tentando localização absoluta utilizando padrão ternário [160]

Existe ainda a possibilidade de tentar reconhecer a própria forma do corredor em que o robot se encontra para realizar navegação [161], [162]. Para tal são levantados diversos tipos de elementos geométricos e depois mapeados com o mapa conhecido do mundo. A tabela 3.3 identifica várias técnicas possíveis para esta abordagem e realça algumas dificuldades inerentes a cada processo.

O trabalho [189] propõe uma estratégia de localização a dois níveis em simultâneo: ajustes locais e estratégia global. O trabalho utiliza as linhas de um campo de futebol robótico da liga F2000 da Federação RoboCup. O método global estuda diversas localizações possíveis que se ajustem aos elementos (linhas) da imagem. O método global é utilizado quando o método local falha. O método local segue os elementos encontrados na imagem em Tempo Real. Nas conclusões deste trabalho é referido, no entanto, que curvas apertadas são mal suportadas pelos algoritmos implementados.

Tabela 3.3 Resumo comparativo e pontos fracos da utilização de diversos elementos a seguir [161]

<i>Elemento Seguido</i>	<i>Entrada</i>	<i>Objectivo</i>	<i>Dificuldades</i>
Detecção de bordas (<i>edge detection</i>)	Intensidades dos pixels	Encontra elementos de grande contraste	Não discrimina bordas
Bordas ajustadas à escala	Taxa de amostragem, pixels amostrados	Encontra bordas não óbvias	Perda de precisão devido à amostragem
Correlação de intensidades dos elementos de primeiro plano	Intensidades dos pixels em primeiro plano: actual e anterior	Evidencia bordas com primeiros planos similares	Dificuldades com iluminação variável, a oclusão parcial e “distracção”
Tom de cor do elemento de primeiro plano	Cor dos pixels: actual e anterior	Evidencia bordas com a mesma cor que o elemento de 1º plano	Podem existir dificuldades com oclusão e “distracção”
Pesquisa de segmentos	Pixels junto aos finais dos segmentos	Identifica rectas/segmentos distorcidas	Utiliza a detecção de bordas
Movimento	Estado anterior e modelo dinâmico	Permite sobreviver à oclusão	Espera que os elementos se movam de acordo com o modelo dinâmico
Modelo Geométrico	Estados de outros elementos	Integrar com outros elementos seguidos	Exige confirmação adicional na imagem

A referência [163] utiliza um sistema em que a localização é baseada em marcadores especiais de uma cor. Os marcadores são seguidos com um sistema *pan-tilt-zoom* que permite observar e medir as dimensões do marcador com precisão. Através da medida na imagem do marcador colorido, é possível aferir distância e ângulo ao marcador. Refere-se uma localização com um erro de cerca de 7 cm numa experiência de 3m por 2.4m.

O trabalho [164] utiliza uma diversidade de sensores para fazer a localização. São utilizados um giroscópio, 2 câmaras e informação proveniente de hodometria. Toda a informação é fundida num Filtro de Kalman Extendido. É referida um erro de localização em (x,y) com uma distribuição normal com média de 0.35 m e 0.3 m de desvio padrão. Para o erro angular é referida uma distribuição normal com média de 0.2°, desvio padrão de 3.5°.

3.4 Localização e Mapeamento Simultâneo

Os dados que robot recolhe relativamente ao meio ambiente podem ser utilizados tanto para a localização como para a actualização do mapa do

ambiente exterior. Esta técnica designa-se **localização e mapeamento simultâneo** (SLAM - *Simultaneous Localization And Mapping*). Estes métodos são tipicamente utilizados para ambientes algo variáveis e as suas aplicações são muitas vezes ao ar livre.

Estes métodos podem ser agrupados em 2 tipos [165] [166]:

- Métodos baseados em medidas directas dos sensores:
 - cada medida tenta produzir uma localização tendo em conta dados os recolhidos anteriormente;
 - se a localização é conseguida, então é acrescentada a nova informação ao mapa;

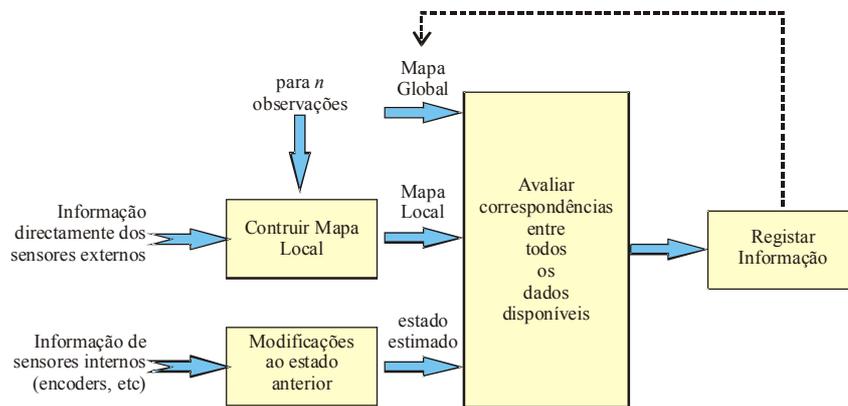


Figura 3.12 Método SLAM utilizando dados directos dos sensores

- Métodos baseados em marcas características (de alto nível) reveladas por dados dos sensores:
 - Dados dos sensores são processados para obter marcas características de alto nível;
 - Tenta obter correspondências dessas marcas em leituras anteriores dos sensores;
 - Se a localização é conseguida, então é acrescentada a nova informação ao mapa.

Uma das questões centrais neste tipo de métodos é o percurso que o robot deve fazer para conhecer o mapa: onde é que o robot não esteve? Algumas soluções são:

- movimentos aleatórios;
- varrer fronteira mais próxima;
- utilizar o histórico dos movimentos medidos pelos sensores próprios para evitar zonas visitadas recentemente.

Um exemplo simples do conceito SLAM é a utilização de modelos probabilísticos

para os mapas e para robot. É então possível utilizar probabilidades condicionadas para ajustar o mapa e a posição do robot com o recurso às novas informação recolhidas pelo sistema [167], [168], [169].

3.5 Conclusões

Neste capítulo foram listadas abordagens genéricas à localização e soluções concretas de localização trabalhadas por outros autores. Esta área é motivo de intensa pesquisa científico-tecnológica. As soluções dependem fortemente da aplicação específica em causa, em particular do o nível de estrutura do ambiente, dentro ou fora de portas, das características do mapa disponível e das dimensões em causa para o movimento.

Foi também mencionada a técnica de localização e mapeamento simultâneo que tenta, ao longo do tempo, ir melhorando a representação do ambiente enquanto o robot se desloca.

Capítulo 4

Sistemas de Visão

4.1 Cor e Imagem

A aquisição da cor por parte do ser humano é baseada em 3 sensores, os cones retiniais que adquirem cada uma das 3 componentes básicas de cor (ver fig. 4.1).

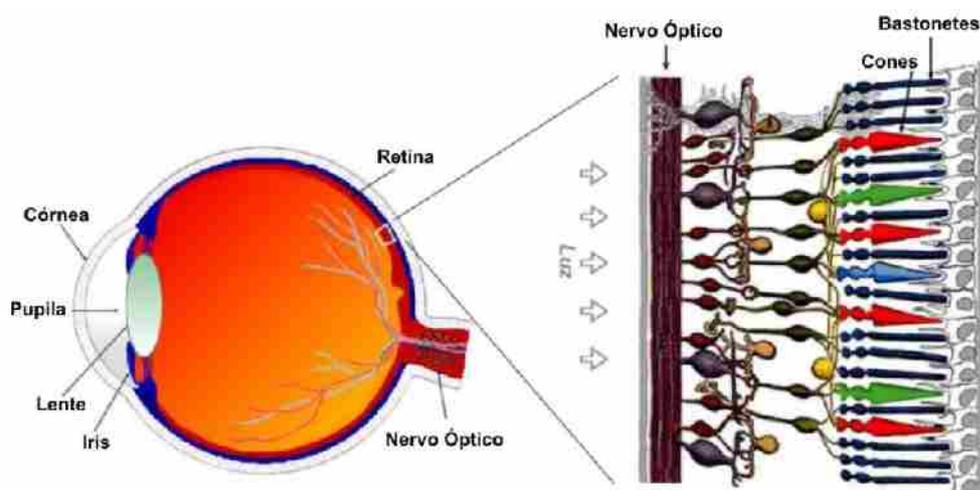


Figura 4.1 Sensores RGB nos Humanos (adaptado de [170])

Este sistema de cor é denominado de RGB, sigla anglo-saxónica respectivamente

para as cores Vermelho, Verde e Azul [171], [172] e [173]. Cada um destes sensores tem uma sensibilidade diferente à luz ao longo do espectro da luz, mostrado na figura 4.2.

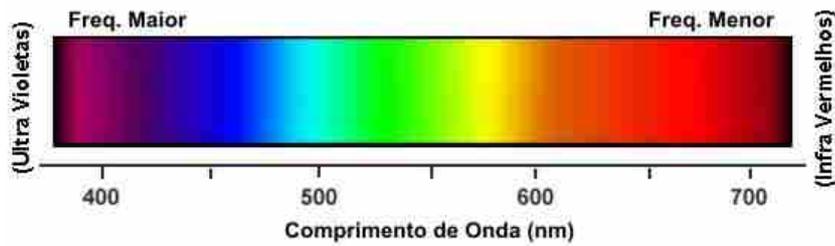


Figura 4.2 Espectro de cor (adaptado de [173])

É possível então caracterizar a resposta do olho humano a cada frequência de luz, tal como mostrado na figura 4.3.

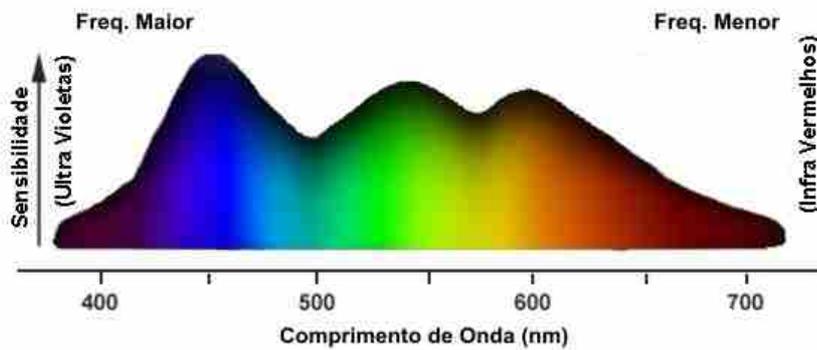


Figura 4.3 Sensibilidade a cada cor (adaptado de [173])

Cada cor tem uma frequência que estimula de forma diferente os sensores de cor da visão humana. Todas as cores que podemos ver podem assim ser obtidas à custa de 3 cores primárias. O modelo RGB é aditivo e assim as cores secundárias são conseguidas através das combinações somadas das cores primárias tal como mostrado na figura 4.4.

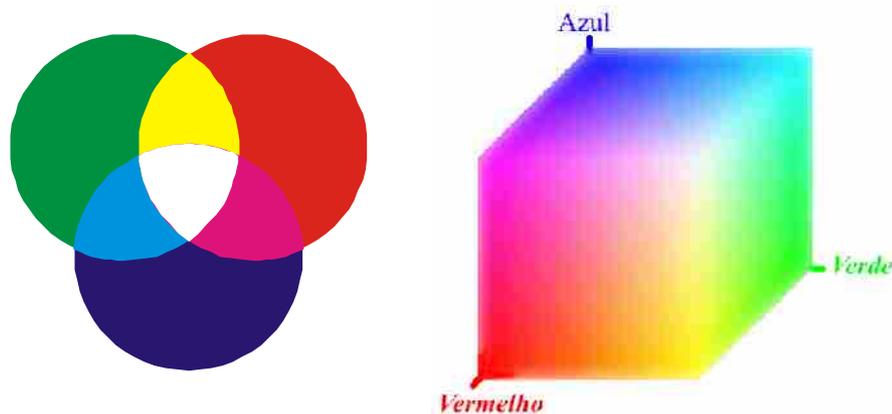


Figura 4.4 À esquerda, cores primárias e secundárias; à direita pode-se ver o cubo RGB

Uma importante questão relativa à visão humana é a interpretação inerente as imagens que tende a realçar as diferenças em torno do que se entende como sendo o fundo da imagem. Tal é devido à grande adaptabilidade do olho humano face às condições de iluminação e à variação desta iluminação mesmo dentro da mesma imagem. Observe-se os quadrados A e B da figura 4.5 que são rigorosamente do mesmo tom de cinzento mas aparentam ser diferentes²⁵. O observador humano desta figura não consegue deixar de a interpretar e assim aparece a ilusão de que A e B têm cores diferentes.

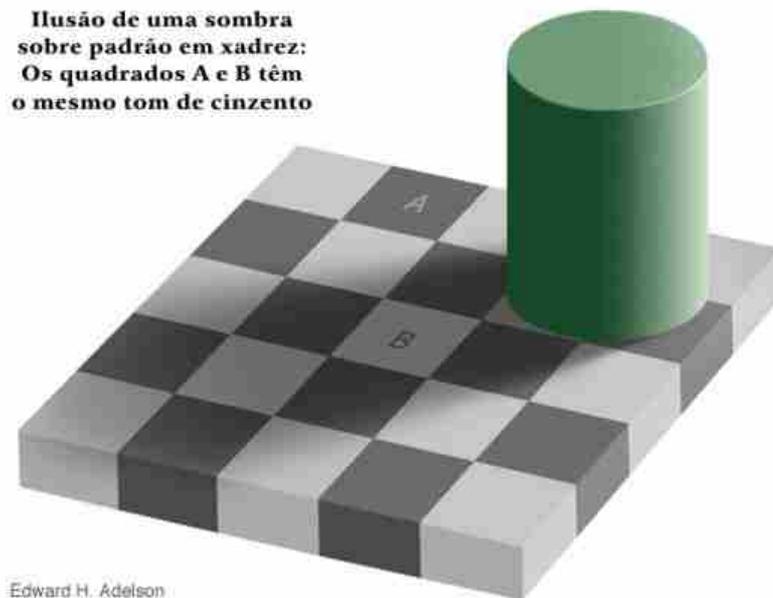


Figura 4.5 Ilusão óptica num tabuleiro em xadrez

Tomando então uma definição matemática, qualquer cor pode ser definida pelo conjunto das suas componentes básicas RGB. Uma imagem é então definida pela função:

$$\begin{aligned}
 rgb: \quad I &\rightarrow \mathbb{R}^3 \\
 p_{uv} &\rightarrow (r, g, b) \\
 r, g, b &\in [0, 1], u \in [0, u_{max} - 1], v \in [0, v_{max} - 1]
 \end{aligned} \tag{4.1}$$

Esta função associa a cada pixel da imagem de dimensões u_{max} por v_{max} uma cor com componentes (r, g, b) para o vermelho, verde e azul com intensidades que podem ser normalizadas no intervalo unitário. Em sistemas digitais estas intensidade (r, g, b) são discretizadas em n_r, n_g, n_b bits respectivamente. A cor assume assim um número não infinito de combinações. Utilizando 1 byte para cada componente de cor ($n_r = n_g = n_b = 8$, total de 24 bits) o efeito da discretização de cor já é visualmente indetectável. Uma profundidade de cor muito utilizada por ser compacta e mesmo assim a perda de informação visual não ser importante é a utilização de 5 bits para cada componente de cor (total de 15

²⁵ Para confirmar, tapar toda a figura excepto os 2 quadrado em causa

bits). Não ultrapassando 2 bytes para cada pixel é também muito utilizada a “profundidade de cor” (nível de discretização) onde $n_r=5$, $n_g=6$, $n_b=5$, num total de 16 bits. A componente relativa ao verde é amostrada com mais bits visto que é na zona das cores verde/amarelo que a nossa visão assume maior sensibilidade quando tomada de forma global, tal como mostrado na fig 4.6.

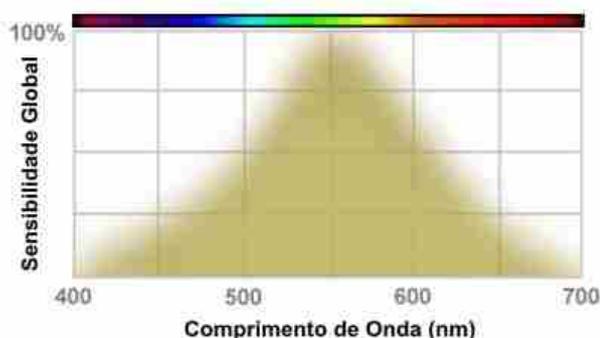


Figura 4.6 Sensibilidade global da visão humana (adaptado de [173])

Para a difusão sem fios de televisão analógica na Europa ocidental utiliza-se a normal PAL. Este sistema utiliza um sistema de cor que mantém a compatibilidade com o sistema antigo de televisão a preto e branco. Este sistema denomina-se de YIQ. A componente y é a luminância, sinal a preto e branco que será transmitido de tal forma que as televisões a preto e branco continuem a funcionar. A componente y avalia a relação claro – escuro da cor em causa, isto é, corresponde à noção psicológica de claridade. Toda a informação relativa à cor é transmitida em duas componentes, I em fase e Q em quadratura. A relação formal entre as componentes (rgb) e (yiq) é a seguinte:

$$\begin{bmatrix} y \\ i \\ q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} \quad (4.2)$$

Note-se que a relação (4.2) quantifica alguma da informação contida na fig 4.6. Pode-se ver que no cálculo da luminância, a cor verde tem o maior peso e a cor azul tem o menor peso. Esta formalização reflecte a noção intuitiva que o azul contribui menos para a claridade de uma dada cor que as outras cores primárias.

4.2 Modelo para a Câmara

Os conceitos apresentados de seguida são complementares às considerações anteriormente referidas relativamente à iluminação de uma dada cena (consultar secção 3.3.6, pág. 59).

4.2.1 Parâmetros intrínsecos

Considere-se o problema da visão num sistema robótico onde a câmara pode ser transportada pelo robot ou pode observar de forma estacionária uma cena. Em qualquer caso há uma projecção de uma realidade tridimensional numa imagem bidimensional (figura 4.7). Observando o mesmo objecto com a mesma câmara é então possível obter imagens muito diferentes (figura 4.8).

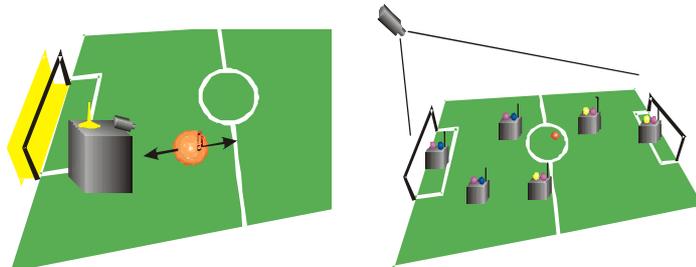


Figura 4.7 Casos frequentes em visão para robótica

Considere-se o modelo “pin-hole” para a câmara, ilustrado nas figuras 4.9 e 4.10. Neste modelo existe um plano virtual onde se forma a imagem que se situa entre o buraco do alfinete (*pin-hole*) e a cena visualizada.

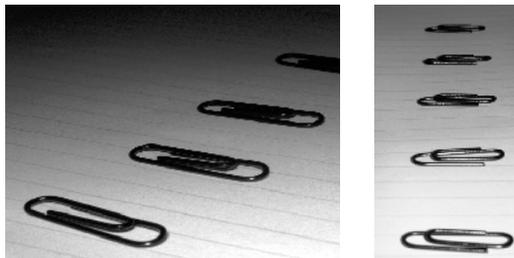


Figura 4.8 Efeitos de uma projecção

Na realidade utilizam-se lentes para focar a imagem e modificar a amplitude da cena registada pela câmara (fig. 4.11).

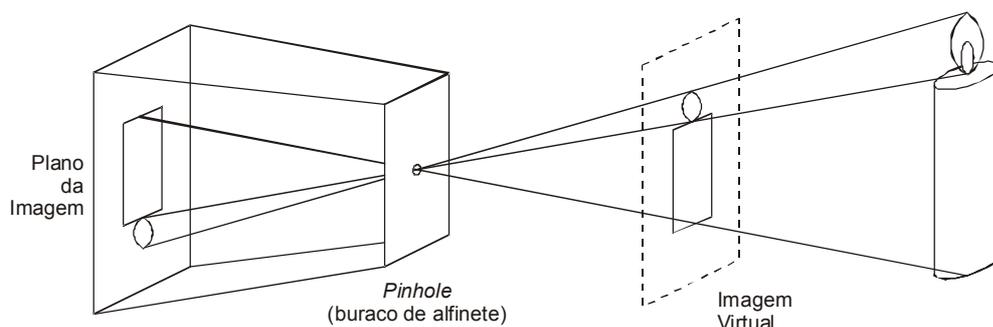


Figura 4.9 Esquema visual de uma câmara mostrando o plano virtual da imagem

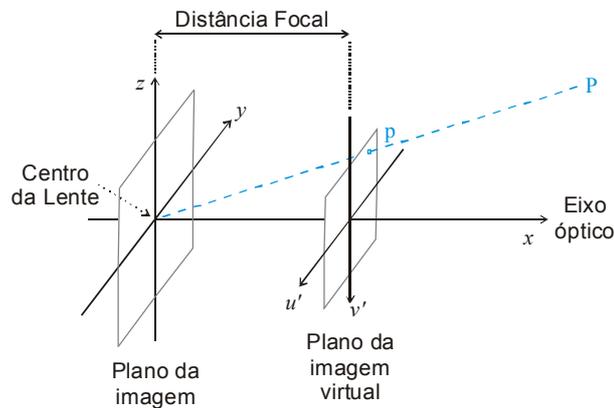


Figura 4.10 Sistema de eixos no modelo da câmara

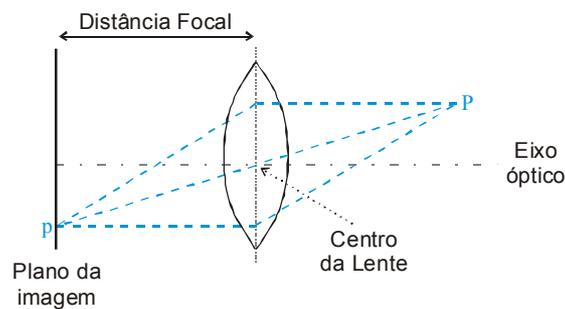


Figura 4.11 Efeito de uma lente

A relação entre os pontos $P(X,Y,Z)$ e os pontos da imagem (u',v') é:

$$\begin{cases} u' = \frac{-y}{x} f \\ v' = \frac{-z}{x} f \end{cases} \quad (4.3)$$

Onde f é a distância focal da lente utilizada.

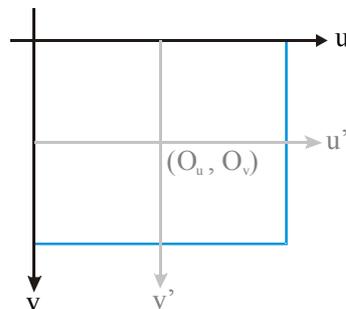


Figura 4.12 Coordenadas da imagem

Considere-se agora um novo sistema de coordenadas centradas para a imagem tal como ilustrado na fig. 4.12 e definido por:

$$\begin{cases} u' = (u - O_u) \cdot S_u \\ v' = (v - O_v) \cdot S_v \end{cases} \quad (4.4)$$

Onde S_u e S_v são as distâncias horizontal e vertical respectivamente entre pixels adjacentes. Se o pixel tiver forma quadrada, $S_u = S_v$.

As lentes beneficiam a formação da imagem mas provocam um efeito chamado efeito de barril que pode ser modelizado por:

$$\begin{cases} u' = (u_d - O_x) \cdot S_u \cdot (1 + k_b \cdot rd^2) \\ v' = (v_d - O_y) \cdot S_v \cdot (1 + k_b \cdot rd^2) \end{cases} \quad (4.5)$$

Onde u_d e v_d são as coordenadas da imagem distorcida e onde a distância ao centro da imagem é $rd = \sqrt{x_d^2 + y_d^2}$. A constante k_b depende do nível de distorção da lente. A figura 4.13 apresenta o aspecto de uma imagem distorcida aparentada com um barril e de uma imagem distorcida aparentada com uma “almofada” (*pin-cushion*). A relação matemática 4.5 corrige uma distorção em barril com um k_b positivo – na figura 4.13 os pontos correctos foram distorcidos.

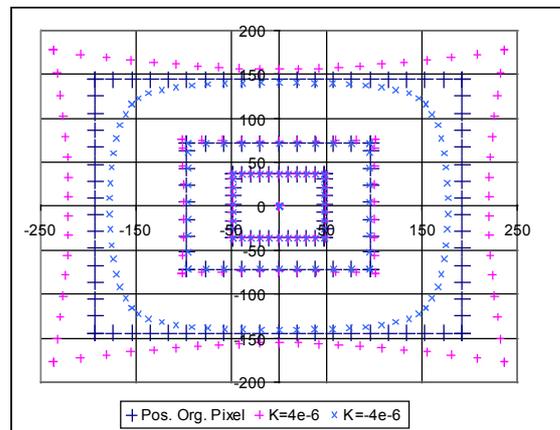


Figura 4.13 Ilustração da imagem com distorções do tipo barril e “pin-cushion” num exemplo apresentando a resolução de 384 por 288 pixels

A modelização da câmara deve ainda referir-se à resolução. Se as distâncias entre pixels adjacentes da câmara forem conhecidas com o valor (P_x, P_y) , então as distâncias dos pixels na imagem são dados por:

$$\begin{cases} S_x = ax P_x \\ S_y = P_y \end{cases} \quad (4.6)$$

Onde ax é um factor de escala para compensação da eventualidade de as frequências de amostragem horizontal e vertical da câmara não tomarem valores iguais.

Estes parâmetros são relativos à câmara e ao sistema óptico e depois de determinados mantêm-se inalterados para esse mesmo conjunto câmara e óptica.

Os parâmetros intrínsecos da câmara são:

- O_u e O_v - centro da imagem segundo eixo dos uu e dos vv respectivamente;
- d - Distância do plano virtual da imagem ao ponto de foco;
- k_b - Parâmetro da distorção em barril (do sistema óptico da câmara);
- M_u e M_v - Número de pixels segundo o eixo dos uu e dos vv , respectivamente.

4.2.2 Parâmetros Extrínsecos

A imagem da câmara depende naturalmente da sua colocação no mundo - estes são os seus parâmetros extrínsecos.

Estes parâmetros extrínsecos podem ser listados por inspeção e dizem respeito à colocação tridimensional de um objecto:

- β Rotação em torno do eixo dos xx ;
- α Rotação em torno do eixo dos yy ;
- θ Rotação em torno do eixo dos zz ;
- (X, Y, Z) Posição da câmara no mundo.

Para facilidade de consulta e como exemplo considere-se o caso da figura 4.14 que será desenvolvido em capítulo posterior.

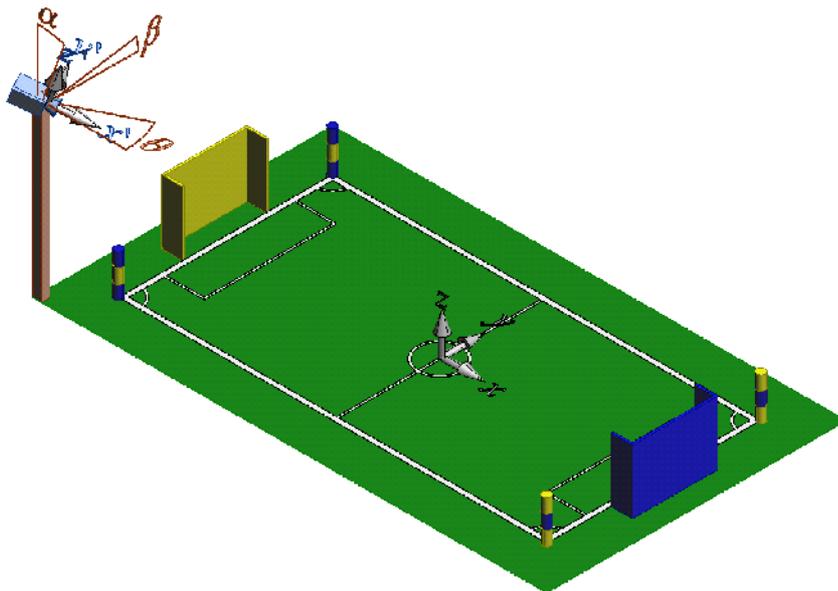


Figura 4.14 Sistemas de eixos utilizados, representação em perspectiva isométrica

Observe-se os eixos do mundo na figura 4.14 e o detalhe do posicionamento e orientação da câmara (e respectivo sistema de eixos), detalhados na figura 4.15.

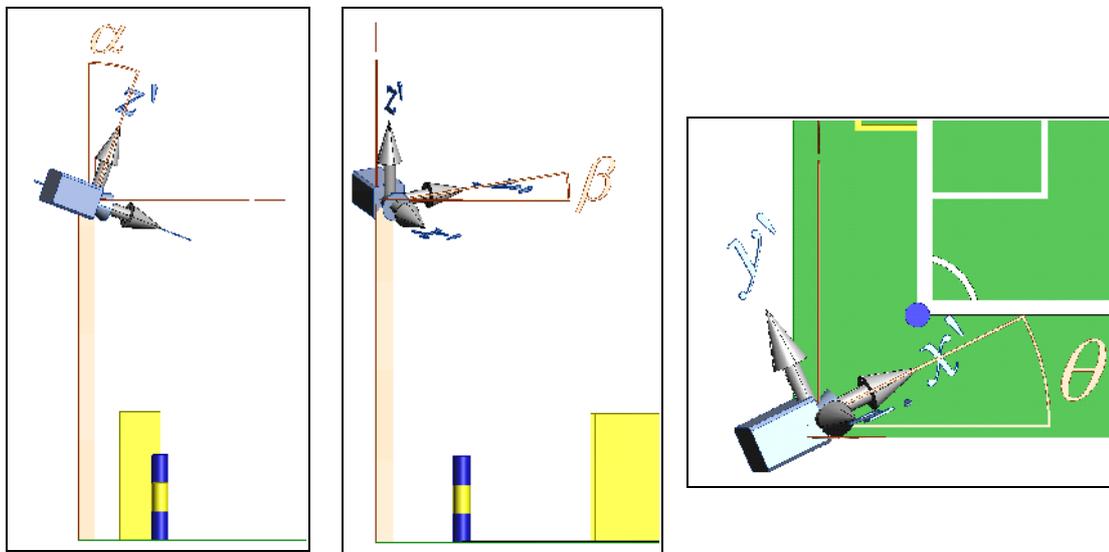


Figura 4.15 Sistemas de eixos em detalhe

4.2.3 Relação imagem-mundo (UVZ2XY)

Para a calibração da câmara e para obter medidas da câmara numa dada posição é necessário estabelecer a relação matemática entre pontos 3D do mundo e pontos da imagem 2D.

Pretende-se uma relação que produza a posição do mundo (X, Y) de um dado pixel da imagem (u, v) . Para tal é necessário uma sequência de transformações listadas de seguida. Utilizam-se como dados os parâmetros da câmara e a altura do ponto no mundo (Z) . Para facilidade designe-se esta técnica como “*algoritmo UVZ2XY*”.

Considere-se então o ponto da imagem (uc, vc) que se pretende centrar no novo sistema de coordenadas da imagem (u', v') é conseguida através de:

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} uc \\ vc \end{bmatrix} - \begin{bmatrix} O_u \\ O_v \end{bmatrix} \quad (4.7)$$

Em que O_u e O_v são os centros da imagem segundo o eixo uu e vv respectivamente.

De seguida roda-se toda a imagem β graus em torno do seu centro (virá a ser o eixo x'):

$$\begin{bmatrix} u_{rot} \\ v_{rot} \end{bmatrix} = \begin{bmatrix} \cos(\beta) & \sin(\beta) \\ -\sin(\beta) & \cos(\beta) \end{bmatrix} \begin{bmatrix} u' \\ v' \end{bmatrix} \quad (4.8)$$

Esta formulação em concreto é válida para o caso dos robots 5dpo-2000 pois é necessário rodar a imagem 90° para compensar a colocação vertical da câmara nos robots.

Aplicando a compensação da distorção em barril, os pontos no plano virtual da imagem são dados por:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} d \\ -v_{rot} \left(1 + k_b \sqrt{u_{rot}^2 + v_{rot}^2} \right) \\ u_{rot} \left(1 + k_b \sqrt{u_{rot}^2 + v_{rot}^2} \right) \end{bmatrix} \quad (4.9)$$

Rodando o plano da imagem virtual α graus em torno do eixo dos y' obtemos os pontos:

$$\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (4.10)$$

É ainda necessário interceptar a recta que passa pela câmara e pelo ponto do plano virtual da imagem com o plano $z=z_w$ do mundo real

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} ((z_w - Z)x_p / z_p) \\ ((z_w - Z)y_p / z_p) \end{bmatrix} \quad (4.11)$$

Fazendo a translação e rotação do ponto anterior para as coordenadas do mundo

$$\begin{bmatrix} x_w \\ y_w \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_c \\ y_c \end{bmatrix} + \begin{bmatrix} X \\ Y \end{bmatrix} \quad (4.12)$$

assim encontramos as coordenadas (x_w, y_w) correspondentes ao ponto (u_c, v_c) da imagem.

4.2.4 Calibração de Parâmetros

Conhecendo o algoritmo anterior, a calibração das câmaras pode ser feita através de métodos numéricos.

É necessário criar uma lista de correspondências entre pontos conhecidos do mundo mapeados em pontos conhecidos da imagem. Essa lista será:

$$L_i : (u_i, v_i) \rightarrow (x_i, y_i, z_i) \quad (4.13)$$

onde (u_i, v_i) são as coordenadas do ponto da imagem e (x_i, y_i, z_i) são as correspondentes coordenadas do mundo. Para facilidade de realização prática da calibração e para facilitar a convergência numérica é usual que z permaneça constante e nulo.

Implementando o algoritmo referido no ponto anterior numa folha de cálculo é

possível utilizar o método numérico da aplicação para obter todos os parâmetros da câmara. Este método numérico tenta minimizar uma função de custo C que confronta os resultados calculados através do algoritmo com os dados:

$$C = \sum_i (|x_i - \hat{x}_i| + |y_i - \hat{y}_i|) \quad (4.14)$$

onde os resultados do algoritmo são representados por (\hat{x}_i, \hat{y}_i) .

Esta é uma função de custo de primeira ordem. Os valores iniciais desta operação devem ser introduzidos de forma cuidada levando a optimização a convergir para o mínimo global.

4.3 Conclusões

Neste capítulo foi contextualizada a visão, a imagem e a cor.

A câmara foi alvo de modelização formal, tendo sido listados os parâmetros intrínsecos e extrínsecos do modelo da câmara.

Foi também ilustrado o algoritmo de mapeamento imagem mundo que permite, entre outras funcionalidades, a calibração de uma câmara numa dada aplicação. A técnica de calibração referida será utilizada em capítulos futuros.

Capítulo 5

Processos Estocásticos e Filtros de Kalman

5.1 Introdução

Este capítulo aborda algumas ferramentas teóricas utilizadas neste trabalho.

A teoria das probabilidades permite modelizar um dado fenómeno, incerto na sua natureza mas acerca do qual temos algum conhecimento [174].

O método dos mínimos quadrados que permite a estimação de um conjunto de parâmetros a partir de medidas ruidosas é também abordado.

O Filtro de Kalman [175] é uma ferramenta que utiliza as probabilidades e que se baseia no método dos mínimos quadrados para fundir toda a informação presente no sistema de forma óptima [174], [176], [177], [178].

5.2 Variáveis aleatórias

5.2.1 Fundamentos acerca de Probabilidades

Consideremos U_Y como sendo o universo de todos os valores que uma dada variável Y pode assumir (Y representa o fenómeno a modelizar). Y será então uma **variável aleatória** e se assumir uma variação temporal $Y(t)$ designa-se por **processo estocástico**.

No caso de o processo estocástico assumir apenas valores discretos para a variável t , então é usual indexar o tempo à variável discreta:

$$\begin{aligned} Y(t) &\rightarrow \text{processo estocástico} \\ t &\in \mathbb{R} \\ Y(k) &\rightarrow \text{processo estocástico em tempo discreto} \\ k &\in \mathbb{Z} \end{aligned}$$

É então necessário estabelecer uma lei de probabilidade que indique quão previsível é que Y assuma um dado valor, o mesmo é dizer, que o fenómeno sob estudo tenha um dado desfecho. A probabilidade de acontecer um desfecho A é representada por $P(Y=A)$ ou $P(A)$.

Uma dada lei de probabilidade deve ainda obedecer a:

1. $P(A) \geq 0 \forall A$
2. $P(U_Y) = 1$
3. $P(A_1 \cup A_2 \cup \dots) = P(A_1) + P(A_2) + \dots$ para qualquer sequência finita ou não de eventos mutuamente exclusivos

Definindo uma **função densidade de probabilidade** (fdp) como sendo a função que calcula para cada valor x a sua probabilidade, $f_y(x): \mathbb{R} \rightarrow \mathbb{R}$, esta fdp deve obedecer a:

1. $f_y(x) \geq 0 \forall x$
2. $\int_{-\infty}^{+\infty} f_y(x) dx = 1$

O cálculo da probabilidade de um evento A é feito através de:

$$P(A) = \int_A f_y(x) dx \tag{5.1}$$

Se a variável aleatória for discreta, então teremos que o universo U_Y é finito e assim deve-se verificar:

$$\sum_i P(Y=i) = 1 \tag{5.2}$$

É também possível ter variáveis aleatórias vectoriais $Y = (y_1, \dots, y_n)$. Teremos f_y

(X) com $X \in \mathbb{R}^n$ e será também:

$$\begin{aligned} f_y(x): \mathbb{R}^n &\rightarrow \mathbb{R} \\ (X) &\rightarrow f_y(X) \end{aligned} \quad (5.3)$$

satisfazendo

1. $f_y(x) \geq 0 \forall X \in \mathbb{R}^n$
2. $\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} f_y(x) dx_1 \dots dx_n = 1$

Para obter P(A):

$$P(A) = \int_A \dots \int f_y(X) dX \quad (5.4)$$

5.2.2 Atributos de Variáveis Aleatórias

Para simplificar o tratamento destes conceitos, é interessante utilizar restrições quanto às classes de distribuições em causa. Nesse sentido definem-se alguns atributos básicos das variáveis aleatórias.

A **média**:

$$\mu_y = E(Y) = \int_{-\infty}^{+\infty} x f_y(x) dx \quad (5.5)$$

A **variância** σ^2 e o **desvio padrão** σ :

$$\sigma_y^2 = VAR(Y) = E\left((Y - \mu_y)^2\right) = E(Y^2) - (\mu_y)^2 \quad (5.6)$$

E a **covariância** entre as variáveis aleatórias Y e Z, com médias μ_y e μ_z :

$$cov(Y, Z) = E\left((Y - \mu_y)(Z - \mu_z)\right) = E(YZ) - \mu_y \mu_z \quad (5.7)$$

No caso multidimensional em que $Y = (y_1, \dots, y_n)$ teremos então que $\mu_y = (E(y_1), \dots, E(y_n))$ e a **matriz de covariância** é a seguinte:

$$\sigma^2 = \begin{bmatrix} cov(y_1, y_1) & cov(y_1, y_2) & \dots & cov(y_1, y_n) \\ cov(y_2, y_1) & cov(y_2, y_2) & \dots & cov(y_2, y_n) \\ \vdots & \vdots & \ddots & \vdots \\ cov(y_n, y_1) & cov(y_n, y_2) & \dots & cov(y_n, y_n) \end{bmatrix} \quad (5.8)$$

Note-se que os $VAR(y_i)$ encontram-se na diagonal da matriz.

Para um processo estocástico $Y(t)$ os parâmetros média e (co)variância aparecem também em função do tempo.

Podem-se definir ainda as funções seguintes:

A **correlação** em instantes t_1 e t_2 :

$$R_y(t_1, t_2) = E(Y(t_1)Y(t_2)^T) \quad (5.9)$$

e uma matriz de **covariância entre variáveis aleatórias**:

$$C_y(t_1, t_2) = cov(Y(t_1)Y(t_2)) \quad (5.10)$$

Como se verifica que:

$$C_y(t_1, t_2) = R(t_1, t_2) - \mu_y(t_1)\mu_y(t_2) \quad (5.11)$$

teremos então para um processo de média nula:

$$R_y(t_1, t_2) = C_y(t_1, t_2) \quad (5.12)$$

Para processos estacionários verifica-se que estas funções só dependem da diferença entre os instantes de tempo $\tau = t_2 - t_1$ e assim é possível escrever $R_y(\tau)$ e $C_y(\tau)$.

5.2.3 Propriedades dos Atributos

O valor esperado de variáveis aleatórias X e Y tem as seguintes propriedades [179]:

$$\begin{aligned} E(K) &= K, & K \in \mathbb{R} \\ E(X+Y) &= E(X) + E(Y) \\ E(KX) &= K \cdot E(X), & K \in \mathbb{R} \end{aligned} \quad (5.13)$$

Se X e Y são variáveis aleatórias independentes então:

$$E(X \cdot Y) = E(X) \cdot E(Y) \quad (5.14)$$

A variância tem as seguintes propriedades:

$$\begin{aligned} VAR(K) &= 0 & K \in \mathbb{R} \\ VAR(KX) &= K^2 \cdot VAR(X) \end{aligned} \quad (5.15)$$

Se X e Y são variáveis aleatórias independentes:

$$VAR(X+Y) = VAR(X) + VAR(Y) \quad (5.16)$$

5.2.4 Expressões Aproximadas para Operações entre Atributos

Seja a variável aleatória X tal que $E[X] = \mu$ e $VAR[X] = \sigma^2$. Sendo $Y = H(X)$ tem-se [179]:

$$E[Y] \approx H(\mu) + \frac{H''(\mu)}{2} \sigma^2 \quad (5.17)$$

$$VAR(Y) \approx [H'(\mu)]^2 \sigma^2 \quad (5.18)$$

Seja (X, Y) uma variável aleatória bidimensional tal que $E[X]=\mu_x$ e $VAR[X]=\sigma_x^2$; e $E[Y]=\mu_y$ e $VAR[Y]=\sigma_y^2$. Sendo $Z=H(X, Y)$, se X e Y forem independentes:

$$E[Z] \approx H(\mu_x, \mu_y) + \frac{1}{2} \left[\left(\frac{\partial^2 H}{\partial x^2} \right)_{(\mu_x, \mu_y)} \sigma_x^2 + \left(\frac{\partial^2 H}{\partial y^2} \right)_{(\mu_x, \mu_y)} \sigma_y^2 \right] \quad (5.19)$$

$$VAR(Z) \approx \left(\frac{\partial H}{\partial x} \right)_{(\mu_x, \mu_y)}^2 \sigma_x^2 + \left(\frac{\partial H}{\partial y} \right)_{(\mu_x, \mu_y)}^2 \sigma_y^2 \quad (5.20)$$

5.2.5 Distribuições Gaussianas

A função densidade probabilidade de uma distribuição gaussiana tem a forma analítica:

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (5.21)$$

Onde μ é a sua média e σ é o seu desvio padrão.

Processos estocásticos que sigam uma distribuição de forma gaussiana são completamente definidos pela sua média e covariância:

$$\mu_y(t) = E(Y(t)) \quad \text{e} \quad C_y(t_1, t_2) = cov(Y(t_1), Y(t_2))$$

Estes processos gaussianos têm propriedades interessantes:

1. A combinação linear de processos gaussianos é ainda um processo gaussiano;
2. Muitos sinais reais são representáveis por processos gaussianos.

A primeira propriedade garante que se pode representar com exactidão e simplicidade combinações lineares de processos estocásticos.

A segunda propriedade é essencial para garantir a correcta modelização da realidade. Uma justificação da propriedade é dada pelo **Teorema do Limite Central** que refere o seguinte:

Sendo $Z(n)$, $n \in \mathbb{N}$, variáveis aleatórias independentes com variâncias $V(Z(n)) = \sigma^2 < \infty$ então a sequência $\frac{S(n) - E(S(n))}{\sqrt{V(S(n))}}$ onde $S(n) = \sum_{i=1}^n Z(i)$ converge para a distribuição gaussiana normalizada²⁶ quando $n \rightarrow +\infty$ desde que $0 < \epsilon \leq V(Z(n)) \leq C < \infty$, isto é, desde que a

²⁶ Com média nula e variância (e desvio padrão) unitários

variância esteja limitada superior e inferiormente.

Pode-se assim dizer que o efeito de muitas perturbações, independentemente das distribuições individuais, se soma para formar uma distribuição gaussiana.

Por estes motivos, a utilização de processos gaussianos é muito relevante para o estudo dos mais diversos fenómenos na actualidade.

5.3 Método dos Mínimos Quadrados

Em muitas situações reais existem dados variados de um fenómeno que se sabe obedecer a uma dada lei conhecida mas com parâmetros desconhecidos. O cálculo da estimativa dos parâmetros pode ser feita de forma expedita aplicando a teoria do **Mínimos Quadrados** [178].

Considere-se então um processo estocástico $\{Y_k, k \in (1, \dots, n)\}$ com um valor médio $E[Y_k] = x_k^T \theta$ onde x_k são elementos conhecidos de um vector onde $k=1, \dots, n$ e θ é um vector de parâmetros. A função a minimizar é:

$$S = \sum_{k=1}^n (y_k - x_k^T \theta)^2 \quad (5.22)$$

Onde $\{y_k, k \in (1, \dots, n)\}$ é uma realização do processo estocástico Y_k .

Pretende-se assim encontrar o valor $\hat{\theta}$ que minimiza S , a soma do quadrado dos desvios. É possível demonstrar que a solução existe e é única:

$$\hat{\theta} = (X^T X)^{-1} X^T Y \quad (5.23)$$

Desde que a matriz $(X^T X)$ seja invertível e com:

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad \text{e} \quad X = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix}$$

Esta teoria é compatível com qualquer sistema que possa ser manipulado até ao ponto em que medidas escalares com ruído são obtidas a partir da combinação linear de parâmetros. A modelização deve ser tal que uma medida só pode ser obtida com um determinado conjunto de parâmetros. A utilização de relações trigonométricas e outras funções periódicas em x_i é assim desaconselhável.

5.4 Filtro de Kalman

Tal como referido anteriormente um dos métodos mais utilizados para a fusão de informação é o Filtro de Kalman (KF), introduzido em 1960 por R. E. Kalman [175].

Este é um método matemático que garante uma estimação óptima no sentido em que se todos os processos forem lineares e respectivos ruídos gaussianos, então o KF minimiza o quadrado dos erros dos parâmetros estimados.

A sua popularidade é impulsionada pela sua optimalidade, pelo não elevado poder de cálculo necessário para a implementação, pela sua implementação ser recursiva e poder ser expandida para diversos tipos de medidas assíncronas. Existem também extensões para sistemas não lineares, o Filtro de Kalman Extendido (EKF) que supera algumas das limitações do método. O EKF deixa, porém, de apresentar garantias de convergência e optimalidade.

O KF utiliza na sua base o conceito de que todas as variáveis envolvidas são gaussianas e por isso são necessárias algumas definições neste campo.

A utilização da palavra “filtro” vem do facto de este método extrair uma estimativa óptima para o estado do sistema a partir de medidas afectadas por ruído. Mas o KF faz mais do que apenas isso pois ele altera o estado do sistema para reflectir as medidas e o ruído existentes.

Consideremos um sistema genérico com dinâmica linear cujas saídas são combinação linear do estado:

$$\begin{aligned}\frac{dx(t)}{dt} &= Ax(t) + Bu(t) \\ y(t) &= Cx(t)\end{aligned}\tag{5.24}$$

onde

$$x = [x_1 \ x_2 \ \dots \ x_n]^T$$

$$u = [u_1 \ u_2 \ \dots \ u_p]^T$$

$$y = [y_1 \ y_2 \ \dots \ y_q]^T$$

$\dim(A) = n \times n$, m é a dimensão do sistema

$\dim(B) = p \times n$, p é o número de entradas

$\dim(C) = q \times n$, q é o número de saídas

É possível converter este modelo em tempo contínuo para um modelo equivalente em tempo discreto:

$$\begin{aligned}x(k) &= \Phi x(k-1) + Gu(k-1) \\ y(k) &= Hx(k)\end{aligned}\tag{5.25}$$

com $k = 1, 2, \dots$ indicando o k -ésimo instante de amostragem

Temos ainda que $\Delta t_k = t_k - t_{k-1}$ e é ainda possível demonstrar que:

$$\Phi(k) = e^{A\Delta t_k} \quad (5.26)$$

$$G = \int_{t_{k-1}}^{t_k} e^{A\Delta t_k} B dt \quad (5.27)$$

$$H = C \quad (5.28)$$

O modelo anterior é um modelo determinístico. Na sua versão estocástica é necessário introduzir incerteza proveniente da dinâmica do sistema $w(k)$ e das medidas $v(k)$:

$$\begin{aligned} x(k) &= \Phi x(k-1) + Gu(k-1) + w(k-1) \\ y(k) &= Hx(k) + v(k-1) \end{aligned} \quad (5.29)$$

Para que o filtro tenha convergência garantida é necessário:

$$R_w(k, i) = \begin{cases} Q(k) & \text{se } i=k \\ 0 & \text{se } i \neq k \end{cases} \quad (5.30)$$

$$R_v(k, i) = \begin{cases} R(k) & \text{se } i=k \\ 0 & \text{se } i \neq k \end{cases} \quad (5.31)$$

$$E(w(k)v(i)^T) = 0, \forall k, i \quad (5.32)$$

Para que o filtro seja óptimo é necessário ainda que $w(k)$ e $v(k)$ sejam processos gaussianos de média nula:

$$\begin{aligned} P(w) &\sim N(0, Q) \\ P(v) &\sim N(0, R) \end{aligned} \quad (5.33)$$

5.4.1 Filtro de Kalman Discreto

O KF mantém então sempre as médias e covariâncias das variáveis do estado.

O seu funcionamento tem duas operações básicas: *previsão* e *actualização*. Na *previsão* o modelo do sistema e o estado anterior é utilizado para prever o estado actual. A covariância do estado nesta altura sobe para indicar a menor confiança na estimativa até agora. Se existirem medidas deste instante actual, então é feita uma *actualização* afectando o estado para que este reflecta toda a nova informação.

Considerando então as condições iniciais:

$$E(x(0)) = X_0 \quad cov(x(0)) = P_0 \quad (5.34)$$

E o modelo do Sistema:

$$x(k) = \Phi x(k-1) + Gu(k-1) + w(k-1) \quad (5.35)$$

Com as observações:

$$y(k) = Hx(k) + v(k) \quad (5.36)$$

Propagação de estado será então:

$$x(k^-) = \Phi x(k-1) + Bu(k-1) \quad (5.37)$$

A associada covariância:

$$P(k^-) = \Phi P(k-1) \Phi^T + Q(k) \quad (5.38)$$

Para a actualização deve-se calcular o Ganho de Kalman:

$$K(k) = P(k^-) H^T (HP(k^-) H^T + R(k))^{-1} \quad (5.39)$$

E actualizar a covariância:

$$P(k) = (I - K(k)H) P(k^-) \quad (5.40)$$

Actualização do estado:

$$x(k) = x(k^-) + K(k) (y(k) - Hx(k^-)) \quad (5.41)$$

Por vezes e em vez da equação 5.40 pode utilizar-se:

$$P(k) = (I - K(k)H) P(k^-) (I - K(k)H)^T + K(k) R K(k)^T \quad (5.42)$$

A equação 5.42 é mais complexa mas é válida para qualquer ganho $K(k)$, mesmo que este seja não óptimo, o que pode ser interessante para algumas aplicações práticas.

Observando 5.39 pode-se concluir que quando a covariância do erro da medida R tende para 0, o ganho K faz pesar com maior importância a parcela denominada de *inovação* $(y(k) - Hx(k^-))$ da equação 5.41. Em concreto fica:

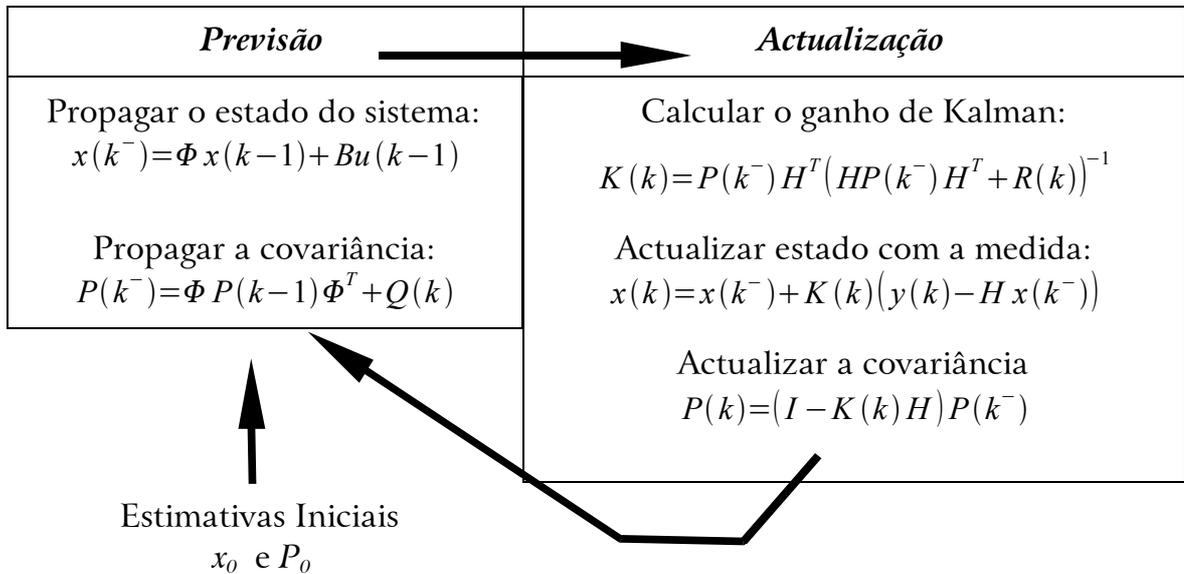
$$\lim_{R_k \rightarrow 0} K_k = H^{-1} \quad (5.43)$$

Se acontecer que a covariância da estimativa do erro do sistema P_k^- se aproxime de 0, então a parcela inovação terá um peso inferior na equação 5.41 tal como se pode verificar a partir de:

$$\lim_{P_k^- \rightarrow 0} K_k = 0 \quad (5.44)$$

Como resumo atente-se à tabela 5.1 que ilustra o funcionamento do KF.

Tabela 5.1 Resumo da operação do Filtro de Kalman



5.4.2 Filtro Kalman Extendido

O Filtro de Kalman Extendido (EKF) deve ser utilizado em sistemas não lineares. Este filtro, para cada instante lineariza a função de transferência em torno do valor estimado e actualiza as estimativas baseado num KF aplicado a esse sistema linear.

O EKF não apresenta garantias teóricas de convergência nem de optimalidade mas na prática um projecto cuidadoso evita que estes problemas se manifestem. O EKF é o melhor filtro linear possível para um sistema não linear. Filtros não lineares podem ser superiores.

Perante as seguintes condições iniciais:

$$E(x(0)) = X_0, \quad cov(x(0)) = P_0$$

Considere-se o sistema estocástico não linear com as dimensões anteriormente explicitadas para o caso do KF discreto:

$$\begin{aligned} x(k) &= f(x(k-1), u(k), w(k-1)) \\ y(k) &= h(x(k), v(k)) \end{aligned} \tag{5.45}$$

Onde $w(k)$ e $v(k)$ mantêm também as restrições anteriormente apresentadas. Estes valores dos ruídos do sistema e da medida são, claro, desconhecidos e têm média nula. Será portanto válida a aproximação:

$$\begin{aligned} \tilde{x}(k) &= f(\hat{x}(k-1), u(k), 0) \\ \tilde{y}(k) &= h(\hat{x}(k), 0) \end{aligned} \tag{5.46}$$

Onde \hat{x} é uma estimativa à *posteriori* do estado (resultando de uma iteração anterior do KF).

Linearizando as equações do modelo do sistema e das medidas em torno dos resultados 5.46:

$$\begin{aligned} x(k) &\approx \tilde{x}(k) + A(x(k-1) - \hat{x}(k)) + Ww(k-1) \\ y(k) &\approx \tilde{y}(k) + h(x(k) - \tilde{x}(k)) + Vv(k) \end{aligned} \quad (5.47)$$

É necessário então conhecer os jacobianos da descrição do modelo e das observações:

$$A = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (5.48)$$

$$H = \frac{\partial h}{\partial x} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \dots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \dots & \frac{\partial h_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_q}{\partial x_1} & \frac{\partial h_q}{\partial x_2} & \dots & \frac{\partial h_q}{\partial x_n} \end{bmatrix} \quad (5.49)$$

Com o jacobiano do ruído:

$$W = \frac{\partial f}{\partial w} = \begin{bmatrix} \frac{\partial f_1}{\partial w_1} & \frac{\partial f_1}{\partial w_2} & \dots & \frac{\partial f_1}{\partial w_n} \\ \frac{\partial f_2}{\partial w_1} & \frac{\partial f_2}{\partial w_2} & \dots & \frac{\partial f_2}{\partial w_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial w_1} & \frac{\partial f_n}{\partial w_2} & \dots & \frac{\partial f_n}{\partial w_n} \end{bmatrix} \quad (5.50)$$

$$V = \frac{\partial h}{\partial v} = \begin{bmatrix} \frac{\partial h_1}{\partial v_1} & \frac{\partial h_1}{\partial v_2} & \dots & \frac{\partial h_1}{\partial v_n} \\ \frac{\partial h_2}{\partial v_1} & \frac{\partial h_2}{\partial v_2} & \dots & \frac{\partial h_2}{\partial v_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_q}{\partial v_1} & \frac{\partial h_q}{\partial v_2} & \dots & \frac{\partial h_q}{\partial v_n} \end{bmatrix} \quad (5.51)$$

Para simplificar a notação não se explicitou a dependência de A , W , H e V do instante de amostragem k .

Pode-se agora definir os erros de predição e o resíduo da medida:

$$\begin{aligned} \tilde{e}_x(k) &\equiv x(k) - \hat{x}(k) \\ \tilde{e}_y(k) &\equiv y(k) - \hat{y}(k) \end{aligned} \quad (5.52)$$

Pode-se assim descrever os erros:

$$\begin{aligned} \tilde{e}_x(k) &\approx A(x(k) - \hat{x}(k-1)) + \epsilon(k) \\ \tilde{e}_y(k) &\approx H \tilde{e}_x(k) + \eta(k) \end{aligned} \quad (5.53)$$

onde $\epsilon(k)$ e $\eta(k)$ são variáveis aleatórias independentes com média nula e covariâncias WQW^T e VRV^T respectivamente, com Q e R a obedecerem às mesmas restrições referidas para o caso linear.

São notórias as semelhanças das equações apresentadas em 5.53 com as equações de um KF discreto. Pode-se então considerar um KF hipotético para estas variáveis em que a estimativa à *posteriori* do processo inicial será:

$$\hat{x}(k) = \hat{x}(k) + e(k) \quad (5.54)$$

com:

$$\begin{aligned} p(\tilde{e}_x(k)) &\sim N\left(0, E[\tilde{e}_x(k)\tilde{e}_x^T(k)]\right) \\ p(\epsilon_x(k)) &\sim N\left(0, E[WQ(k)W^T]\right) \\ p(\eta_x(k)) &\sim N\left(0, E[VR(k)V^T]\right) \end{aligned}$$

Fazendo que o valor esperado de $\hat{e}(k)$ seja 0 vem:

$$\hat{e}(k) = K_k \tilde{e}_y(k) \quad (5.55)$$

o que resulta em:

$$\hat{x}(k) = \hat{x}(k) + K(k) \tilde{e}_y(k) = \hat{x}(k) + K(k)(y(k) - \hat{y}(k)) \quad (5.56)$$

Que é a equação da actualização do EKF.

A propagação do estado será:

$$\hat{x}(k^-) = f(\hat{x}(k-1), u(k), 0)$$

A propagação da covariância:

$$P(k^-) = A(k)P(k-1)A(k)^T + W(k)Q(k-1)W^T(k) \quad (5.57)$$

O cálculo do ganho de Kalman $K(k)$:

$$\begin{aligned} L(k) &= (H(k)P(k^-)H(k)^T + V(k)R(k)V(k)^T)^{-1} \\ K(k) &= P(k^-)H(k)^T L(k) \end{aligned} \quad (5.58)$$

Actualização da covariância:

$$P(k) = (I - K(k)H(k))P(k^-) \quad (5.59)$$

E finalmente a actualização do estado:

$$x(k) = x(k^-) + K(k)(y(k) - h(x(k^-), 0)) \quad (5.60)$$

O EKF utiliza uma dinâmica linearizada pelo que mesmo que se a estimativa inicial do estado $x(0)$ seja gaussiana, não há garantias que assim se mantenha ao longo do tempo. A propagação da covariância é também feita com recurso à linearização em torno do estado estimado. Estes dois factores de imprecisão são tanto mais importantes quanto menos válida for a aproximação utilizada.

Na discussão acima considera-se que o EKF utiliza a matriz A do processo dinâmico discretizado.

Como procuramos uma aproximação da dinâmica do sistema em tempo discreto, parte-se da seguinte descrição dinâmica do sistema no tempo contínuo:

$$\frac{dX(t)}{dt} = f(X(t), u(t_k), t), \quad t \in]t_{k-1}, t_k] \quad (5.61)$$

Calcula-se de seguida:

$$A_k = \left. \frac{\partial f}{\partial x} \right|_{\substack{x=x(t_k) \\ u=u(t_k) \\ t=t_k}} \quad (5.62)$$

E considerando os valores constantes entre instantes de amostragem obtemos finalmente:

$$A(k) = \exp(A_k(t_k - t_{k-1})) \quad (5.63)$$

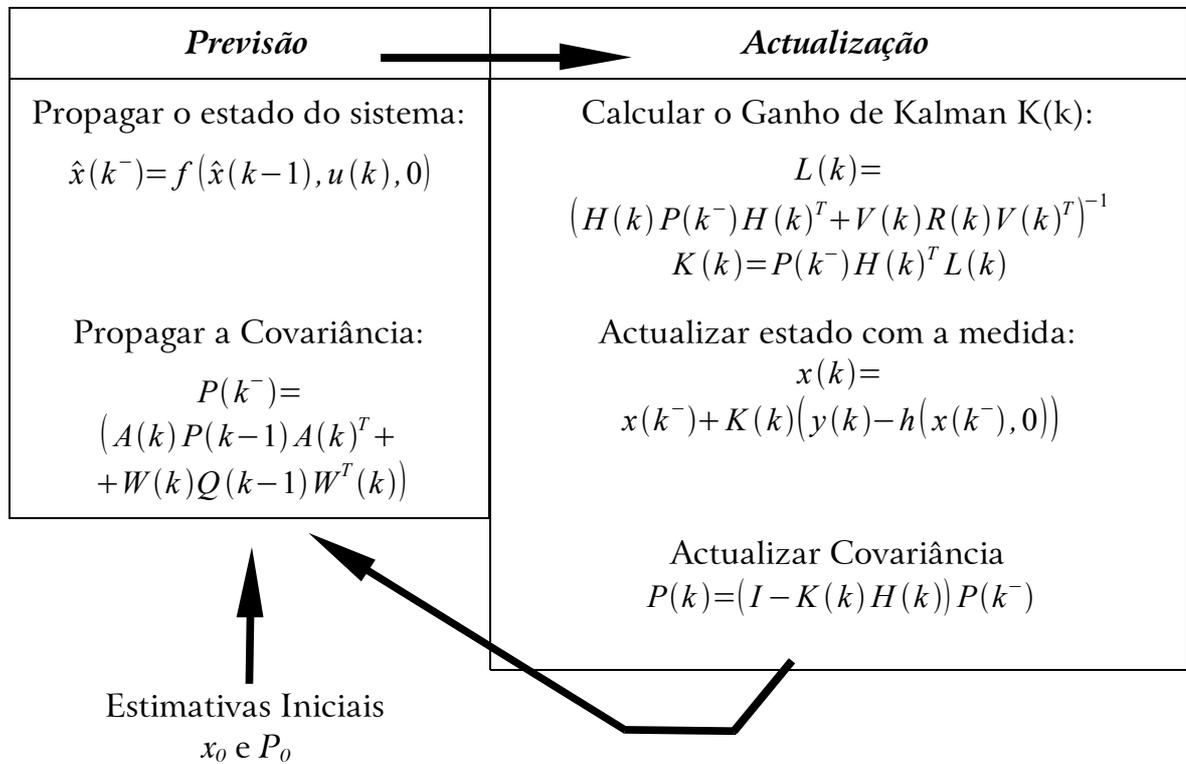
Uma característica importante dos Filtros de Kalman e do EKF em particular é que o jacobiano H_k da equação 5.58 que calcula o ganho de Kalman propaga correctamente a informação que a medida traz acerca de cada elemento do

estado. Desta forma se a medida não afectar todos os elementos do estado as covariâncias do erro do estado são afectadas de acordo com essa situação e a informação que a medida traz.

O KF e o EKF só lidam com um conjunto de medidas sempre disponível. A implementação pode, no entanto, contornar a inexistência de medidas introduzindo covariâncias grandes nas medidas não disponíveis mas tal procedimento poderá trazer problemas numéricos. Outra solução possível é repetir a *actualização* para as diferentes medidas que existam de facto. O processo descrito é uma aproximação à teoria mas é utilizado na prática [176].

Se o conjunto das medidas não trazer (nunca) informação acerca de todas as partes do vector de estado, então o filtro torna-se divergente visto o processo não ser observável.

Tabela 5.2 Resumo da operação do EKF



5.4.3 Teste Estatístico

Como o estado é representado por uma distribuição gaussiana, é possível utilizar testes estatísticos *standard* para aferir da qualidade da medida que pode ser errada por exemplo por motivos de avaria de um sensor.

Esta técnica aparece referida por vezes como “*validation gate*”, porta de validação

que faz um teste de verosimilhança à medida antes de a considerar como válida.

Utiliza-se o seguinte teste estatístico:

$$\begin{aligned} M(k, t_k) &= (y(k) - h(x(k), t_k)) \\ L(k) &= (H(k)P(k^-)H(k)^T + V(k)R(k)V^T(k))^{-1} \\ e(k)^2 &= M(k, t_k)^T K^{temp}(k)M(k, t_k) \leq g^2 \end{aligned} \quad (5.64)$$

M é também apelidado de inovação e multiplica o ganho de Kalman na equação 5.34 (pág 90) de actualização do estado do EKF; L é uma matriz intermédia que aparece na fórmula do cálculo do ganho de Kalman. O erro normalizado e^2 varia com a distribuição do χ -quadrado com tantos graus de liberdade como a dimensão q do vector das medidas. O valor g^2 é um determinado nível de confiança.

Se uma dada medida $y(k)$ apresentar um erro $e(k) > g$ então essa medida falhará este teste estatístico que pode também ser visto como um teste de verosimilhança.

5.5 Conclusões

Neste capítulo fez-se a introdução aos processos estocásticos como sendo ferramentas matemáticas adequadas à representação de conhecimento parcial.

Apresentou-se ainda o formulário relativo ao método dos Mínimos Quadrados que permite estimar parâmetros dado um certo modelo e a partir de resultados experimentais ruidosos.

Foi também mostrado o Filtro de Kalman e respectiva versão Extendida. Estes filtros permitem reunir informação de várias fontes e fundi-la numa estimativa global para o estado do sistema. Esta estimativa terá elevada qualidade pois a fusão de informação é realizada tomando em consideração a modelização dos erros associados a cada tipo de medidas. No entanto, não há garantias teóricas de convergência do EKF.

Capítulo 6

Codificador de Janela Deslizante

6.1 Introdução aos Códigos de Barras

Os códigos de barras são de utilização generalizada nos dias de hoje. Permitem armazenar de forma pouco dispendiosa informação que pode ser lida de forma expedita e robusta. Os códigos de barras podem assumir várias formas de acordo com diferentes normas, tal como ilustrado na figura 6.1. Um dos mais frequentes para o consumidor final pode ser encontrado nos livros veiculando informações tal como o preço de capa e o ISBN - *International Standard Book Number*. Os códigos de barras mais frequentes são uni-dimensionais e a sua estrutura é propícia a serem lidos de perto. A informação pode estar codificada na espessura da barra ou no espaço entre barras ou na altura das barras. Para robustez acrescida, os códigos de barras utilizam apenas barras pretas. Devido à forma como é gravada a informação nos códigos de barras, é possível ler a informação sem um alinhamento perfeito entre leitor e código o que os torna também de fácil leitura automatizada. Os dispositivos leitores deste género de códigos de barras tipicamente utilizam iluminação a LASER mecanicamente associada ao sensor linear de imagem. O processamento necessário para recuperar a informação é limitado e a quantidade de dados em causa não é elevada.

Os códigos de barras podem também assumir duas dimensões. A densidade de

informação neste caso é muito superior à custa de menor robustez e maior complexidade na leitura. Na figura 6.1b) observe-se as duas regiões assinaladas que veiculam a mesma informação. A leitura destes códigos bidimensionais deve ser feita por uma câmara matricial e é utilizado processamento de imagem para fazer o tratamento da imagem que permite uma leitura correcta na presença de alinhamentos imperfeitos entre código a ler e câmara. A quantidade de dados a processar é elevada devido à imagem ser bidimensional. O sistema de leitura destes códigos presume geralmente distâncias curtas e a iluminação é feita com LEDs de elevado brilho directamente em torno da câmara para evitar a influência da iluminação externa.

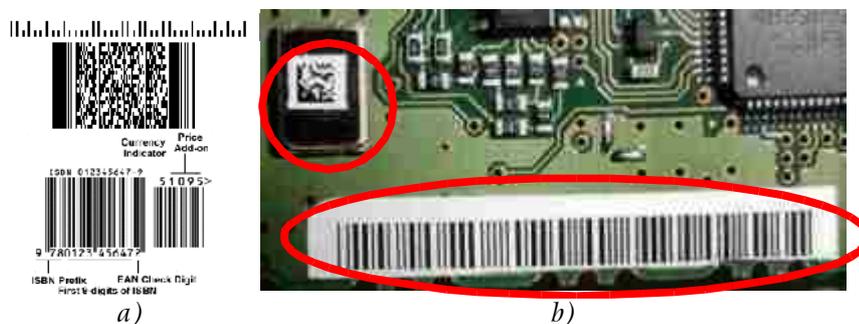


Figura 6.1a) Diversos exemplos de códigos de barras; b) os dois códigos assinalados contêm a mesma informação

Na figura 6.1 estão representados diversos tipos de códigos de barras que podem ser encontrados no mercado. Tomando a figura 6.1a) e percorrendo-a de cima para baixo temos: o código de barras do sistema postal norte-americano; um código bidimensional de acordo com a norma PDF417 que transporta um ficheiro PDF²⁷; um código de barras que pode estar presente em livros de acordo com uma norma europeia EAN que regista um número unívoco para o livro (o ISBN) e ainda informação adicional.

6.2 Introdução aos Codificadores

Os codificadores (*encoders*) são equipamentos de utilização generalizada em robótica. Podem ter diferentes resoluções e fornecer informação absoluta ou incremental. A figura 6.2b) ilustra um codificador absoluto de 10 bits – 10 anéis concêntricos. Uma sequência de sensores dispostos radialmente lê cada bit a partir de cada anel. O padrão obtido ao longo da periferia não se repete fornecendo assim uma medida angular absoluta.

Um codificador incremental é mais simples, tal como mostrado na fig. 6.2a). Neste sensores, a informação repete-se periodicamente. Pode no entanto existir

²⁷ *Portable Document Format* – formato muito utilizado para troca de informação na *Internet*

uma referência absoluta, uma marca que pode ser vista no anel exterior, às 12 horas na fig. 6.2a). Através desta marca é possível em muitas situações obter informação equivalente à do codificador absoluto.

Considera-se que no interior de um codificador cada leitura de um sensor óptico gera um sinal 0 ou 1. Num codificador absoluto, a medida angular será obtida directamente pela conversão de um tipo de código binário para decimal. Por razões de implementação utiliza-se o código binário *gray* e não o código binário comum.

Existem também codificadores lineares nas suas versões absoluta e incremental. Estes sensores utilizam as mesmas técnicas para medir deslocamentos em vez de ângulos.

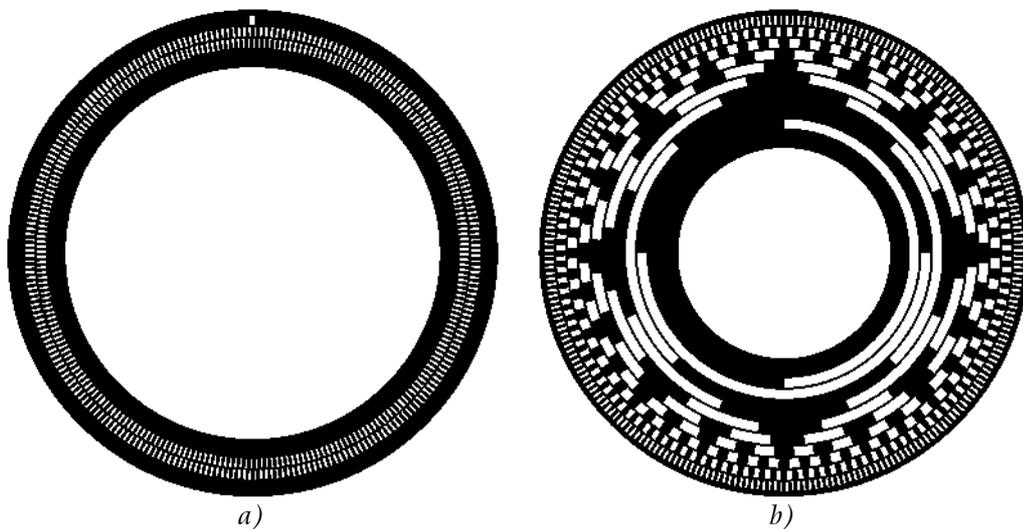


Figura 6.2 Discos de codificadores para medida angular: a) versão incremental; b) versão absoluta

6.3 Definições

Um código de barras é lido assumindo que é visualizado na sua totalidade. Já um codificador incremental fornece informação parcial angular. Estude-se pois a possibilidade de obter a posição dentro da sequência a partir dos dados de uma *janela*, uma região limitada de um código extenso conhecido.

Considere-se uma sequência S de dígitos de base b e de comprimento n :

$$S = (s_1, s_2, \dots, s_n), \quad s_i \in [0, b-1] \quad (6.1)$$

Da sequência completa, a “janela” permite adquirir apenas j dígitos em simultâneo. Tome-se s_i como o i -ésimo dígito da sequência. Defina-se também $J_j(i)$ como sendo o operador “janela” que recupera a “imagem” com j dígitos de

comprimento, vista da posição i em diante. Para $i > n - j + 1$ a função J utiliza dígitos do início da sequência. A sequência S fica então circular, isto é, com re-entrada fim para o princípio. Na presença de circularidade, a sequência representa-se por \hat{S} e a função janela que “recolhe a imagem” da sequência passa a ser:

$$J_j(S, i) = \begin{cases} (s_i, s_{i+1}, \dots, s_{i+j-1}) & \text{se } 1 \leq i \leq n - j + 1 \\ (s_i, \dots, s_n, s_1, \dots, s_{(i+j)-n-1}) & \text{se } n - j + 1 < i \leq n \end{cases} \quad (6.2)$$

Procure-se então uma sequência S constituída apenas por janelas que não se repetem, isto é, que verifique:

$$J_j(S, a) \neq J_j(S, b) \Leftrightarrow a \neq b, \quad \forall a, b \in [1, n] \quad (6.3)$$

O tamanho máximo da sequência S será a situação em que todas as combinações de base b com comprimento j estão incluídas dentro de S . Será então $\hat{n} = b^j$ para uma sequência circular.

Como exemplo tome-se a sequência $\hat{S}_3^{exemplo} = 00010111$ em base 2 ($b=2$). Esta é uma sequência circular \hat{S}_3 válida (entre outras existentes) pois nenhum grupo de 3 bits adjacentes aparece em posições diferentes da sequência, mesmo quando considerada circular.

A sequência referida pode ser extendida para o caso não circular com $\bar{S}_3^{exemplo} = 0001011100$. Generalizando esta expansão verifica-se que a ausência de circularidade permite aumentar a sequência (por exemplo no seu final) em $j-1$ bits. A função que recolhe a imagem passa a ser representada por:

$$\bar{S}_j(i) = (s_i, s_{i+1}, \dots, s_{i+j-1}) \quad \text{para } 0 < i < n - j + 1 \quad (6.4)$$

Onde $\bar{n} = b^j + j - 1$.

A sequência será circular ou não conforme a aplicação em causa e assim pode utilizar-se sem causar confusão a função $J_j(S, i)$ genérica:

$$J_j(S, i) = \begin{cases} \hat{J}_j(S, i) & \text{se } S \text{ é circular} \\ \bar{J}_j(S, i) & \text{se } S \text{ é não circular} \end{cases} \quad (6.5)$$

6.4 Sequências Unívocas

Para maior facilidade de linguagem utiliza-se a expressão “**sequência unívoca**” em vez de “sequência constituída por janelas unívocas” ou, de forma ainda mais

correcta, “sequência em que todas as porções da largura da janela contêm códigos que não se repetem noutra zona da sequência”.

Recorde-se que para uma dada sequência unívoca S a função $J_j(S,i)$ devolve j dígitos de base b . Estes dígitos podem ser considerados um número de base b :

$$\begin{aligned} S &= 0001011100, & \text{com } b &= 2 \\ J_3(S, 4) &= 101_{(2)} = 5_{(10)} \end{aligned} \quad (6.6)$$

Desta forma é possível estabelecer uma tabela de pesquisa que mapeia o resultado da função janela em posição dentro da sequência. Esta tabela de pesquisa tem tamanho n . A implementação tabelável é importante para implementação fácil e eficiente em sistemas de baixo poder de cálculo tal como MCUs. Esta tabela tem ainda a vantagem adicional de ser apenas de leitura para uma dada implementação.

O sistema que adquire uma janela numa sequência unívoca e devolve uma posição linear ou angular designa-se **codificador de janela deslizante**.

Se um dado sistema tiver acesso a uma janela de visão maior do que a janela da sequência, então existirá um certo nível de confirmação na medida. Sendo verdade que todas as combinações do tamanho da janela foram provavelmente utilizadas, os dígitos extra podem ser confrontados com a sequência e assim eventualmente determinar erros de leitura.

6.5 Exemplo de Implementação

Como exemplo de aplicação real de codificadores de janela deslizante, veja-se o robot da figura 6.4. Este é um robot futebolista para a liga F2000, numa geração anterior ao apresentado na secção 7.4, pág. 110. Este robot tinha uma câmara que roda em torno da vertical num mecanismo apelidado de “pescoço”. Com o objectivo de medir o ângulo do pescoço face ao corpo do robot, era empregado um codificador de janela deslizante. A medida era conseguida sem acrescentar qualquer sensor graças ao facto da câmara ver o próprio corpo do robot e aí realizar a medição. A figura 6.3 apresenta o codificador efectivamente utilizado, que implementa uma sequência circular de 160 bits e uma janela de 8 bits.

Considere-se a sequência unívoca S definida anteriormente em 6.1 e com as restrições apresentadas por 6.2 e 6.3. Acrescente-se a restrição adicional:

$$J_j(S, i) \notin M_j, \quad \forall i \in [1, n] \quad (6.7)$$

$$M_j = \left\{ (s_1^1, \dots, s_b^1), (s_1^2, \dots, s_b^2), \dots, (s_1^m, \dots, s_b^m) \right\}$$

em que M_j é o conjunto de todos os “números” indesejáveis de base b e largura j .

Esta formalização é interessante para certas aplicações com dificuldades em identificar determinadas sequências. Para uma aplicação hipotética que utilize uma câmara, pode não conveniente obrigar a que existam sempre transições de cor em todas as janelas da sequência.

A implementação construída permite gerar sequências sem qualquer conjunto arbitrário de números indesejáveis M_j .

Como exemplo, apresenta-se os resultados da geração de sequências com uma quantidade limitada de dígitos repetidos consecutivos. Estas sequências são mostradas nas figuras 6.6, 6.7 e 6.8. Para outros exemplos consultar o Anexo 3 Tabela de Algumas Sequências Unívocas, pág. 200. Nestes casos não há garantias teóricas quanto ao tamanho da sequência.

A1	B2	C3	D4	E5	F6	G7	H8	I9	J10	K11	L12	M13	N14	O15	P16	Q17	R18	S19
0	0	0	0	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1																		

Figura 6.6 Sequência unívoca Binária, Janela=4, sem restrições

A1	B2	C3	D4	E5	F6	G7	H8	I9	J10	K11	L12	M13	N14	O15	P16	Q17
0	0	0	1	0	1	1	1	0	1	0	0	1	1	0	0	0
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1																

Figura 6.7 Sequência unívoca Binária, Janela=4, sem sequências de 4 dígitos iguais consecutivos

A1	B2	C3	D4	E5	F6	G7	H8	I9	J10
0	0	1	0	0	1	1	0	1	1
A	B	C	D	E	F	G	H	I	J
1									

Figura 6.8 Sequência unívoca Binária, Janela=4, sem sequências de 3 nem de 4 dígitos iguais consecutivos

6.8 Geração de Sequências Unívocas

A geração da sequência unívoca não está sujeita a restrições de Tempo Real. Ainda assim dada a explosão combinatória que este problema assume, um bom algoritmo é necessário na prática para os obter em tempo aceitáveis com um PC vulgar. Se se fossem testar todas as combinações possíveis para uma sequência de base b com tamanho n , seria necessário avaliar b^n ramos da árvore, o que é claramente inaceitável, por ex. $b=2$, $j=6$, circular, $n=2^6$ implica avaliar 2^{64} combinações possíveis; se cada avaliação demorar 1 micro segundo então será necessário mais de 213 milhões de dias de processamento.

O algoritmo implementado vai criando a sequência pretendida à custa de acrescentar progressivamente dígitos à sequência inicial. A sequência inicial pode ser dada ou procurada sequencialmente do número 0 em diante²⁸. É mantido um vector `UsedNumber` que mantém todos números já utilizados. Um número neste contexto é o resultado de uma conversão para decimal de uma sequência de dígitos de base b com largura j . É necessário manter outra lista `BadSeqDigit` que contém para a posição actual dentro da sequência o conjunto de todos os dígitos que já foram tentados. Chegando-se a uma situação em que todos os dígitos possíveis não são permitidos, então faz-se o necessário *backtracking* que inclui apagar o ultimo character que se pensava fixo, incluí-lo no lista `BadSeqDigit` e marcar em `UsedNumber` a sequência que afinal está livre.

Através deste funcionamento temos uma pesquisa através da árvore de soluções que elimina os ramos inviáveis da árvore extremamente cedo e permite assim convergências rápidas.

Antes do arranque deste algoritmo são marcadas como já estando utilizadas todos os números indesejáveis por exemplo aqueles com repetições excessivamente longas do mesmo dígito.

O algoritmo pode ser resumido da seguinte forma:

```
if SeqCircular then N=b^j else N=b^j+j-1

UsedNumber[0..MaxNumber]:=false
BadSeqDigit[0..N]:=empty_set

for i:=0 to MaxDigits-1 do
  if IsNumberBad(i) then
    Used[i]:=True

Seq:=GetinitialSeq()
pos:=WindowSize
FlagExit:=true

while (FlagExit) do
  cur_number:=DecodeBaseCircular
    (Seq+'0',b,j,Len(Seq)-WindowSize+2, SeqCircular)
```

²⁸Entenda-se o número 0 como sendo 0...0 com largura j , na base b

```

    k:=0
    while (Used[number+k] or (k in BadSeqDigit[pos+1])) and (k<base)
do
    k:=k+1
endwhile

    if k<base then
        Seq:=Seq+Char(ord_zero+k)
        used[number+k]:=True
        pos:=pos+1
    else
        //backtracking
        BadSeqDigit[pos]:=BadSeqDigit[pos]+[(ord(Seq[Len(Seq)])-
ORD_Zero)]
        if Len(Seq)>Len(StableSeq) then StableSeq:=Seq
        Seq:=copy(Seq,1,Len(Seq)-1)
        number:=DecodeBaseCircular
            (Seq,b,j,Len(Seq)-WindowSize+1,SeqCircular)
        used[number]:=False; // marca afinal esta' livre
        BadSeqDigit[pos+1]:=[]
        pos:=pos-1
        if (pos<WindowSize) then
            FlagExit=true; // Fim da pesquisa
        endif
    endif

endwhile // FlagExit

```

A rotina `DecodeBaseCircular(Seq,b,j, pos, CircularOrNot)` trata de devolver o inteiro convertido a partir do número de base `b` com comprimento `j` na posição `pos` da string `Seq`, considerada circular ou não de acordo com o parâmetro `CircularOrNot`.

Uma tabela de algumas sequências unívocas e respectivos tempos de processamento encontra-se no Anexo 3: Tabela de Algumas Sequências Unívocas, pág. 200.

6.9 Conclusões

Foram formalmente apresentados os inovadores conceitos de “codificador de janela deslizante” e “sequência constituída por janelas unívocas” (ou “sequência unívoca”).

Foi discutido um algoritmo para a geração de sequências unívocas. Este algoritmo é baseado numa pesquisa otimizada e permite convergências em tempos curtos (cálculo sem restrições temporais de Tempo Real).

Os conceitos apresentados dão lugar a diversas aplicações tal como medida angular absoluta e medida linear absoluta. A parte experimental beneficiará ainda dos conceitos apresentados para uma aplicação de auto-localização em

robótica.

Capítulo 7

Implementações Relevantes para o Ambiente RoboCup

7.1 Introdução

As competições robóticas são simultaneamente visualmente apelativas e didáticas. Talvez por essa razão aparecem muitas competições [180] nesta área que desperta o interesse de público em geral, de professores, de alunos e ainda de investigadores. A definição correcta de um dado problema de investigação contribui para o desenvolvimento nessa área e a competição permite a rápida comparação da qualidade das soluções obtidas. Muitos dos resultados obtidos podem posteriormente ser facilmente transportados para outras aplicações tais como armazéns automáticos, sistemas de transporte de materiais fabris, sistemas de entregas em escritório, higiene, aplicações de vigilância, salvamento, etc..

O RoboCup [181], [182] é um exemplo deste tipo de competições robóticas porque fornece uma banca de ensaios padrão a nível mundial para a robótica. O autor está fortemente envolvido em competições de futebol robótico quer na liga F180 (desde 1998) quer na liga F2000 (desde 1999). Surgiram assim várias gerações de robots totalmente concebidas dentro da FEUP [16],[17],[71]. Foram assim retirados diversos ensinamentos, muitos dos quais apresentados neste documento.

7.2 Federação RoboCup

A *RoboCup Federation* [181], [182] fornece várias ligas onde testes padrão permitem catalizar a investigação científica em problemas diferentes de liga para liga. As ligas excepto o salvamento robótico (*rescue*) e a liga de humanóides são totalmente autónomas. Uma descrição geral pode ser consultada em [183].

A descrição que se segue diz respeito às regras relativas ao ano de 2003.

O desafio colocado pelo RoboCup aos investigadores mundiais subdivide-se em 3 categorias principais [181]:

- *RoboCup Soccer* – Futebol Robótico, incluindo uma liga de simulação e quatro ligas robóticas com regras distintas;
- *RoboCup Rescue* – Investigação no domínio da busca e salvamento em grandes catástrofes. A categoria divide-se numa modalidade de simulação e uma modalidade robótica;
- *RoboCup Júnior* – *RoboCup* ao serviço da educação. Crianças e jovens utilizando uma infra-estrutura simples, criam equipas de robots para jogar futebol (1 contra 1 ou 2 contra 2), dançar ou resgatar vítimas.

O Futebol robótico divide-se em 5 ligas principais e diversos desafios associados:

- Liga de **Simulação**;
 - Equipas de 11 agentes de software independentes;
 - Ambiente simulado, jogando um desafio de futebol virtual;
 - Sistema de simulação *soccerserver*;
 - Promoção de autonomia:
 - Limitação de qualidade nas percepções;
 - Introdução de limitações nas comunicações;
- Liga de Robots Pequenos – **F180**:
 - Equipas de 5 robots de 18 cm de diâmetro;
 - Visão e controlo centralizados;
 - Campo com dimensões de 2.8 m por 2.4 m com paredes a 45°²⁹;
 - Promoção de autonomia através da utilização da caixa de arbitragem (*referee box*):
 - Movimentação manual dos robot é proibida;
 - Notificação automática das decisões do árbitro;

²⁹ Está previsto as paredes serem retiradas em 2004 e é também possível que a dimensão do campo aumente

- Os jogadores podem movimentar-se normalmente durante as pausas do jogo (faltas, etc.) desde que cumpram tempos e distâncias especificadas nas regras;
- Liga de Robots Médios – **F2000**:
 - Equipas de 4 robots de 50 cm por 50 cm;
 - Robots de elevada autonomia, que utilizam apenas sensores internos;
 - Campo de cerca de 5 m por 10 m;
 - Promoção de autonomia através da redução da estruturação do ambiente removendo as paredes do campo
- Liga de Robots com Pernas – *Sony Legged*:
 - Equipas de 4 robots cães *Aibo* da *Sony*;
 - Campo de cerca de 3 m por 5 m;
- Liga de robots **humanóides** – Espera-se uma forte evolução neste ambicioso domínio. Neste momento têm existido unicamente demonstrações e competições de velocidade de deslocamento bípede e marcação de penalidades.

Para além das ligas principais do RoboCup, existem diversos desafios diferentes de liga para liga. Exemplos de desafios são a competição para elaborar um treinador inteligente e desafios próprios de cada liga (ex: competição para marcar o máximo de golos com oponentes parados).

Todos os anos, de forma a aferir e incentivar o progresso científico realizado pelos vários grupos de investigação nos diversos tópicos mencionados, a Federação *RoboCup* organiza o evento *Robot World Cup Soccer Games and Conferences*, vulgarmente conhecido por *RoboCup*.

Existem ainda diversas competições e eventos regionais relacionados com o *RoboCup* tais como o que se poderia chamar “campeonato Europeu”, o *German Open*. Noutros pontos do globo existem também eventos regionais tal como o *Japan Open*, o *China Open*, *Singapura Open*, *Australian Open*, *American Open*, *Iranian Open*, etc..

Em Portugal existe também o Festival Nacional de Robótica que tem assumido importância crescente. Tomando o evento de 2003, este festival incluiu algumas demonstrações e competições ligadas à federação *RoboCup* mas não é limitado apenas a estas modalidades.

7.3 Robots futebolistas F180 (ano 2003)

7.3.1 Requisitos

De acordo com a metodologia identificada no Capítulo 2, os seguintes requisitos foram identificados no início do projecto [16]:

- Tamanho máximo de acordo com as regras [181]:
 - 18 cm de diâmetro;
 - 15 cm de altura;
- Locomoção Omni-Direccional:
 - Velocidade de ponta superior a 1 m/s;
 - Acelerações superiores a 5 ms⁻²;
- Manipuladores:
 - Dispositivo de chute da bola com possibilidade de força controlada e tempo mínimo para “armar” o chute;
 - Zona viável para chutar deve ser máxima;
 - Capacidade de dominar a bola – eventualmente com um *dribbler* – estes dispositivos que fazem a bola rolar contra o corpo do robot são quase omni-presentes nesta liga;
- Alimentação:
 - Puramente eléctrica;
 - Autonomia mínima de 1 hora;
- Admite-se a ausência de sensores locais;
- Comunicações do exterior para o robot através de:
 - 2 frequências RF;
 - Infravermelhos (permite treinar sem interferir com o exterior);

Para contornar o reduzido tamanho do robot, as regras desta competição permitem a utilização de um sistema de visão e controlo centralizado, externo ao robot. Para estes sistemas, foram identificados os seguintes requisitos:

- Sistema de Visão:
 - Identificação de marcadores coloridos de 4 cm de diâmetro;
 - Precisão de posicionamento mínima de cerca de 15mm e orientação cerca de 0.01 rad;

- Suportar objectos (marcadores/bola) extremamente rápidos (na ordem dos 5 m/s);
- Suporte a diferentes níveis de iluminação;
- Altura máxima do sistema de aquisição de visão de 3 m;
- Controlo:
 - Seguimento de: 5 robots da equipa, 5 adversários e da bola
 - Minimizar o atraso total no ciclo de visão e controlo (alguns adversários mencionam cerca de 80 ms);
- Decisão:
 - Suporte à utilização da caixa de arbitragem (*Referee Box*);
 - Implementação de estratégias para cumprir os objectivos do jogo;

7.3.2 Descrição Geral

O robot desenvolvido é apresentado de forma muito sumária na figura 7.1. A tampa do robot apresenta um código de barras sem repetição angular em torno de um marcador com a cor da equipa. O código de barras lido fornece a informação de identificação do robot e seu ângulo. O termo “*Kicker*”, no âmbito do futebol robótico, diz respeito ao dispositivo que chuta a bola em situação de remate.



Figura 7.1 Descrição geral do robot desenvolvido para liga F180

7.3.3 Sistema de Alimentação

A solução encontrada para a alimentação foi a utilização de 3 pacotes de baterias de 6 volts. Tendo sido considerado que a electrónica tem consumo reduzido face aos outros consumos, o sistema electrónico é alimentado a 5V através regulador *low-drop* a partir de um único pacote de baterias. Como as baterias são carregadas individualmente, o problema da carga dos pacotes não ser exactamente igual é corrigido a cada carga. Como o sistema de comando dos motores recebe energia directa das baterias, o conversor de energia é muito simplificado na configuração implementada.

7.3.4 Localização

A localização é feita externamente ao robot através de 2 câmaras fixas onde é possível estabelecer mapeamentos entre a posição do *pixel* na imagem e a posição do campo [4].

O servidor de visão onde ligam as 2 câmaras está baseado no Sistema Operativo Linux e a parte de decisão corre em Windows, comunicando os 2 sistemas através de UDP [16]. O diagrama de blocos do sistema é apresentado na fig. 7.3 e 2 robots reais aparecem na figura 7.4.

7.3.5 Locomoção

As rodas omni-direccionais mostradas na figura 7.2 não estão colocadas a 120° por motivos de aproveitamento de espaço. Esta colocação das rodas permite ainda movimento omni-direccional mas faz com que o robot tenha uma direcção privilegiada de movimento: o robot anda mais depressa em frente que noutras direcções. A estratégia de controlo tem de tomar este facto em consideração.

7.3.6 Actuadores

O *dribbler* projectado não se revelou tão eficiente como esperado. Em vez disso, utiliza-se uma pequena rampa com material que serve de amortecedor do choque da bola. Este apêndice serve como dispositivo de recepção de passes e ainda algum domínio sobre a bola quando o robot avança.

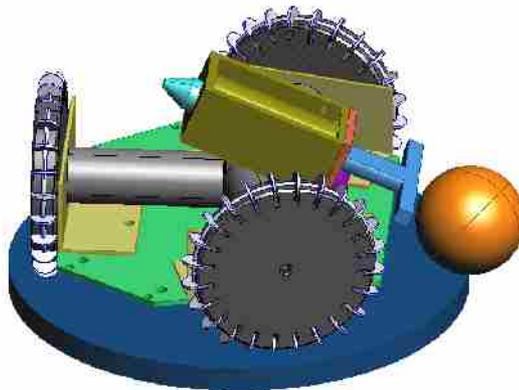


Figura 7.2 Imagem do CAD para projecto do sistema de chuto

O dispositivo de chuto é baseado num solenóide que por motivos de espaço teve que ser posicionado tal como mostra a figura 7.2. Este dispositivo de chuto utiliza energia acumulada em condensadores para um impulso com força variável sobre a bola conforme o tempo que o solenóide está alimentado.

7.3.7 Visão

O sistema de visão centralizado utiliza informação proveniente da calibração câmara mundo para situar todos os marcadores importantes. As câmaras são consideradas sensores complementares e por isso o conjunto das duas câmaras é considerado como um só elemento de medida, também graças ao facto de ser possível sincronizar a produção das imagens pelas câmaras.

O sistema de visão é uma herança que tem as suas origens em 1998 [4]. Apesar do processamento apresentado ser muito rápido e de boa qualidade, o sistema sofre do ruído de cor inerente à codificação e decodificação em PAL (ver figura 8.2, pág. 119).

As cores são identificadas através da sua colocação dentro do cubo RGB. Cada cor é definida através de uma região previamente identificadas através de treino via interface gráfica (ver figura 8.1, pág. 119).

7.3.8 Arquitectura global

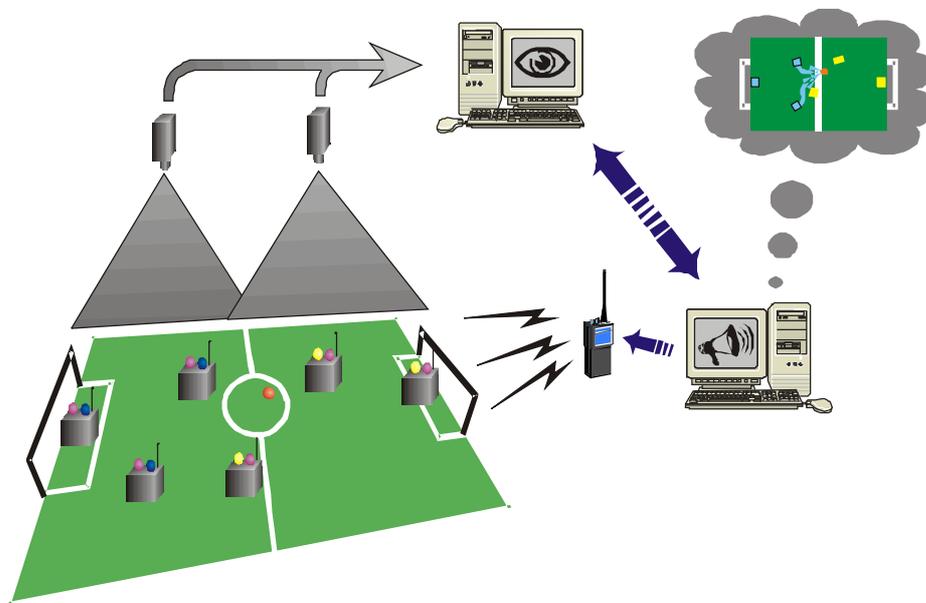


Figura 7.3 Diagrama do sistema 5dpo para a liga F180



Figura 7.4 Robots da equipa 5dpo, liga F180 (Foto tirada no RoboCup 2003, Pádua, Itália)

A nível do sistema de decisão, é utilizado um sistema a 3 níveis, similar ao apresentado para os robots médios, na secção 7.4.9, pág. 115.

O sistema robótico constituído pelo conjunto dos sistemas apresentados necessita de conhecer o estado actual do jogo para poder obedecer a um conjunto de regras estratégicas diferentes conforme o estado corrente do jogo. Algumas situações de jogo necessitam de tratamento especial tal como o comportamento durante a marcação de uma falta ou de uma grande penalidade.

7.3.9 Melhoramentos

Foram identificados os seguintes pontos a melhorar neste sistema robótico:

- Melhorar o dispositivo de domínio da bola;
- Estudar as vantagens da existência de comunicações partindo do robot (comunicar à estratégia se o robot tem a bola em contacto ou não);

7.4 Robots futebolistas F2000 (ano 2003) – plataforma experimental

7.4.1 Requisitos

O robot adequado à liga F2000 é muito diferente do adequado ao jogo na liga F180. Os robots utilizados na liga F2000 são robots com elevado nível de autonomia em que a localização deve ser conseguida sem recurso a sistemas exteriores ao campo. É possível, no entanto, utilizar de um sistema “treinador” externo, por exemplo para coordenação dos robots em campo. As restrições impostas pelas regras [181] são também muito diferentes. Os requisitos

identificados são os seguintes:

- Tamanho máximo de acordo com as regras [181]:
 - Projecção do robot no chão menor que 50 cm por 50 cm;
 - 30 a 85 cm de altura;
- Locomoção:
 - Velocidade de ponta superior a 1 m/s;
- Preferência por locomoção omni-direccional;
- Manipuladores:
 - Dispositivo de chuto da bola com tempo mínimo para “armar” o chuto;
 - Zona viável para chutar deve ser máxima;
 - Alguma capacidade de domínio da bola;
- Alimentação:
 - Puramente eléctrica;
 - Autonomia mínima de 1 hora;
- Admite-se a existência de PC embarcado;
- Sistema embebido:
 - Subsistemas de comando dos motores (implica utilização de MCUs);
 - Possibilidade de utilização de sensores locais tal como US ou IR;
 - Possibilidade de utilização de sistema de orientação (*pan*) para a câmara;
- Comunicações bidireccionais entre robot e exterior.

Um das situações graves na implementação prática é que o sistema robótico construído tem vindo a sofrer constantes actualizações que lhe conferem alguns problemas de herança de propriedades e que obrigam a um trabalho extra para conviver com uma certa diversidade de implementações.

7.4.2 Descrição geral

O robot desenvolvido para esta liga deve ter elevada capacidade de autonomia. Para tal, cada robot transporta uma câmara que roda em torno do sistema de “pescoço” (figura 7.5).

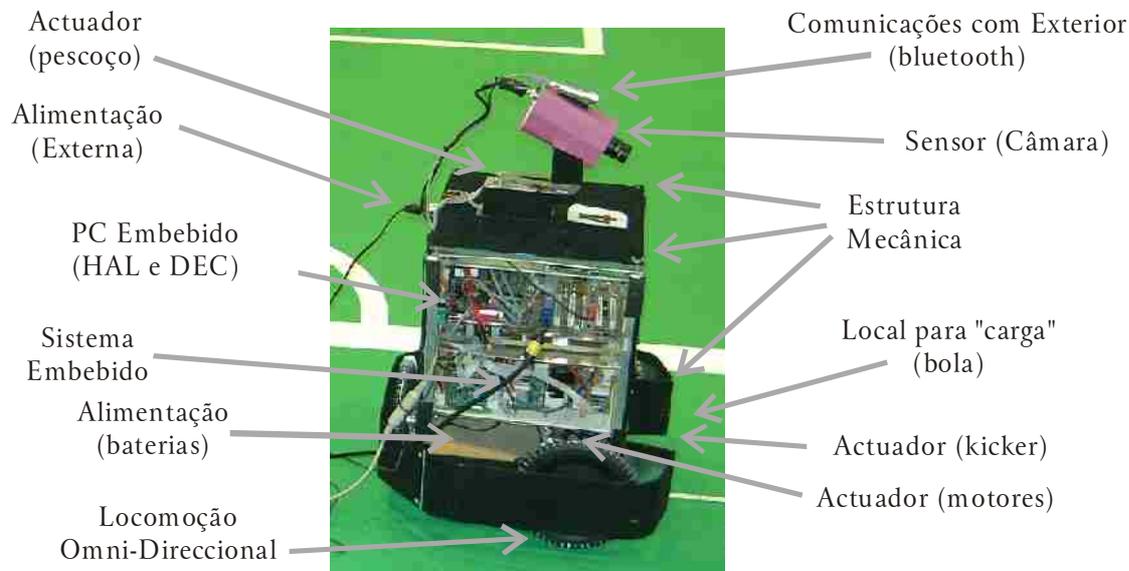


Figura 7.5 Descrição geral do robot desenvolvido para liga F2000

7.4.3 Sistema de alimentação

Em termos de sistemas de alimentação, são utilizados conversores convencionais CC/CC do tipo *Step Down* construídos pela equipa para obter as tensões positivas de alimentação necessárias ao funcionamento do PC. Utiliza-se ainda um conversor do tipo *charge pump* para gerar a tensão de -12V.

7.4.4 Locomoção

Durante o *RoboCup* 2003 a equipa 5dpo-2000 apresentou os 2 robots atacantes com modificações que permitem movimentação omni-direccional (ver figuras 7.7 e 7.8). A configuração utilizada não é ideal (3 rodas omni-direccionais colocadas perpendicularmente no robot) mas permite de facto movimentos omni-direccionais.

O robot classificado de “defesa” e o “guarda redes” utilizam locomoção diferencial pois têm, em geral, movimentação mais limitada no campo.

7.4.5 Actuadores

Para todos os robots, o subsistema de comando dos motores é baseado em MCUs e comunica com o PC do sistema através de porta série. O MCU utilizado varia conforme a geração da placa electrónica utilizada.

Está também implementado um sistema que permite a utilização de TWI para ligar várias placas de electrónica, cada uma com o seu MCU através desta rede de sensores e actuadores, assegurando assim a possibilidade de ligar mais sensores e ou actuadores ao mesmo subsistema.

Os dois robots atacantes são os únicos a terem mecanismo de chuto. Tal como nos robots F180, o dispositivo de chuto destes robots utiliza também um solenóide com núcleo móvel onde a energia provém de condensadores.

7.4.6 Comunicações com o exterior

Durante os jogos de futebol robótico, o robot comunica através de uma rede sem fios suportada por módulos *bluetooth*. O PC supervisor executa a aplicação de treinador e utiliza 4 dispositivos *bluetooth*, um por canal de comunicações com cada robot. Desta forma, a configuração do sistema é facilitada e passa a ser possível fazer difusão “simultânea” de informação (*broadcasts*), coisa que o protocolo *bluetooth* não permite. Esta situação não resulta provavelmente em colisões devido ao elevadíssimo ritmo de mudança de frequências destes sistemas (*frequency hopping*). A comunicação directa entre robots não é possível devido ao facto de a comunicação ser do tipo mestre-servidor. Tal comunicação é, no entanto, feita pelo sistema de *bridging* do sistema operativo Linux do computador que suporta o sistema de comunicações (o PC supervisor/treinador).

7.4.7 Arquitectura

A arquitectura adoptada [17] tem elementos semelhantes à arquitectura dos robots F180. Tal como mostrado na figura 7.6, também estes robots tem um servidor de visão que tenta ao mesmo tempo fornecer algum nível de abstracção do hardware (HAL) e um módulo de decisão (DEC). As aplicações HAL e DEC comunicam por UDP.

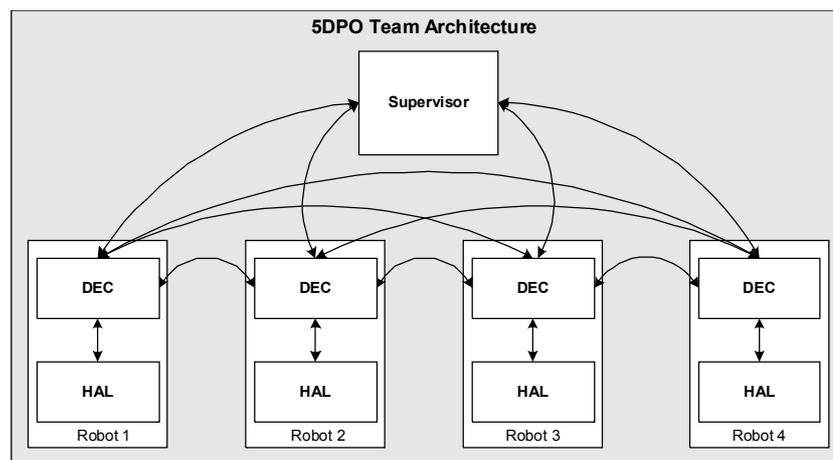


Figura 7.6 Arquitectura de Software para a equipa 5dpo-2000

7.4.8 Visão

No ano de 2003 verificaram-se grandes mudanças internas aos robots pois todo o software foi portado de Windows para Linux. Todo o sistema dentro do robot passou a caber num *flash card* (disco de memória estática) de 256MB. Num

sistema de ficheiros comprimido e apenas de leitura foi possível incluir as partes importantes do Mandrake Linux 9.1 incluindo X, KDE, algumas ferramentas de configuração gráfica e todos os protocolos necessários ao funcionamento do robot. Sobra ainda espaço para todos os ficheiros configuráveis e todos os ficheiros do utilizador possam ser mantidos num *File System* EXT3, em espaço gravável, sendo assim fácil modificar aplicações, gravar calibrações, etc..



Figura 7.7 Robot 5dpo-2000 versão modificada para movimento omni-direccional a ser configurado (foto no RoboCup 2003, Pádua, Itália)

Todos os módulos de decisão (incluindo treinador) estão reunidos dentro de um mesmo programa, isto é, todos os robots e o supervisor correm o mesmo executável de decisão mas com configurações diferentes. Mudar uma versão de programa é assim fácil pois é só substituir o executável mantendo os ficheiros de configuração antigos.

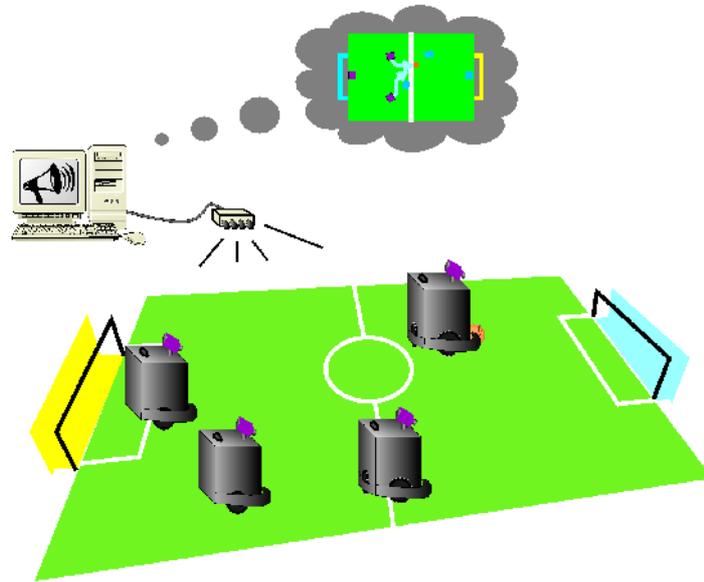


Figura 7.8 Ilustração do funcionamento da equipa 5dpo-2000 para o RoboCup 2003, Pádua, Itália

7.4.9 Decisão

O sistema implementado no robot 5dpo-2000 funciona de forma hierárquica sendo que no nível superior encontra-se o conceito de “*Role*” - papel actual do robot. Esse tal *Role* pode ser o de ser “guarda-redes” ou o “primeiro atacante”, etc.. O treinador pode permitir/impedir ou ordenar a troca de papéis entre jogadores.

Tendo um robot um dado Role, ele deve decidir a sua próxima actividade. Para a executar o robot vai necessitar de diversas “*Tasks*” - tarefas para cumprir os seus objectivos. As tarefas representam um nível intermédio de complexidade nas actividades do robot e representam conceitos tal como aproximar-se da bola e levá-la para a baliza.

Cada tarefa pode necessitar por sua vez de diversas “*Actions*” - acções elementares. Um exemplo de uma acção elementar é seguir uma linha. É neste nível que se gera as referências instantâneas de velocidade v_r e ω_r .

7.4.10 Controlo

As referências instantâneas de velocidade v_r e ω_r são as entradas de um sistema de controlo de movimento. Este sistema é baseado num PI com *feed-forward* onde a componente integral tem um limitador *anti-wind-up*.

Na configuração 2003 dos robots 5dpo-2000 existem dois níveis de actividades simultâneos: o nível do robot e o nível do pescoço.

Todos os dados trocados e mantidos entre os diversos níveis hierárquicos podem

ser registados num *log* para poderem eventualmente ser analisados mais tarde.

7.4.11 Detalhes de implementação

Os MCUs utilizados para os subsistemas dos sensores e actuadores são *Atmel* AVR 2313, 8515 e ATMega8.

A câmara utilizada é uma câmara de vigilância com saída PAL, sendo a imagem adquirida no PC através de uma placa de aquisição de vídeo baseada no integrado BT878. A câmara pode rodar em torno do eixo dos *zz* através de um suporte rotativo accionado por um motor de passo-a-passo.

O robot movimenta-se através de motores DC, actuados via *drivers* controlados por PWM produzidos num MCU dedicado.

Toda a electrónica e software foram desenvolvidos pela equipa.

O ciclo de controlo, que impõe a meta temporal a cumprir, tem um período de 40 ms e está sincronizado com a chegada de cada imagem. A comunicação serve para sincronizar as aplicações HAL e DEC.

Tal como referido anteriormente HAL e DEC comunicam por UDP e podem ou não residir na mesma máquina.

O software implementado neste trabalho é baseado em 2 *hooks* que chamam as rotinas `AJSProcessStuff` e `AJSShowStuff`. Como os nomes sugerem, estes *hooks* separaram o processamento computacional com restrições RT. Desta forma, o ciclo de controlo fecha-se mais cedo e a visualização de resultados é feita mais tarde apenas no caso de haver tempo para tal operação. No caso de falhas RT esporádicas na aplicação de Decisão, o sistema recupera rapidamente o sincronismo já que o tempo de desenho³⁰ é muito superior ao tempo de processamento. O subsistema de motorização faz parar o robot em qualquer caso de ausência prolongada de ordens (promove-se a avaria segura).

Para o presente trabalho experimental e para maior facilidade de desenvolvimento, o módulo DEC é executado num PC portátil e utiliza-se *Ethernet* a 100 Mbps para troca de informação. Todas as comunicações utilizam fios e passam por um *switch* de 100Mbps. A existência de fios não comprometem as experiências laboratoriais a realizar e beneficia-se ainda de facilidade de interligação entre todos os componentes do sistema com elevada largura de banda.

A utilização de UDP é adequada ao desempenho RT que se pretende do sistema. Se um pacote se perder, não há interesse em repeti-lo para não prejudicar as comunicações seguintes. Não há também interesse na noção de conexão, sendo no entanto necessário cuidados para a limitação do tamanho máximo do pacote de dados.

³⁰ O tempo gasto a desenhar é muito difícil de aferir uma vez que o Kylix / QT / X desenhavam efectivamente de forma assíncrona com ordens anteriores

7.4.12 Melhoramentos

Os robots 5dpo-2000 no ano de 2003 encontram-se numa situação de transição e por isso é necessária a consolidação de uma série de situações. Identificaram-se assim diversos melhoramentos genéricos:

- Transformar toda a equipa em robots omni-direccionais;
- Alargar os dispositivos de chuto a todos os robots; estudar um mecanismo específico para o guarda-redes;
- Estudar a mudança para um sistema omni-direccional de visão;
- Modelizar completamente a mecânica em CAD.

O sistema de visão é uma herança que tem as suas origens em 1998 e quase não sofre alterações desde 2000. Apesar do processamento apresentado ser muito rápido e de boa qualidade, os dados fornecidos são algo básicos e outros sistemas mais recentes apresentam maior robustez na extracção de dados e maior flexibilidade. Este sistema de visão poderia ser melhorado através de fazer com que a classificação de cor passe a ter em consideração os pixeis vizinhos do pixel em causa.

Uma importante alteração de filosofia parece ser necessária e diz respeito a que o sistema de visão deve passar a transmitir à sua saída objectos identificados em vez de manchas de cor, permitindo assim flexibilidade acrescida ao processamento de imagem.

7.5 Conclusões

Neste capítulo apresentam-se os dois mais recentes projectos em que autor participou, no âmbito do *RoboCup*.

As soluções apresentadas servem para exemplificar os conceitos dos capítulos iniciais. São justificadas as decisões tomadas e sugerem-se alguns melhoramentos.

O trabalho experimental de localização utiliza uma plataforma da liga F2000 aqui apresentada, pelo que este capítulo serve para o introduzir.

Capítulo 8

Validação Experimental

8.1 Procedimento experimental

O procedimento experimental utiliza a plataforma robótica desenvolvida para a liga F2000 do RoboCup, descrita na secção 7.4, pág. 110, na versão do robot defesa que tem tracção diferencial.

Várias técnicas de localização foram estudadas nesta dissertação. Foi estudada a localização com recurso a uma câmara exterior e várias técnicas de auto-localização. Para a fusão dos dados no interior do robot foi utilizado o EKF.

Cada método de localização foi estudado e foi modelizado cada tipo de medida presente no sistema. O EKF foi simulado e depois implantado no robot.

Para as simulações utiliza-se o programa *matlab* sob *windows* e a implementação no robot utiliza *Kylix / QT / X* sob *Linux*.

Para os testes dinâmicos utiliza-se um outro PC ligado a uma câmara numa posição elevada que corre as mesmas aplicações (HAL e DEC) parametrizados devidamente. Os dados dos testes dinâmicos de toda uma missão são guardados em forma de relatório textual na aplicação DEC do robot e depois desenhados mais tarde por um programa para *matlab*. Os dados são, no entanto, visualizados no monitor do computador que executa a aplicação DEC.

O software HAL dos robots agrupa pixels classificados com a mesma cor em manchas (*blobs*) de forma rectangular, de cor uniforme e onde o centro da mancha corresponde ao centro de massa e não ao centro geométrico do rectângulo que limita a mancha de cor [4]. Cada cor é definida como estando numa região determinada do cubo RGB. A classificação de cada pixel é independente dos pixels vizinhos. Esta região RGB pode ter qualquer forma e é definida através da interface gráfica disponível no HAL [4], gerando-se depois informação que pode ser visualizada tal como na figura 8.1.

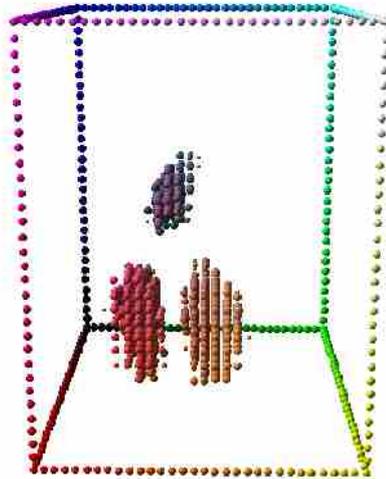


Figura 8.1 Manchas RGB para 3 conjuntos de cores dentro do cubo RGB (cores “laranja”, “azul” e “amarelo”)

A classificação de cor conseguida através deste método [4] pode ser vista ao longo do capítulo de resultados. A classificação de cor conseguida é boa apesar do sinal de vídeo recebido conter uma grande quantidade de ruído na cor. O ruído na cor tem duas fontes principais: o facto do sinal ter sido modulado e posteriormente desmodulado em PAL e também aparentemente o facto de a câmara fazer passar a imagem através de um filtro de realce (*sharpen*).



Figura 8.2 Ampliação da imagem obtida com uma vulgar CamCorder tal como a utilizada para a localização externa

Este filtro de realce é especialmente importante para a *CamCorder* da câmara de localização externa. Na figura 8.2 apresenta-se uma porção de imagem muito ampliada onde se pode avaliar do ruído de cor existente nas linhas brancas, rodeadas por muitas cores que não existem na realidade.

8.2 Câmara Externa (VHC)

8.2.1 Introdução

Nesta secção descreve-se o projecto e implementação de um sistema de localização independente e preciso. Para obter estes objectivos, optou-se por um sistema de localização externa. Este sistema servirá futuramente para controlo dos outros algoritmos de auto-localização. A experimentação estará limitada à localização numa região de cerca de meio campo da liga F2000 – 5 m por 5 m).

Através da colocação de uma câmara externa, numa posição elevada relativamente aos objectos que se movimentam, é possível obter uma localização de elevada qualidade e que cubra a zona do campo pretendida. Para facilidade de designação chame-se a este sistema VHC – *Very High Camera*. Esta câmara alta deve determinar a pose do robot que se movimenta na imagem. Através da utilização de marcadores unívocos sem repetição angular é possível adquirir facilmente a pose do robot. Esta localização pode ser conseguida à custa de uma cor e de um código de barras radial em preto e branco. Tal procedimento é utilizado pela equipa 5dpo nos anos de 2000 em diante, (ver fig. 7.4) [16],[71]. Uma outra solução frequentemente utilizada na liga F180 do futebol robótico da Federação *RoboCup* é utilizar várias manchas de cor para identificar a posição robot e a sua direcção.

Para montar um sistema de localização externo, é geralmente mais interessante e mais fácil posicionar a câmara de forma “oblíqua” sobre a cena a visualizar (ver fig. 8.3). Esta forma de colocação tem as seguintes características:

- A colocação da câmara é facilitada;
- A porção de campo coberta pela imagem aumenta;
- Passa a existir uma distorção relativa à perspectiva inerente às distâncias muito díspares dos pontos da imagem;
- A iluminação e as cores mudam ao longo de toda a imagem pelo que a calibração de cor não é imediata.

Por facilidade de implementação optou-se por recorrer a um “chapéu” com duas cores que cobre o robot, permitindo assim determinar a pose do robot (ver fig. 8.3). A configuração apresentada suportaria múltiplos robots desde que estes fossem calibrados em cores diferentes.

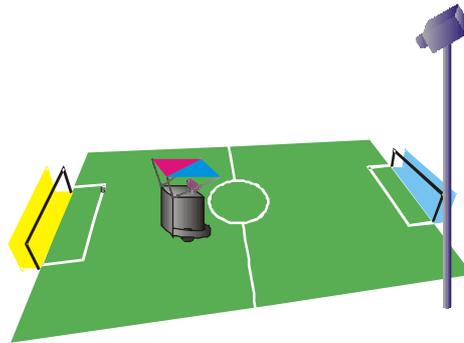


Figura 8.3 Robot sendo localizado exteriormente

8.2.2 Determinação dos parâmetros

Através do algoritmo UVZ2XY (secção 4.2.3, pág.75) é possível mapear pixels da câmara em coordenadas do mundo. Para tal, é preciso proceder à calibração da câmara e determinação dos todos os seus parâmetros.

Na figura 8.6 pode-se ver o mapeamento conseguido através da calibração dos pontos da imagem da figura 8.4 que correspondem a pontos do mundo da figura 8.5.

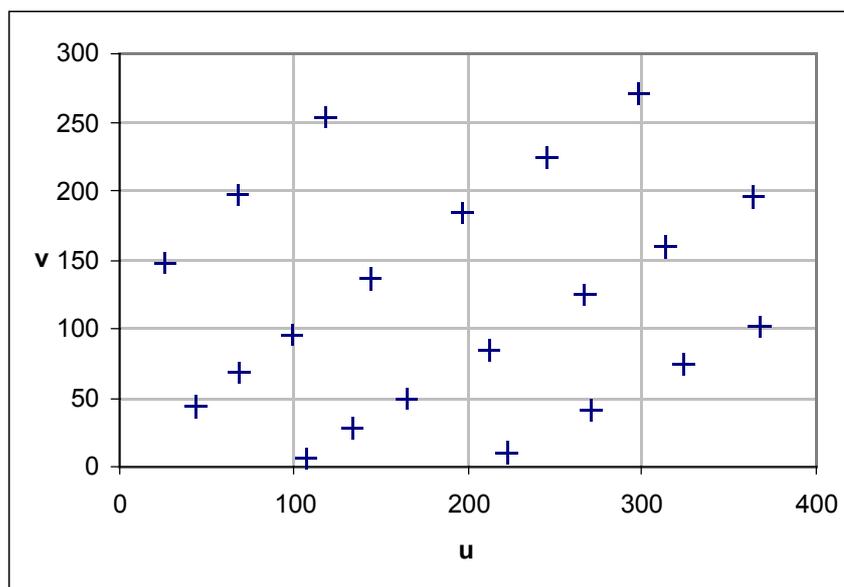


Figura 8.4 Pontos da imagem (U,V) para VHC

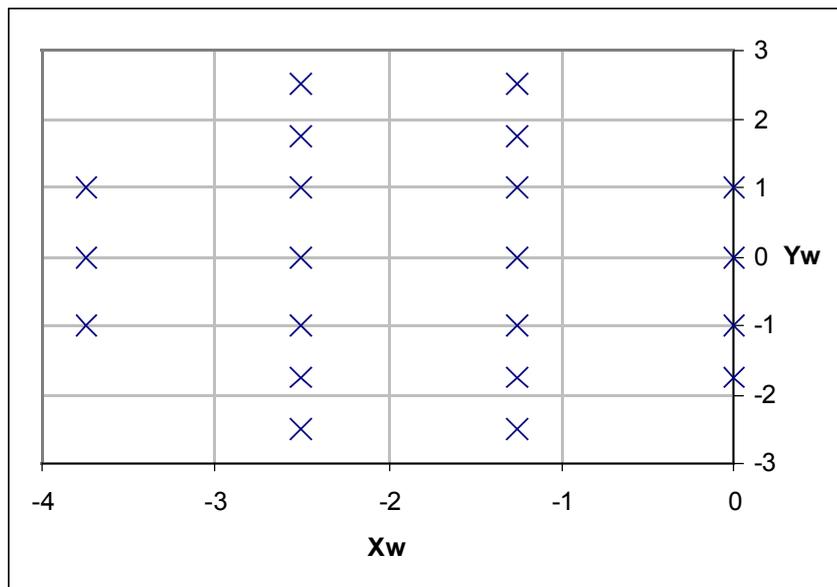


Figura 8.5 Pontos da imagem (X_w, Y_w) para VHC

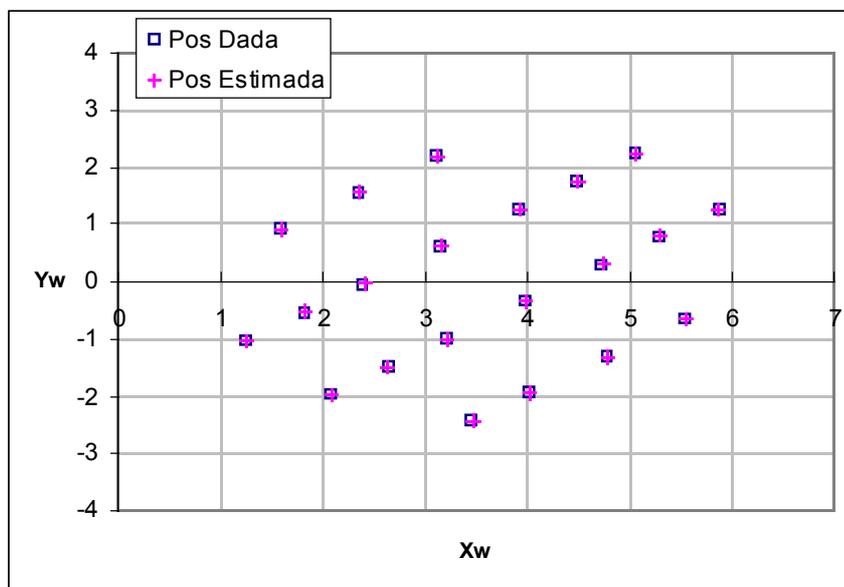


Figura 8.6 Mapeamento VHC

Na figura 8.7 pode-se ver o funcionamento do sistema. Analisando a referida figura, pode-se ver em cima à esquerda a imagem da VHC, à direita a classificação de cor conseguida, e em baixo as imagens da aplicação DEC. Em baixo, à esquerda estão as manchas de cor à entrada e à direita a localização conseguida.

Os parâmetros determinados são:

$$\begin{array}{lll}
\alpha_{VHC} = -56.92^\circ & \beta_{VHC} = -94.14^\circ & \theta_{VHC} = 49.50^\circ \\
x_{VHC} = -4.102\text{ m} & y_{VHC} = -2.788\text{ m} & z_{VHC} = 4.585\text{ m} \\
k_{b_{VHC}} = 4.8 \cdot 10^{-7} & d_{VHC} = 425.283 & \\
\Sigma = 0.338 & \Delta_{max} = 0.032\text{ m} & \\
\Delta_{\mu} = 0.008 & \Delta_{\sigma} = 0.008 &
\end{array} \tag{8.1}$$

A descrição do significado de cada parâmetro pode consultado na secção 4.2.1, pág. 71 e na secção 4.2.2, pág. 74. Σ é o valor obtido pela função C a minimizar (equação 4.14, pág. 69); Δ_{max} é o desvio máximo cometido no mapeamento; Δ_{μ} e Δ_{σ} são respectivamente a média e o desvio padrão dos erros do mapeamento; x_{VHC} , y_{VHC} e z_{VHC} são a posição da câmara no mundo.

A posição real do suporte da câmara é: $x = -4.2\text{ m}$, $y = -3.0\text{ m}$, $z = 4.6\text{ m}$.

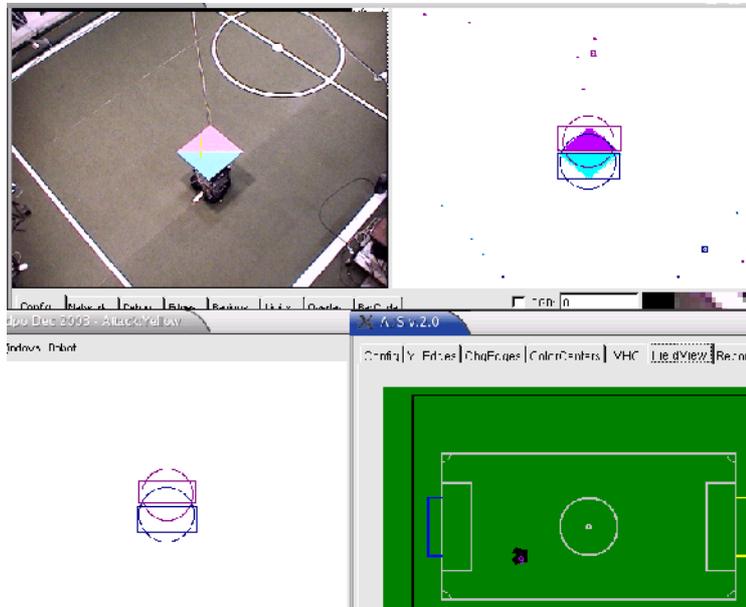


Figura 8.7 Imagens de todas as aplicações VHC – em cima imagem e classificação de cor; em baixo manchas de cor de facto transmitidas por rede e resultado visual do robot no campo; fundo da aplicação e cor não classificada representadas a branco

8.2.3 Localização Externa do Robot

Tal como se pode verificar na figura 8.7, utilizou-se um “chapéu” com as cores divididas pela diagonal (um triângulo de cada cor) em detrimento de manchas de cor de forma rectangular (figura 8.9). Tal decisão ficou a dever-se ao facto de, em testes preliminares, ter sido verificado que manchas estreitas e finas frequentemente conduzem a classificações de cor de com pequeno número de pixeis (ver figura 8.8). Tal deve-se ao facto de a imagem não ser boa em zonas próximas de duas cores em simultâneo: existe um efeito de mistura de cores que

é piorado pelo filtro de “*sharpen*” inerente à *CamCorder* utilizada (ver estes efeitos nas imagens 8.2 e 8.7). A utilização de outro tipo de marcadores de cor (por exemplo cilindros ou esferas) foi rejeitada já que se introduz incerteza adicional ao não saber com precisão a altura do objecto a medir.

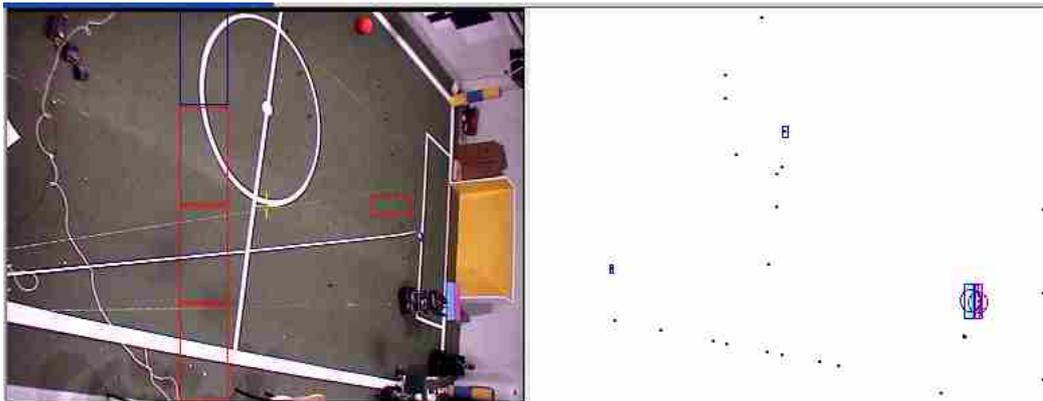


Figura 8.8 Classificação de cor em zonas estreitas resulta em poucos pixels correctamente classificados; cor não classificada representada a branco

A realização prática utiliza uma placa quadrada de alumínio de 1 m², recoberta a papel crepe de cores ciano e magenta que depois são calibradas no sistema de visão.

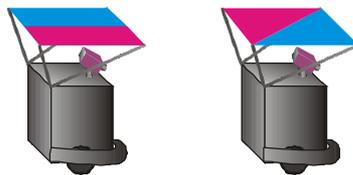


Figura 8.9 Tipos de “chapéu” para localização externa do robot

Utilizando os centros de cada mancha de cor e o algoritmo UVZ2XY, consegue-se a pose do robot:

$$\begin{aligned}
 (x_m, y_m) &= uvz2xy(u_m, v_m, z_m) \\
 (x_c, y_c) &= uvz2xy(u_c, v_c, z_c) \\
 x_r &= 0.5 \cdot (x_m + x_c) + x_0 \\
 y_r &= 0.5 \cdot (y_m + y_c) + y_0 \\
 \theta_r &= atan2(y_m - y_c, x_m - x_c) + \theta_0
 \end{aligned}
 \tag{8.2}$$

Os valores x_0, y_0 e θ_0 representam distâncias e o ângulo entre o centro do chapéu e o centro do robot³¹; z_m e z_c são as alturas de cada mancha. Estes parâmetros podem ser determinados por medição física já que são valores facilmente acessíveis. Os valores obtidos são de $x_0=0.05m$ e $y_0=0.00m$ $\theta_0 \approx \pi/4 rad$ e $z_m=z_c=0.74m$. Estes valores não carecem de uma calibração precisa já que a construção real de alumínio é sujeita a alguma torção e as irregularidades do campo provocam pequenas alterações nestes valores.

31 O centro do robot é o seu centro de rotação que está aproximadamente na mesma vertical que o ponto de focagem da câmara

8.2.4 Caracterização da medida

A análise estatística de 256 medidas do robot parado resulta nos histogramas das figuras 8.10 e 8.11. Na figura 8.10 está também o conjunto das aproximações gaussianas às curvas obtidas. Estas figuras apresentam “picos” devido à existência de quantização mas a aproximação gaussiana parece válida e natural.

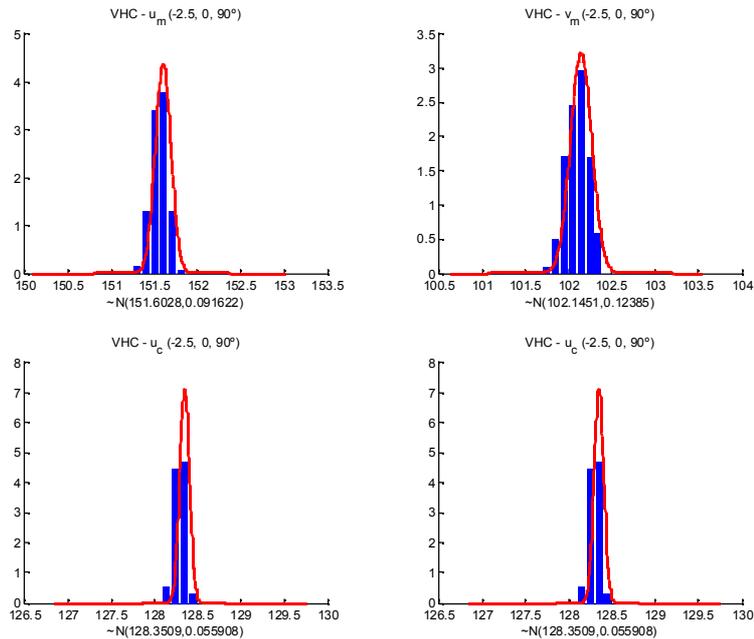


Figura 8.10

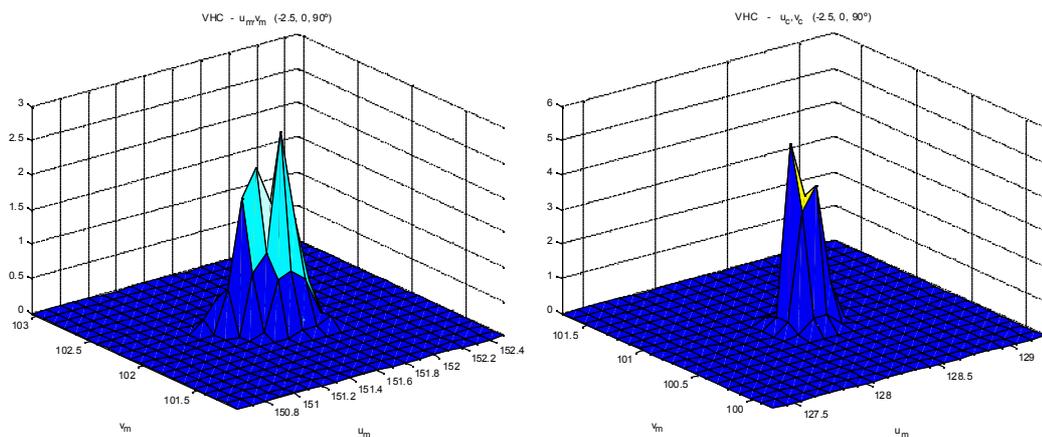


Figura 8.11 Histogramas bidimensionais para as manchas de cor referidas nas figura 8.10

Procure-se então qual o efeito deste ruído considerado gaussiano em termos de posicionamento no mundo.

Para o exemplo apresentado nas figuras 8.10 e 8.11:

$$\begin{aligned} u_m &\sim N(151.68, 0.091) & v_m &\sim N(102.14, 0.123) \\ u_c &\sim N(128.35, 0.055) & v_c &\sim N(100.83, 0.066) \end{aligned} \quad (8.3)$$

Calculando o algoritmo para cada amostra recolhida, a distribuição resultante no mundo vêm:

$$\begin{aligned} x &\sim N(-2.404, 0.002) \\ y &\sim N(0.076, 0.0008) \end{aligned} \quad (8.4)$$

Utilizando as formulas relativas às propriedades dos processos gaussianos (secção 5.2.3, pág. 81 e secção 5.2.4, pág. 81) é possível estimar analiticamente as características da distribuição depois do anterior algoritmo *UVZ2XY* que resulta nas seguintes médias:

$$\begin{aligned} \mu_{x_w} = & \frac{\cos(\theta)(z_w - z_{VHC})(d_{VHC} \cos(\alpha) - \sin(\alpha)(\cos(\beta)\mu_u + \sin(\beta)\mu_v))}{(\sin(\alpha)d_{VHC} + \cos(\alpha)(\cos(\beta)\mu_u + \sin(\beta)\mu_v))} + \\ & - \frac{\sin(\theta)(z_w - z_{VHC})(\sin(\beta)\mu_u - \cos(\beta)\mu_v)}{(\sin(\alpha)d_{VHC} + \cos(\alpha)(\cos(\beta)\mu_u + \sin(\beta)\mu_v))} + x_{VHC} \end{aligned} \quad (8.5)$$

$$\begin{aligned} \mu_{y_w} = & \frac{\sin(\theta)(z_w - z_{VHC})(d_{VHC} \cos(\alpha) - \sin(\alpha)(\cos(\beta)\mu_u + \sin(\beta)\mu_v))}{(\sin(\alpha)d_{VHC} + \cos(\alpha)(\cos(\beta)\mu_u + \sin(\beta)\mu_v))} + \\ & + \frac{\cos(\theta)(z_w - z_{VHC})(\sin(\beta)\mu_u - \cos(\beta)\mu_v)}{(\sin(\alpha)d_{VHC} + \cos(\alpha)(\cos(\beta)\mu_u + \sin(\beta)\mu_v))} + y_{VHC} \end{aligned} \quad (8.6)$$

E respectivas variâncias:

$$\begin{aligned} \sigma_{x_w}^2 = & \cos(\theta)^2 \left(\frac{(z_w - z_{VHC})^2 \sin(\alpha)^2 (\cos(\beta)^2 \sigma_u + \sin(\beta)^2 \sigma_v)}{((\sin(\alpha)d_{VHC} + \cos(\alpha)(\cos(\beta)\mu_u + \sin(\beta)\mu_v))^2} + \right. \\ & \left. + \frac{(d_{VHC} \cos(\alpha) - \sin(\alpha)(\cos(\beta)\mu_u + \sin(\beta)\mu_v))^2 \cos(\alpha)^2 (\cos(\beta)^2 \sigma_u + \sin(\beta)^2 \sigma_v)}{((\sin(\alpha)d_{VHC} + \cos(\alpha)(\cos(\beta)\mu_u + \sin(\beta)\mu_v))^4} \right) + \\ & + \sin(\theta)^2 \left(\frac{(z_w - z_{VHC})^2 (\sin(\beta)^2 \sigma_u + \cos(\beta)^2 \sigma_v)}{((\sin(\alpha)d_{VHC} + \cos(\alpha)(\cos(\beta)\mu_u + \sin(\beta)\mu_v))^2} + \right. \\ & \left. + \frac{(\sin(\beta)\mu_u - \cos(\beta)\mu_v)^2 \cos(\alpha)^2 (\cos(\beta)^2 \sigma_u + \sin(\beta)^2 \sigma_v)}{((\sin(\alpha)d_{VHC} + \cos(\alpha)(\cos(\beta)\mu_u + \sin(\beta)\mu_v))^4} \right) \end{aligned} \quad (8.7)$$

$$\begin{aligned}
\sigma_{y_w}^2 = & \sin(\theta)^2 \left(\frac{(z_w - z_{VHC})^2 \sin(\alpha)^2 (\cos(\beta)^2 \sigma_u + \sin(\beta)^2 \sigma_v)}{((\sin(\alpha) d_{vhc} + \cos(\alpha)(\cos(\beta) \mu_u + \sin(\beta) \mu_v))^2)} + \right. \\
& \left. + \frac{(d_{vhc} \cos(\alpha) - \sin(\alpha)(\cos(\beta) \mu_u + \sin(\beta) \mu_v))^2 \cos(\alpha)^2 (\cos(\beta)^2 \sigma_u + \sin(\beta)^2 \sigma_v)}{((\sin(\alpha) d_{vhc} + \cos(\alpha)(\cos(\beta) \mu_u + \sin(\beta) \mu_v))^4)} \right) + \\
& + \cos(\theta)^2 \left(\frac{(z_w - z_{VHC})^2 (\sin(\beta)^2 \sigma_u + \cos(\beta)^2 \sigma_v)}{((\sin(\alpha) d_{vhc} + \cos(\alpha)(\cos(\beta) \mu_u + \sin(\beta) \mu_v))^2)} + \right. \\
& \left. + \frac{(\sin(\beta) \mu_u - \cos(\beta) \mu_v)^2 \cos(\alpha)^2 (\cos(\beta)^2 \sigma_u + \sin(\beta)^2 \sigma_v)}{((\sin(\alpha) d_{vhc} + \cos(\alpha)(\cos(\beta) \mu_u + \sin(\beta) \mu_v))^4)} \right) \quad (8.8)
\end{aligned}$$

O formulário (8.5) a (8.8) permite considerar que o erro de localização após a transformação é gaussiano apenas em aproximação mas permite algum estudo. A partir dos dados enviados por rede para a aplicação de decisão, é ainda possível mais uma classificação estatística dos dados, apresentada na figura 8.12. Para esta experiência, confirma-se a proximidade dos resultados experimentais e do cálculo analítico, uma vez que os resultados de ambos os métodos são muito idênticos. Na verdade e para *todos* os dados experimentais recolhidos e na colocação real da VHC, os desvios máximos entre os dados recolhidos e os calculados analiticamente para a posição do robot são inferiores a 1 mm na média e o desvio padrão é inferior a 0.1 mm. Assume-se assim válido o estudo analítico.

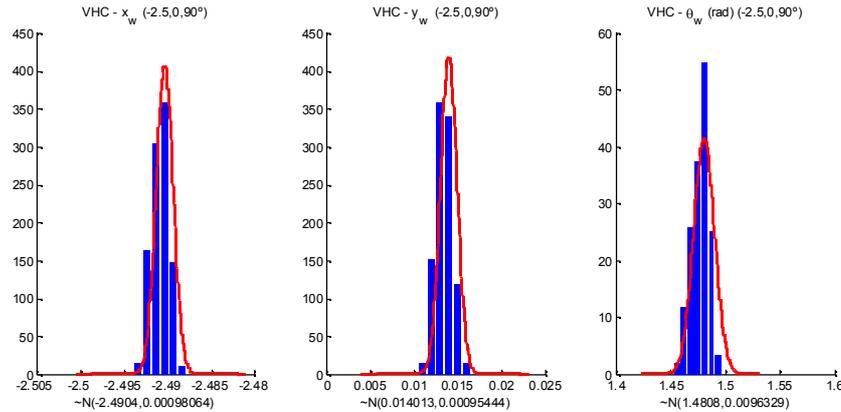


Figura 8.12 Histogramas dos dados enviados após localização por VHC

Refira-se que a qualidade dos dados experimentais obtidos beneficia do facto de os centros das manchas de cor terem resolução sub-pixel. São, no entanto, dependentes da classificação de cor anteriormente treinada que pode ser sujeita a perdas de precisão importantes por exemplo por variação não prevista das condições de iluminação.

8.2.4.1 Estudo Analítico do Erro em Função da Distância à Câmara (VHC)

Produziu-se então analiticamente a informação da figura 8.13 que mostra o desvio padrão na medida para o eixo dos xx e para o eixo dos yy no mundo. Os

pontos desta figura são separados de 30 pixels em cada direcção u e v , para toda a extensão da imagem, o desvio padrão das distribuições de u e v foram tomados como sendo unitários (um pixel). Esta imagem permite concluir que o desvio padrão da actual localização por VHC pode ser modelizada por uma constante. Como o seu valor máximo na referida figura é 0.013 m, pode-se considerar que, tomando à entrada uma distribuição gaussiana normalizada nas machas de cor (em ambas as direcções), obtém-se um erro de localização com forma aproximadamente gaussiana e com dispersão inferior a 2 cm.

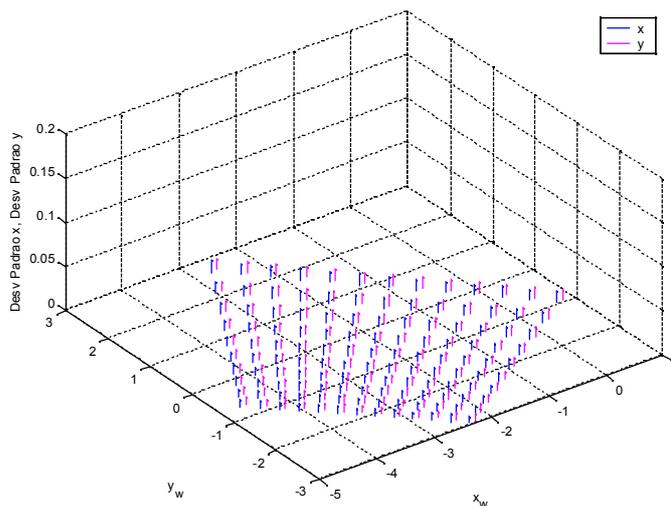


Figura 8.13 Desvio padrão da localização VHC (todas as grandezas em metros)

Para um outro posicionamento que abranja uma maior zona do campo, pode-se ver na figura 8.14 que o desvio padrão sobe à medida que a distorção da projecção aumenta em conjugação com a distância da câmara ao chão.

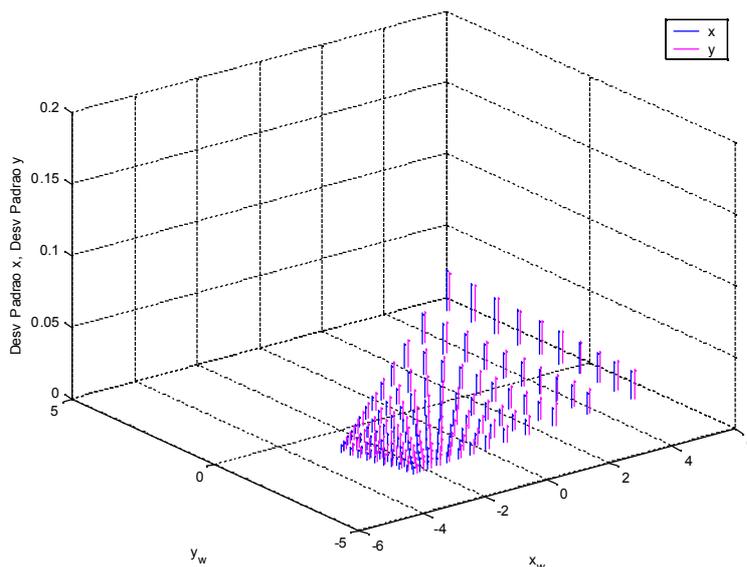


Figura 8.14 Desvio padrão para outros valores de localização VHC (todas as grandezas em metros)

A figura 8.15 mostra o gráfico experimental da variação do desvio padrão com a

distância à VHC. Verifica-se que os valores obtidos são muito baixos, abaixo dos 2 mm em todos os casos.

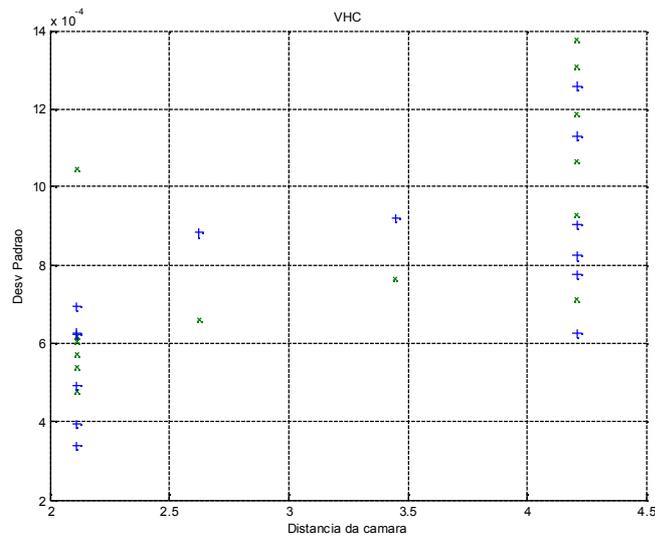


Figura 8.15 Gráfico do desvio padrão dos resultados experimentais a diferentes distâncias à câmara

8.2.4.2 Medição Experimental do Erro de Posicionamento

Pela análise da tabela 8.1 pode-se verificar que o erro máximo obtido foi de 7 cm e que o erro na medida da atitude foi, no máximo inferior a 8 graus sexagesimais (cerca de 0.14 rad). Refira-se que nem todo o erro é proveniente do sistema de medida, algum erro é devido a dificuldades experimentais tal como colocar o robot exactamente no local e na orientação correcta.

Tabela 8.1 Tabela do erro médio na medida VHC face à pose real (cada experiência com 256 medidas)

x (m)	y (m)	Atitude ($^{\circ}$)	Erro x (m)	Erro y (m)	Erro Atitude ($^{\circ}$)
-2.5	0.00	90	-0.01	-0.02	4.8
-2.5	-1.00	90	0.00	-0.01	7.4
-2.5	-1.75	90	-0.03	-0.01	6.5
-2.5	-1.75	-180	-0.06	-0.03	-1.5
-2.5	-1.75	-90	-0.01	-0.02	-1.9
-2.5	-1.75	-30	-0.01	0.00	2.6
-2.5	-1.75	0	0.00	0.01	-0.4
-1.25	0.00	90	-0.02	-0.02	6.6
-1.25	0.00	120	-0.03	-0.04	1.4
-1.25	0.00	-180	-0.01	-0.07	-3.3
-1.25	0.00	-90	0.03	-0.06	-0.6

x (m)	y (m)	Atitude ($^{\circ}$)	Erro x (m)	Erro y (m)	Erro Atitude ($^{\circ}$)
-1.25	0.00	-45	0.04	-0.04	-0.3
-1.25	0.00	0.00	0.03	-0.02	0.4

8.3 Calibração da Câmara do Robot

Recorrendo ao mesmo método foi calibrada a câmara transportada pelo robot³². O “pescoço” foi centrado e mecanicamente bloqueado para certeza de que não se move durante a calibração.

Para garantir a correcção da calibração foi levantado o padrão de pontos da câmara visível nas figuras 8.16 e 8.17. Procurou-se obter uma grande variedade de pontos, cobrindo toda a imagem. A imagem que o robot vê é exemplificada na fig. 8.16.



Figura 8.16 Imagem da câmara dentro do robot durante a sua calibração (capturada através de um utilitário de desktop remoto)

O mapeamento efectivamente utilizado pelo robot para as conversões imagem mundo é o mostrado na figura 8.18 que resulta dos seguintes parâmetros:

$$\begin{aligned}
 \alpha_{cr} &= -29.7^{\circ} & \Delta x_{cr} &= 0.1 \text{ m} & \beta_{cr} &= 0 \text{ (imposto)} \\
 d_{cr} &= 315.78 & \Delta y_{cr} &= -0.003 \text{ m} & \Sigma &= 1.29 \\
 z_{cr} &= 0.546 \text{ m} & \Delta \theta_{cr} &= -0.5^{\circ} & D_{max} &= 0.25 \text{ m} \\
 k_b &= 3.74 \times 10^{-6} & & & &
 \end{aligned} \tag{8.9}$$

³² Ver por exemplo a figura 7.7 onde a câmara está envolta em cor magenta

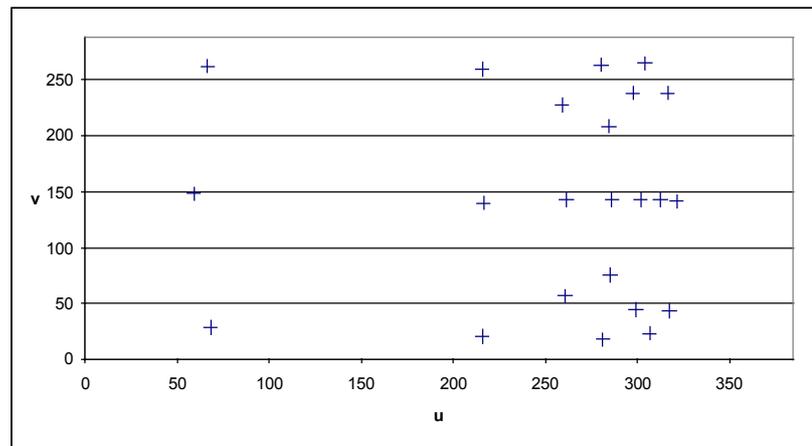


Figura 8.17 Posições na imagem (u,v) dos pontos de calibração

A descrição do significado de cada parâmetro pode ser consultada na secção 4.2.1, pág. 71 e na secção 4.2.2, pág. 74. Σ é o valor obtido pela função C a minimizar (equação 4.14, pág. 69); D_{max} é o erro máximo cometido no mapeamento. Os valores Δx_{cr} , Δy_{cr} e $\Delta \theta_{cr}$ são as diferenças entre o posicionamento da câmara e o posicionamento do robot.

O referido mapeamento foi conseguido impondo o valor do ângulo $\beta_{rc}=0$.

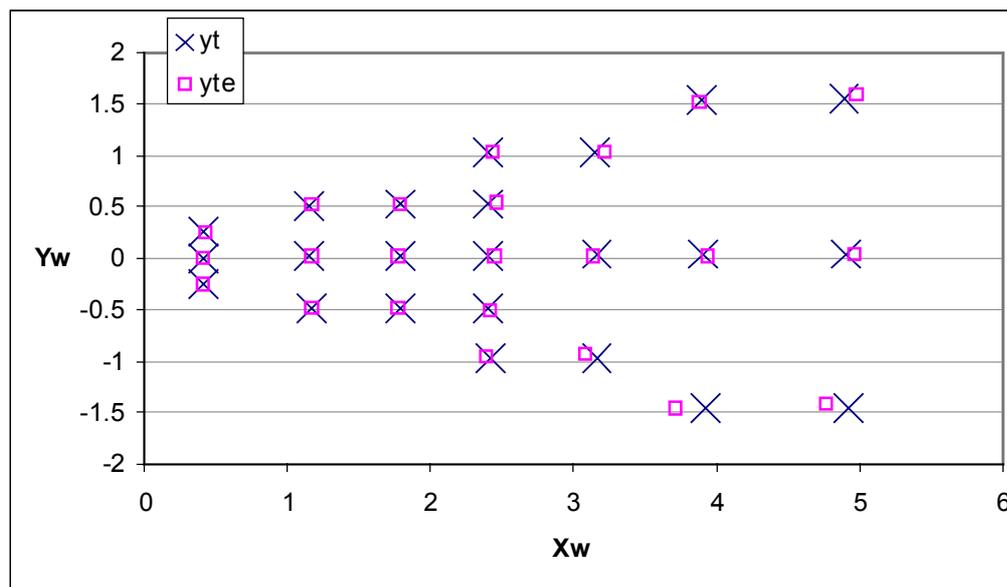


Figura 8.18 Mapeamento imagem mundo dentro do robot com os parâmetros mostrados em (8.9)

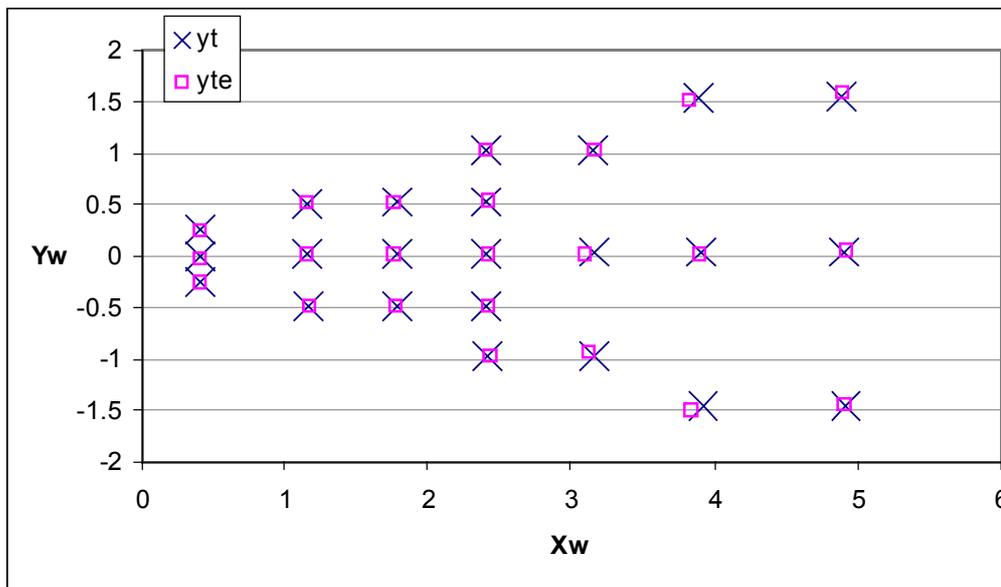


Figura 8.19 Mapeamento imagem mundo dentro do robot com os parâmetros mostrados em (8.10)

Foi também ensaiado um mapeamento em que se permitiu o ajuste de β_{cr} que resultou nos seguintes parâmetros:

$$\begin{aligned}
 \alpha_{cr} &= -29.5^\circ & \Delta x_{cr} &= 0.01 \text{ m} & \beta_{cr} &= 0.5^\circ \\
 d_{cr} &= 319.74 & \Delta y_{cr} &= -0.003 \text{ m} & \Sigma &= 0.87 \\
 h_{cr} &= 0.545 \text{ m} & \Delta \theta_{cr} &= -0.5^\circ & D_{max} &= 0.05 \text{ m} \\
 k_b &= 3.74 \times 10^{-6}
 \end{aligned}
 \tag{8.10}$$

Este mapeamento imagem mundo é ilustrado na figura 8.19.

Comparando as figuras 8.18 e 8.19 pode-se verificar que só a grandes distâncias é que se nota diferença entre os mapeamentos. Para os pontos a cerca de 5 metros de distância segundo o eixo dos x possível observar que a estimativa falha o ponto por excesso nos yy positivos enquanto que nos yy negativos falha o ponto por defeito. O mapeamento beneficia do grau de liberdade extra e a estimativa melhora um pouco na figura 8.19.

Pode observar-se, por exemplo na figura 8.26 (pág. 138) que o sistema sofre de uma importante distorção em barril³³. Tal deve-se ao facto de se utilizarem lentes de elevada abertura para que o robot tenha um campo de visão mais amplo. O factor necessário para a correcção da distorção é positivo e apresenta pois uma forma de almofada (*pin-cushion*).

³³ Ver o arco da borda da de cima da baliza amarela

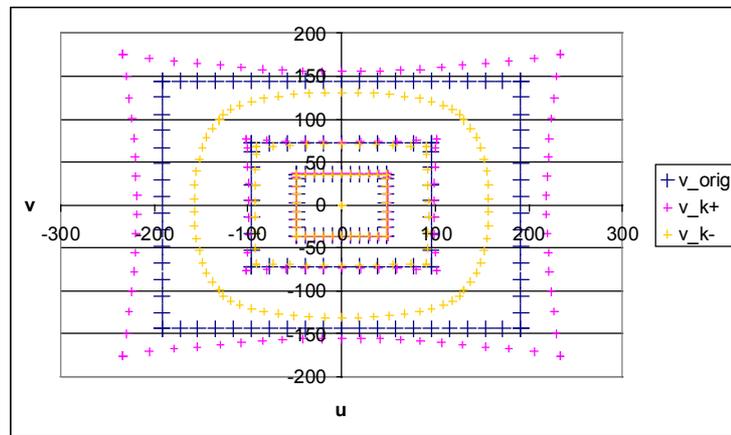


Figura 8.20 Correção do efeito de barril $k_b = \pm 3.74 \times 10^{-6}$

Outro parâmetro que será necessário é o parâmetro k_c que relaciona imagem e mundo na direcção contrária ao k_b . Este valor é calculado por optimização e é importante para relacionar pontos da imagem e pontos na realidade.

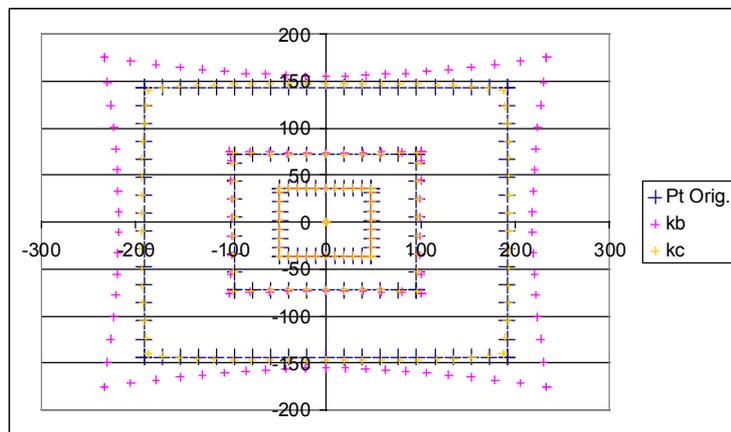


Figura 8.21 Pontos da imagem transformados de acordo com a correção do efeito de barril re-mapeados nos seus lugares iniciais através do parâmetro k_c

Os valores numéricos encontrados para a câmara do robot foram $k_b = 3.86 \times 10^{-6}$ e $k_c = -2.32 \times 10^{-6}$.

Refira-se que o valor numérico do parâmetro k_b é cerca de 10 vezes superior ao parâmetro análogo da VHC – confirma-se que a distorção é muito maior no caso do robot devido à utilização da lente grande ocular.

O sistema de localização por VHC corre uma aplicação HAL e uma aplicação DEC tal como se de um robot se tratasse, apenas as parametrizações são adequadas. Estas duas aplicações residem num mesmo PC *desktop* e lançam na rede os dados relativos à localização VHC 25 vezes por segundo, sincronizados com a chegada da imagem da câmara utilizada.

8.4 Auto-Localização por Código de Barras Unívocos

Através da utilização de Códigos de Barras Unívocos é possível desenvolver uma estratégia de localização que permite estimar todos os dados da pose do robot. Designe-se este tipo de localização como BC – *Bar Code*. A localização por BC é esquematizada na figura 8.22 e depois clarificada na figura 8.23.

O algoritmo baseia-se nos dados construtivos do Códigos de Barras Unívoco e nas medidas dos ângulos das fronteiras do código de barras para conseguir estimar distância e ângulo entre robot e este marcador.

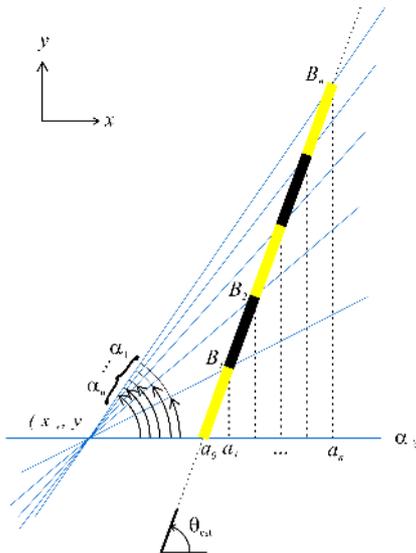


Figura 8.22 Esquema para localização por BC

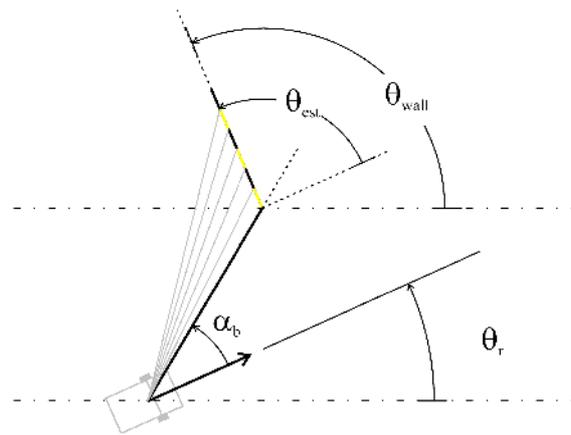


Figura 8.23 Detalhe dos ângulos envolvidos na localização por BC

Defina-se então a pose do robot nas coordenadas do mundo como $[x_r \ y_r \ \theta_r]^T$. O valor d representa a largura de cada região do código de barras e presume-se conhecido. Pretende-se encontrar o melhor $y=[a_0 \ \theta_{est}]^T$ que obedece às restrições mostradas na Figura 8.22 e 8.23. O valor a_0 representa a distância do código de barras, na direcção α_b e onde θ_{est} é o ângulo entre o código e o robot.

A definição do problema conduz a:

$$dist(B_i, B_{i+1})=d \quad \forall i \in [1, n-1] \quad (8.11)$$

onde n é o número de regiões de código de barras avistadas e d é a largura de cada bit do código de barras.

Cada ponto $B_i=[x_i \ y_i]^T$ têm as coordenadas:

$$\begin{cases} x_i = a_0 + i \cdot d \cdot \cos(\theta_{est}) \\ y_i = i \cdot d \cdot \sin(\theta_{est}) \end{cases} \quad i \in [0..n] \quad (8.12)$$

Verifica-se ainda que:

$$\frac{y_i}{x_i} = \operatorname{tg}(\alpha_i) \quad i \in [1..n] \quad (8.13)$$

Refira-se que os ângulos $\alpha_1, \alpha_2, \dots$ são medidos relativamente a α_b .

Combinando então as equações 8.12 e 8.13 obtém-se:

$$\begin{cases} y_i = \beta_i \cdot x_i & \Rightarrow & \beta_i = \operatorname{tg}(\alpha_i) \\ x_i & = & a_0 + i \cdot d \cdot \cos(\theta) \\ \beta_i \cdot x_i & = & i \cdot d \cdot \sin(\theta) \end{cases} \quad (8.14)$$

Note-se que o parâmetro β_i depende apenas dos dados do problema.

Rearranjando para uma única equação:

$$i d = \beta_i \frac{a_0}{\sin(\theta_{est})} + i d \beta_i \operatorname{cotg}(\theta_{est}) \quad (8.15)$$

A equação 8.15 confirma o dado intuitivo que se o ângulo entre robot e marcador for pequeno ($|\theta_{est}|$ for baixo) provavelmente existirão problemas numéricos com a localização. De facto, se $\theta_{est} = 0$ então o robot está alinhado com o código de barras e não é possível a localização.

Aplica-se então o método Mínimos Quadrados (secção 5.3, pág 83) à equação 8.15:

$$i d = \beta_i \hat{\theta}_0 + i d \beta_i \hat{\theta}_1, \quad i \in [0, q-1], q \geq 3 \quad (8.16)$$

Onde q é o número de ângulos em que se baseia a localização.

As matrizes dos Mínimos Quadrados ficam então:

$$\theta = \begin{bmatrix} a_0 / \sin(\theta) \\ \operatorname{cotg}(\theta) \end{bmatrix} \quad X = \begin{bmatrix} \beta_0 & 0 \\ \beta_1 & d \beta_1 \\ \vdots & \vdots \\ \beta_{q-1} & (q-1) d \beta_{q-1} \end{bmatrix} \quad Y = \begin{bmatrix} 0 \\ d \\ \vdots \\ (q-1) d \end{bmatrix} \quad (8.17)$$

A medida obtida transmite informação suficiente para uma localização absoluta do robot:

$$y^{BC*} = \begin{bmatrix} a_0 \\ \theta_{est}^{\alpha_b} \end{bmatrix} \quad (8.18)$$

Por facilidade de tratamento define-se $\theta_{est} = \theta_{est}^{\alpha_0} + \alpha_0$.

Note-se que uma estimativa deste tipo fornece apenas uma possibilidade de localização. Na verdade existe uma outra localização do outro lado do código de

barras que pode ser alcançada tomando o valor simétrico para a_0 .

Para generalização do tratamento deste e de todos os outros métodos que permitam localização completa da pose do robot, considera-se que este método devolve uma pose:

$$y^{BC} = \begin{bmatrix} x_r \\ y_r \\ \theta_r \end{bmatrix} \quad (8.19)$$

A localização do robot y^{BC} é calculada por:

$$\begin{cases} x_r = x_{wall} - a_0 \cos(\alpha_0 + \theta_{wall} - \theta_{est}) \\ y_r = y_{wall} - a_0 \sin(\alpha_0 + \theta_{wall} - \theta_{est}) \\ \theta_r = \theta_{wall} - \theta_{est} \quad , \quad \text{se } a_0 > 0 \\ \theta_r = \theta_{wall} - \theta_{est} + \pi \quad , \quad \text{senão} \end{cases} \quad (8.20)$$

Com as definições assumidas o jacobiano da medida será a matriz identidade.

$$H^{BC} = \frac{\partial(x_r, y_r, \theta_r)}{\partial(x_r, y_r, \theta_r)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8.21)$$

Toda esta informação será incluída no EKF para fusão de todas as medidas disponíveis.

8.4.1 Modelo do erro na medida

$$R^{BC} = \begin{bmatrix} \sigma_{x_r}^2 & 0 & 0 \\ 0 & \sigma_{y_r}^2 & 0 \\ 0 & 0 & \sigma_{\theta_r}^2 \end{bmatrix} \quad (8.22)$$

8.4.2 Processamento de Imagem

Para que seja possível reconhecer o código de barras, deve existir um padrão regular que acompanha o código de janela unívoca. Optou-se por juntar estes dois códigos, por baixo o padrão regular e por cima o padrão de janela unívoca, tal como mostrado nas figuras 8.25 e 8.26.

A separação dos dois códigos de barras facilitaria a determinação dos limites de

cada um deles mas traria também as seguintes desvantagens:

- Necessidade de analisar uma imagem maior (ou redução de tamanho dos códigos de barras);
- Maior esforço computacional para o processamento de imagem (mais rectas para processar e transmitir por rede, etc.)
- Maior possibilidade de haver erros indetectáveis (ocultação parcial indetectável)

Optou-se assim por juntar as duas linhas do padrão regular e do código de janela unívoca.

A implementação real utiliza simplesmente cartolinas pretas e brancas coladas à parede. Cada bit tem cerca de 25 cm de largura por 30 cm de altura. A sequência unívoca utilizada tem uma janela de 6 bits. A posição do código no mundo é exemplificada na figura 8.34, com os bits dispostos ao longo da recta $y=3.33m$. Aproximadamente 5.5 metros de código foi construído.

Para a localização por códigos de barras (unívocos), BC, é feito um processamento à imagem que produz, no final, rectas tendencialmente verticais que se ajustam às transições horizontais da cena que o robot observa.

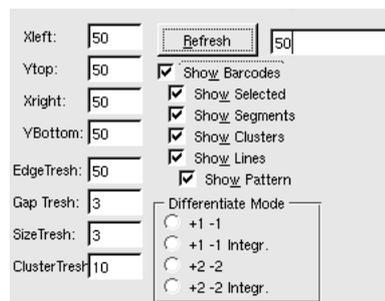


Figura 8.24 Configurações do Processamento de Imagem para localização BC

O aspecto da imagem dentro do robot é mostrado na figura 8.25. Nesta figura pode-se ver à esquerda a imagem da câmara. Envolto num fino rectângulo branco está delimitada a zona a processar para obter as rectas dos códigos de barras, zona esta que é definível na aplicação (ver figura 8.24). Os resultados do processamento são mostrados à direita ainda mesma figura 8.25. Para melhor legibilidade, as porções de interesse desta imagem foram ampliadas e rodadas tal como na figura 8.26. Nestas figuras pode ver-se o primeiro passo de processamento de imagem que consiste em tirar uma derivada na luminância sobre a vertical da imagem, considerando cada linha da imagem independente. Esta é uma técnica frequentemente utilizada em visão para obter alguma independência a variações locais de iluminação, tal como acontece com a visão humana. Calcular a derivada sobre a vertical da imagem é especialmente adequado ao desempenho da aplicação visto que a matriz da imagem tem os seus elementos contíguos sobre a vertical, tirando assim máximo proveito das

otimizações disponíveis na plataforma da processamento.



Figura 8.25 Imagem no robot à esquerda e derivada na luminância à direita (imagem capturada através de desktop remoto)

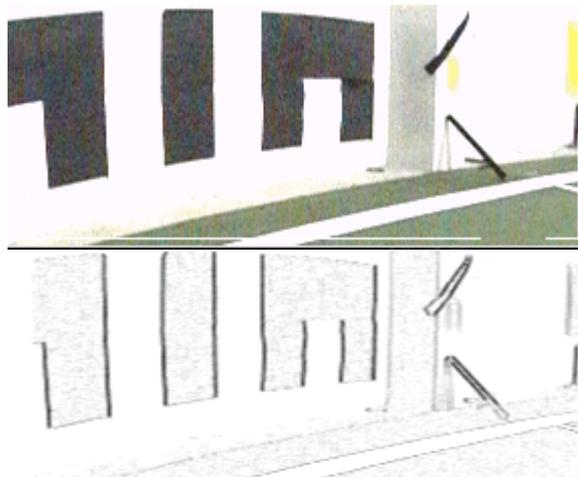


Figura 8.26 Figura 8.25 tratada, porção invertida e ampliada para melhor legibilidade (por imagem invertida entenda-se transformar cores escuras em claras e vice versa)

Os 4 casos possíveis para cálculo da derivada e todas as outras configurações para a detecção de códigos de barras podem ver vistas na figura 8.24.

Os resultados dos 4 modos de processamento para cálculo da derivada podem ser consultados na figura 8.27. Os quatro modos para cálculo da derivada³⁴ $d(u,v)$ são:

- Modo 1: $d(u,v) := \text{imagem}(u,v-1) - \text{imagem}(u,v+1)$
- Modo 2: $d(u,v) := (\text{imagem}(u,v-1) - \text{imagem}(u,v+1)) + d(u,v-1)$
- Modo 3: $d(u,v) := \text{imagem}(u,v-2) - \text{imagem}(u,v+2)$
- Modo 4: $d(u,v) := (\text{imagem}(u,v-2) - \text{imagem}(u,v+2)) + d(u,v-2)$

O modo 1 beneficia as transições rápidas; o modo 2 permite derivadas um pouco mais lentas ao acumular com a derivada anteriormente calculada; o modo 3 vai buscar pixels mais distantes e como tal a zona da derivada alarga (sobre a

³⁴Parte do algoritmo, simplificado, válido para $v > 0$; é admitido que os modos 2 e 4 resultam em derivadas com o dobro da amplitude para situações similares

horizontal); o modo 4 permite alargar ainda mais a derivada e valoriza também derivadas lentas mas progressivas ao acumular com a derivada do pixel anterior.

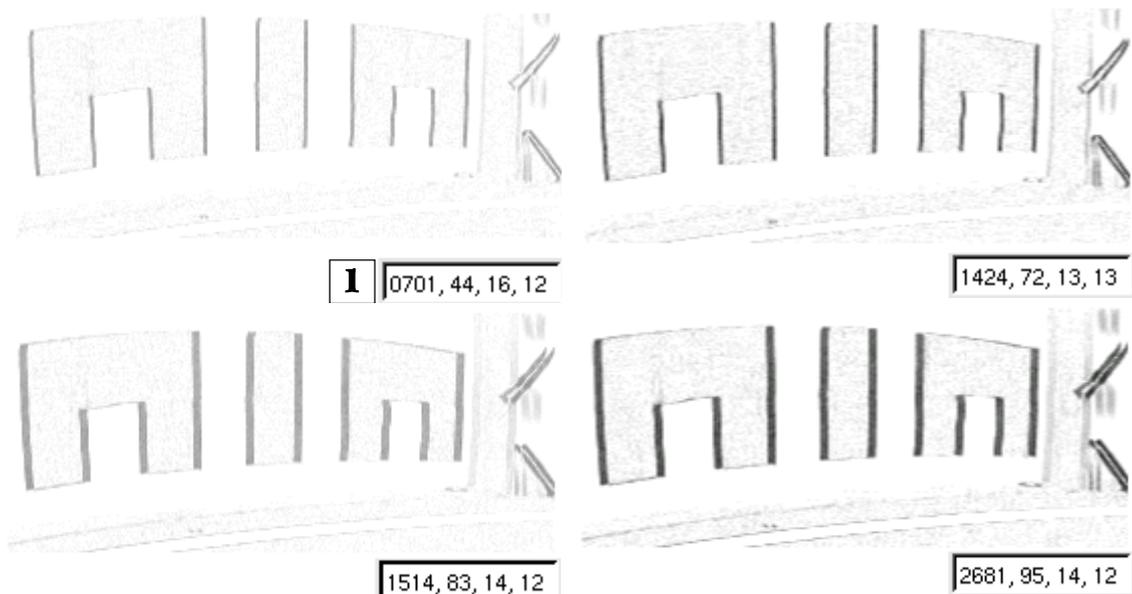


Figura 8.27 Comparação dos 4 modos de cálculo de derivadas: Modo 1 – cima esquerda ; Modo 2 – cima direita; Modo 3 – baixo esquerda; Modo 4 – baixo direita; os números dentro das caixas representam, da esquerda para a direita: número de pontos seleccionados, número de segmentos horizontais, número de regiões rectangulares e número final de segmentos

Após o cálculo da derivada na luminância ao longo das colunas da imagem, são guardados os pontos que superam um dado limiar (*threshold*). No caso da imagem da figura 8.25, são guardados 1365 pontos. Em cada uma das 4 partes da figura 8.27, o número dentro da caixa à esquerda representa o número de pixels acima do limiar. Como seria de esperar, os diferentes modos de cálculo da derivada $d(u,v)$ influenciam muito o número de pixels seleccionados. Os pontos seleccionados com limiar de valor 9 no modo 2 formam a figura 8.28.

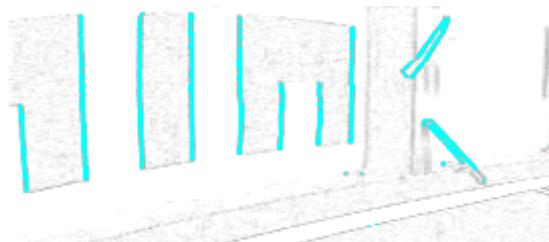


Figura 8.28 Evolução do processamento de imagem da situação da figura 8.25, pontos seleccionados por a derivada superar dado limiar

De seguida percorre-se todos os pixels seleccionados e é feito um agrupamento intermédio em segmentos obrigatoriamente horizontais³⁵. O facto de estes segmentos intermédios serem obrigatoriamente horizontais permite importantes optimizações de velocidade de execução à custa de ir guardando os pixels

³⁵ Horizontal na imagem real, tal como na figura 8.25

seleccionados em estruturas de dados por linha. Estes segmentos horizontais “perdoam” pixels em falta até um dado valor e segmentos demasiado pequenos são ainda eliminados (valores parametrizáveis – ver figura 8.24). Acompanhe-se a redução acentuada de informação a pesquisar consultando o segundo número das caixas da figura 8.27; os segmentos são também visíveis na figura 8.29. O número mínimo de pixels do segmento influencia decisivamente o ângulo máximo que os segmentos finais poderão ter.

O passo seguinte é agrupar os segmentos horizontais em regiões rectangulares. Tal processamento é mostrado na figura 8.29. Cada agrupamento é conseguido determinando todos os segmentos que têm pixels “vizinhos”. A implementação necessita apenas que fazer um conjunto de testes às extremidades dos segmentos. As regiões rectangulares podem ser vistas na figura 8.29.

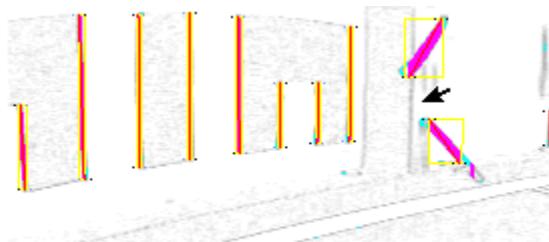


Figura 8.29 Evolução do processamento de imagem da situação da figura 8.25, mostrando segmentos e agrupamentos de pontos em zonas rectangulares

Os segmentos finais são ajustados aos cantos das regiões rectangulares coincidentes com extremidades de segmentos horizontais. Os segmentos podem ser vistos na figuras 2.25 e 2.26.

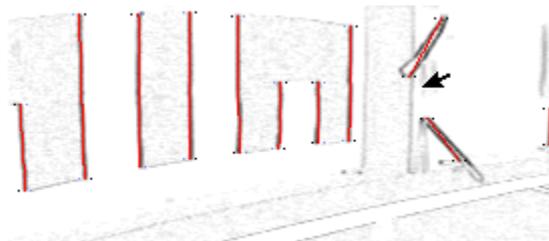


Figura 8.30 Evolução do processamento de imagem da situação da figura 8.25, ajuste de segmentos

Após a determinação dos segmentos finais é ainda necessário acrescentar a informação da cor do bit do código de barras é clara (bit a 1) ou escura (bit a 0). Esta informação é conseguida verificando a luminância à esquerda e à direita em ambas as extremidades dos segmentos. Tal é necessário devido à existência do padrão regular por baixo na imagem, independente do código de barras unívoco existente em cima. Esta informação é apresentada com um pixel claro ou escuro à esquerda e à direita de cada extremidade dos segmentos da figura 8.30 (relembre-se que esta imagem está invertida claro-escuro face à imagem real, da figura 8.25).

Os segmentos finais são depois transmitidos, por rede, da aplicação da visão para

a aplicação da decisão onde ganham o aspecto mostrado na figura 8.31. Pode-se ver também a classificação de cor (clara/escuro) transmitida à aplicação da decisão: cada segmento é desenhado com uma cor ilustrativa da classificação claro/escuro transmitida pela visão. Por exemplo o segundo segmento da esquerda para a direita na figura 8.31 apresenta bits 0 (preto) à sua esquerda em cima e em baixo; bits 1 (branco) à direita em cima e em baixo. Poderia haver bits diferentes do mesmo lado em cima e em baixo – assume-se que essa divisão seria a meio do segmento.



Figura 8.31 Segmentos recebidos pela aplicação DEC

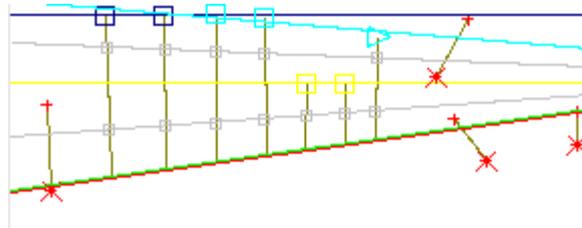


Figura 8.32 Determinação das 2 "horizontais" do código de barras

De seguida é necessário recuperar a "horizontal" do código de barras. Esta "horizontal" pode ser encontrada como sendo o ajuste de uma recta aos pontos inferiores de cada segmento efectivamente pertencentes ao código de barras (e que não são ruído).

Para marcar os ruídos como inválidos é feita uma sequência de testes:

- Teste 1: padrão de bits válido;
- Teste 2: posição do segmento dentro da zona de processamento;
- Teste 3: ajuste sobre uma linha de ângulos e diferença de ângulos;
- Teste 4: ajuste cego sobre a parede.

O Teste 1 verifica que o padrão de bits à esquerda e à direita representa uma combinação viável. Por exemplo na figura 8.31 existem 2 segmentos sobre a direita que apresentam bits a um (branco) a toda a volta (linhas puramente amarelas) o que implica que estes 2 segmentos são ruído e devem ser marcados como inválidos. A aplicação de visão marca também os segmentos em que foram detectados cores fortes (longe do cinzento no cubo RGB). A aplicação de Decisão pode decidir neste ponto se invalida esses segmentos ou não.

O Teste 2 determina segmentos perto dos limites da imagem que devem ser marcados como inválidos para prevenir segmentos parcialmente dentro da zona de processamento.

O Teste 3 monta um mapa de pontos onde $P_i=(\theta_i, \theta_i-\theta_{i-1})$ em que θ_i é o ângulo da extremidade superior do segmento (utiliza-se ainda a nomenclatura apresentada na figura 4.15, pág. 75). Para facilitar a análise, defina-se $\Theta_i=\theta_i-\theta_{i-1}, i>0$ e $\bar{\Theta}$ como sendo a média de todos os Θ_i válidos (até ao momento). Existem 4 critérios que podem invalidar um segmento:

- Critério 1: $\Theta_i - \bar{\Theta} > \Theta_{LimiarRelativoMax}$
- Critério 2: $dist(P_i, P_{i-1}) > d_{LimiarDist2Pontos}$
- Critério 3: $\Theta_i < \Theta_{LimiarAbsolutoMin}$
- Critério 4:
 - ajustar os pontos válidos à linha L_j ;
 - identificar o ponto P_i mais longe de L_j , à distância d_i ;
 - se $d_i > d_{LimiarDistLinhaPonto}$, então marcar P_i como inválido e voltar ao início do critério 4 [senão sair]

O critério 3 não é relativo ao valor médio $\bar{\Theta}$ uma vez que se verificou que nessa situação o limiar teria de ser tão baixo que deixaria de ser interessante.

O funcionamento deste Teste 3 pode ser acompanhado na figura 8.33. O ponto da esquerda é marcado com uma cruz porque já foi marcado inválido por um teste anterior (neste caso concreto o Teste 2). A recta do primeiro ajuste é mostrada a amarelo na figura 8.33; os pontos da direita são invalidados respectivamente da direita para a esquerda pelos critérios 1, 2 e 3, tal como ilustrado junto aos pontos. A ajuste final da recta é mostrado a vermelho e os segmentos válidos são representados por quadrados.

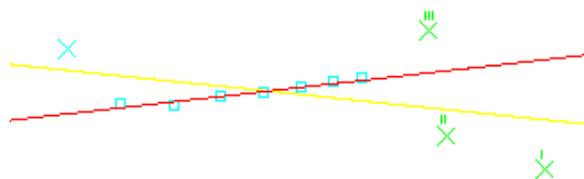


Figura 8.33 Classificação de segmentos válidos: Teste 3 – ajuste sobre uma linha de ângulos e diferença de ângulos

O Teste 4 tenta imediatamente calcular uma localização mesmo sem tentar decifrar a posição dentro do código de barras unívoco. Essa localização temporária dá uma distância e um ângulo que permitem calcular uma estimativa do tamanho de cada bit visto pela câmara através da intersecção da linha onde se situa o código de barras com todos os ângulos visualizados pela câmara. Se algum bit for demasiado estreito ou demasiado largo, então também é marcado como inválido. Observe-se a figura 8.34 onde se pode ver os resultados da visualização do Teste 4 como sendo um feixe de rectas que se unem; a localização é bem sucedida mais tarde (feixes que vão ter a uma moldura de robot); na zona do código de barras pode-se ver marcas amarelas

correspondendo a bits validados e bits magenta (maiores) correspondendo a bits invalidados.

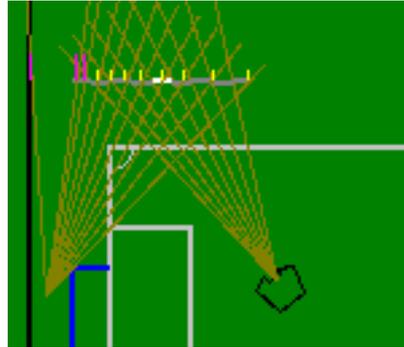


Figura 8.34 Resultados do teste do ajuste cego sobre a parede

Após cada um destes testes, é ainda feita uma verificação adicional (“limpeza”): percorrendo os segmentos do centro dos segmentos para as extremidades, ao encontrar um ponto inválido, todos os pontos daí em diante devem ser invalidados. Esta verificação é importante para impedir tirar conclusões inválidas quando existe um ruído no meio do código de barras mas permitindo ainda assim utilizar correctamente a parte da imagem sem ruídos, tolerando uma ocultação parcial.

Este sistema de processamento faz um número elevado de verificações no sentido de tentar aproveitar informação apenas se ela for segura já que é importante não ter localizações erradas – quaisquer dados à entrada do algoritmos de posicionamento dos Mínimos Quadrados produzirão uma localização pelo que é importante que os dados não contenham ruído catastrófico.

Após a determinação dos pontos válidos, é altura de calcular as 3 rectas “horizontais” que delimitam os bits das duas linhas de códigos de barras utilizados.

É ajustada uma recta à extremidade inferior dos segmentos recebidos (ver figura 8.32) utilizando a técnica dos Mínimos Quadrados recorrendo ao formulário disponível na referência [179]. A implementação mantém todos os somatórios intermédios para utilização futura. Alguns dos pontos que estiveram no primeiro ajuste são ruído e desviam a recta da “horizontal” do código de barras. De seguida entra-se num ciclo em que o ponto mais distante da recta é eliminado e novo ajuste é calculado. Esta operação é computacionalmente leve se se utilizar os somatório intermédios. Quando todos os pontos estiverem a uma distância da recta horizontal inferior a um dado limiar, considera-se que esta recta foi encontrada com sucesso. Os segmentos retirados nesta fase são marcados como “inválidos” e não participarão em computações posteriores. Existe ainda uma validação adicional que é o ângulo máximo que a recta horizontal pode fazer (na imagem).

É ainda necessário descobrir uma segunda horizontal, que passa pelo topo

superior dos códigos de barras. Para tal, faz-se passar uma recta temporária pelos topos dos 2 segmentos mais compridos (pontos assinalados por quadrados azuis na figura 8.32). Uma vez mais, dá-se início ao processo de criar somatórios intermédios que permitem retirar e acrescentar pontos de forma eficiente à recta que se vai ajustando. Os pontos mais perto desta recta que um dado limiar são acrescentados para novo ajuste (quadrados de cor ciano na figura 8.32). De novo, os pontos próximos desta nova recta são acrescentados aos somatórios (assinalado com um triângulo na figura 8.32). A implementação limita propositadamente este ciclo a duas iterações que se verificou ser suficiente para todos os casos reais testados. Os segmentos que ficaram incluídos no ajuste das rectas do topo superior são considerados “full” e os outros “half”, isto é, de tamanho completo e de meio tamanho respectivamente; os quadrados de cor amarela, no centro da imagem 8.32 são marcados como *half*.

Após estes cálculos, é feita uma média dos valores que definem as rectas para obter as rectas que passam a meio dos bits, representadas a cinzento na figura 8.35. Os quadrados cinzentos assinalam os pontos que finalmente são utilizados para calcular o conjunto de ângulos θ_i que produzem a localização por BC.

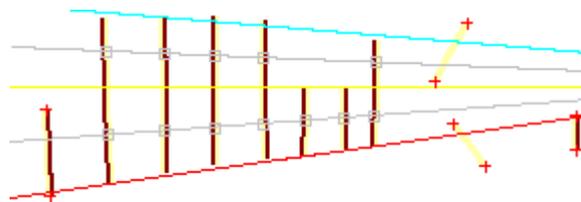


Figura 8.35 A localização BC utiliza os pontos nos centros dos segmentos relativos ao padrão regular (por baixo) e ao código unívoco (por cima)

Outra possibilidade é utilizar pontos em cima das rectas calculadas tal como se mostra na figura 8.36. Os conjunto de ângulos gerados são teoricamente os mesmos e na prática não se vislumbra diferenças significativas.

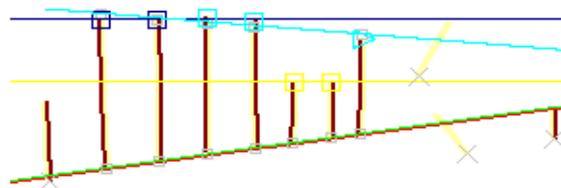


Figura 8.36 A localização BC também pode utilizar as extremidades dos segmentos ajustados para a localização

A aplicação de Decisão mostra a localização conseguida na figura 8.37 para a imagem que a aplicação de Visão processa tal como na figura 8.25.

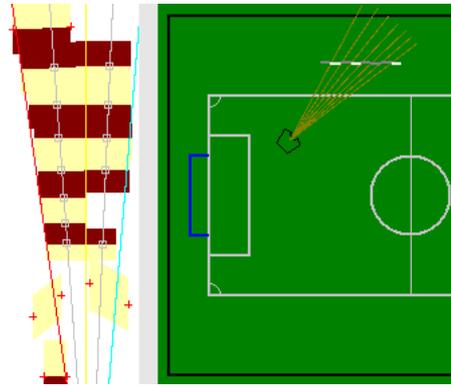


Figura 8.37 À esquerda alguns dados da aplicação de processamento de imagem, mostrando o ajuste das rectas conseguido e à direita a localização BC

Para conseguir a localização é ainda necessário fazer a leitura da sequência inscrita no código unívoco. Tal é feito através da anterior classificação de segmentos como *full* e *half*, contando quantos segmentos *half* existem entre segmentos *full*. No exemplo da figura 8.25, é possível ler a sequência 01010001, mostrada esquematicamente na figura 8.37. O posicionamento necessita de localizar esta janela dentro da sequência unívoca completa e fazer incidir os ângulos calculados sobre a posição no mundo do código de barras. A sequência unívoca completa e as suas características dimensionais e de colocação são completamente parametrizáveis no programa, estando previsto no trabalho futuro completar o código já iniciado para permitir a utilização de várias sequências unívocas espalhadas pelo ambiente do robot. Todo o processamento da localização é feito 25 vezes por segundo, de forma síncrona com a chegada da informação da imagem e com o resto do ciclo de controlo. São utilizados os segmentos enviados por rede pela aplicação HAL, não é utilizada informação adicional proveniente do estado do sistema.

8.4.3 Análise da medida

Os histogramas das figuras 8.38, 8.39, 8.40 e 8.41 mostram resultados típicos da medida de localização BC.

Em muitas situações, a medida assume uma dispersão tal como mostrado nas figuras 8.38 e 8.39, com dispersão aproximadamente gaussiana. Existem porém outras situações em que um dos segmentos do código de barras está muito perto do limiar de validação. Nessa situação e conforme o ruído, esse segmento será ou não considerado para efeitos de localização, produzindo-se nessa situação uma dispersão com dois picos de frequência tal como documentado nas figuras 8.40 e 8.41. Curiosamente, verifica-se que tomar em consideração mais um segmento para a mesma medida praticamente não altera a coordenada y do posicionamento – a diferença de orientação do robot acaba por cancelar, nessa componente, o efeito desta dupla moda.

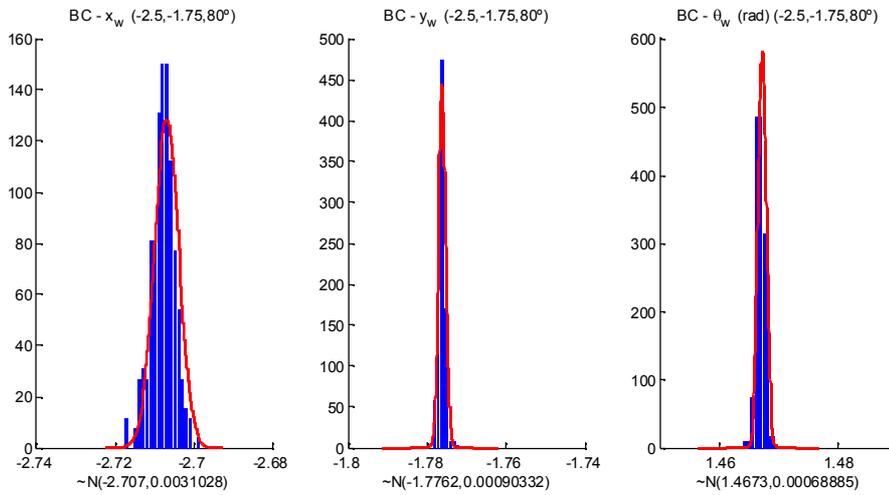


Figura 8.38 Histogramas para localização por BC (todas as distâncias em metros)

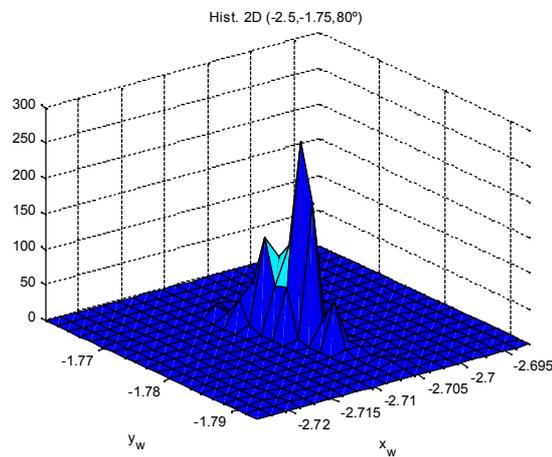


Figura 8.39 Histograma 2D da localização BC (todas as distâncias em metros)

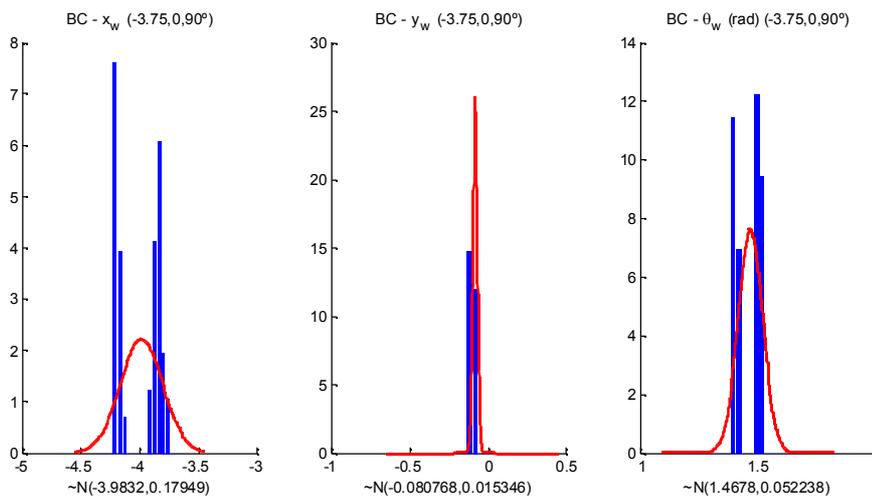


Figura 8.40 Histogramas para localização por BC na situação bi-modal (todas as distâncias em metros)

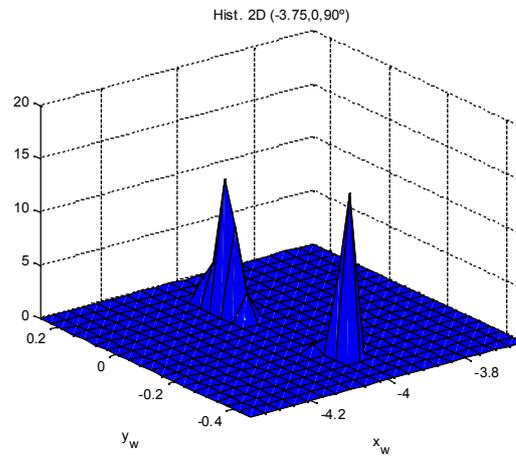


Figura 8.41 Histogramas para localização por BC na situação bi-modal (todas as distâncias em metros)

Na figura 8.42 pode-se ver a disposição no campo das medidas experimentais feitas, descritas ainda na tabela 8.2.

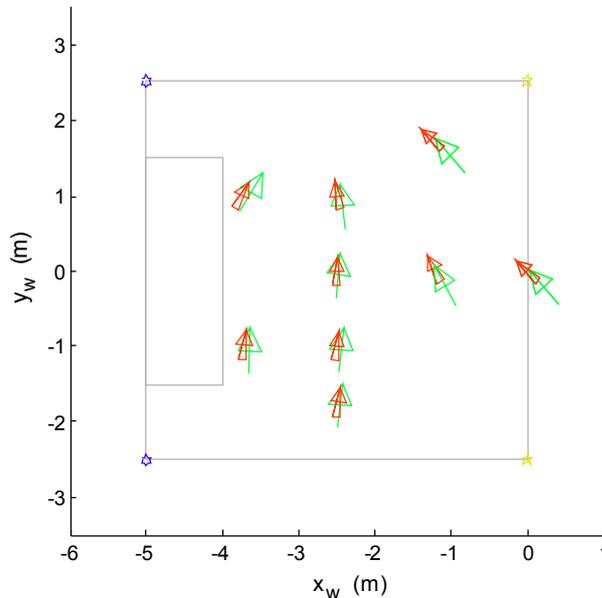


Figura 8.42 Disposição das medidas BC no campo experimental

Tabela 8.2 Resumo das experiências relativas à localização por BC

Pose			Medida VHC			Média BC			Desvio Padrão BC		
x	y	$\theta(^{\circ})$	x	y	$\theta(^{\circ})$	x	y	$\theta(^{\circ})$	x	y	$\theta(^{\circ})$
-2.5	1	100	-2.48	1.01	100.3	-2.41	0.86	96.5	0.003	0.001	0.06
0	0	135				0.21	-0.21	130.6	0.000	0.000	0.00
-1.25	0	120	-1.21	0.03	117.2	-1.08	-0.19	119.0	0.004	0.007	0.10
-1.25	0	120	-1.23	0.03	118.7	-1.08	-0.20	118.9	0.005	0.010	0.13
-2.5	-1.75	90	-2.47	-1.74	83.6	-2.41	-1.81	85.5	0.015	0.002	0.17

Pose			Medida VHC			Média BC			Desvio Padrão BC		
-2.5	0	90	-2.49	0.01	85.2	-2.45	-0.02	84.5	0.003	0.001	0.06
-2.5	-1	90	-2.50	-1.00	82.6	-2.43	-1.04	84.5	0.011	0.001	0.14
-3.75	-1	90	-3.72	-0.98	83.0	-3.65	-1.06	86.4	0.005	0.001	0.07
-3.75	1	60				-3.63	1.04	59.6	0.002	0.005	0.10

8.5 Auto-Localização por Linhas

Outra fonte de localização são as linhas avistadas. Este tipo de localização utiliza a informação relativa à recta que a câmara consegue avistar. Informação relativa à espessura da linha pode ser introduzida através de duas linhas a distâncias conhecidas. Neste momento não é feita medida nenhuma que permita retirar informação relativa ao início e fim da linha.

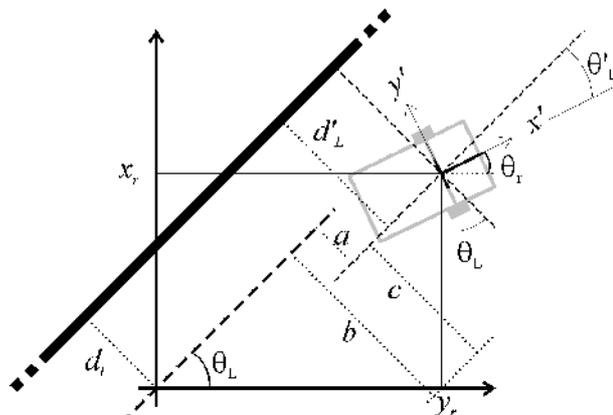


Figura 8.43 Esquema da localização por linhas

Considere-se o problema de um robot que avista uma parte de uma linha, esquematizado na fig 8.43. Perante as considerações anteriormente enunciadas, os dados de entrada para este algoritmo são apenas a distância e o ângulo entre robot e linha em coordenadas do robot, respectivamente d'_L e θ'_L . Presume-se também conhecida a posição da linha no mundo. Neste caso específico e observando a figura 8.43 pode-se concluir que:

$$\begin{cases} \theta_L = \theta'_L + \theta_R \\ d_L = d'_L - a \end{cases} \quad (8.23)$$

e também:

$$\begin{cases} b = a + c \\ b = x_R \sin(\theta_L) \\ c = y_R \cos(\theta_L) \end{cases} \quad (8.24)$$

Combinando as expressões 8.23 e 8.24 obtém-se:

$$\begin{cases} \theta_L = \theta'_L + \theta_R \\ d'_L = d'_L - x_R \sin(\theta_L) + y_R \cos(\theta_L) \end{cases} \quad (8.25)$$

As medidas são então:

$$\begin{cases} \theta'_L = \theta_L - \theta_R \\ d'_L = d'_L + x_R \sin(\theta_L) - y_R \cos(\theta_L) \end{cases} \quad (8.26)$$

É ainda necessário o cálculo do jacobiano da medida:

$$H^{Line} = \frac{\partial(d'_L, \theta'_L)}{\partial(x_r, y_r, \theta_r)} = \begin{bmatrix} 0 & 0 & -1 \\ \sin(\theta_L) & -\cos(\theta_L) & 0 \end{bmatrix} \quad (8.27)$$

8.5.1 Modelo do erro na medida

As variâncias dos erros da medida são:

$$R^{Line} = \begin{bmatrix} \sigma_{d'_L}^2 & 0 \\ 0 & \sigma_{\theta'_L}^2 \end{bmatrix} \quad (8.28)$$

Considera-se então que não existe correlação entre o erro da medida de distância e o erro na medida do ângulo. Tal afirmação é válida devido à forma como as medidas são obtidas na prática.

Devido ao facto de a localização por linhas ser considerada eminentemente local, não existem problemas relativos à utilização de funções periódicas na relação 8.26. Na implantação do EKF no robot incluiu-se código que considera limites de verosimilhança para as medidas obtidas a partir deste método o que reforça a utilidade deste método de localização ser apenas de correcção local da posição.

8.5.2 Implementação

A determinação de transições é feita ao longo de linhas parametrizáveis na aplicação HAL. Apenas estas linhas são percorridas para determinar a existência de transições de cor, isto é, para determinar um ponto em que no espaço de uma janela de alguns pixels há uma transição consistente de uma dada cor para outra dada cor. Este procedimento deve-se à necessidade imperiosa de limitar o tempo

de processamento na área da visão. Para a determinação das linhas do campo, deve-se determinar as transições envolvendo verde do campo e branco das linhas.

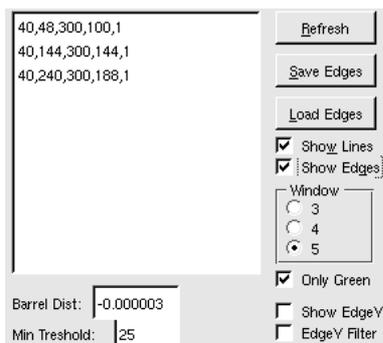


Figura 8.44 Opções relativas a transições

Na figura 8.44 pode-se ver a caixa de diálogo relativa à configuração da pesquisa de transições de cor. Na figura 8.45 pode-se ver uma porção da imagem da aplicação HAL que encontra diversas transições na imagem. Presume-se que se encontrou uma linha do campo quando as transições têm a mesma cor de origem e destino, estão em 3 ou mais linhas de pesquisa diferentes e se encontram alinhadas a menos de valores limiares adequados. Na figura 8.45 representa-se 3 linhas de pesquisa, linhas rectas finas a vermelho. A pesquisa sobre essas linhas descobre 3 linhas fronteira: uma é uma linha fronteira válida de verde para branco; outra é uma linha fronteira válida de branco para verde e existe ainda uma outra linha fronteira de verde para branco que é inválida. Esta linha inválida é constituída por uma transição pertencente ao círculo central de um campo de futebol robótico e não existe na imagem uma linha recta que ligue os 3 pontos. Esta é uma limitação importante do algoritmo implementado, que não invalida no entanto o estudo a realizar uma vez que o reconhecimento de linhas inexistente é filtrado que em testes estatísticos de verosimilhança quer na definição de parâmetros de ajuste face ao conhecimento da situação anterior.

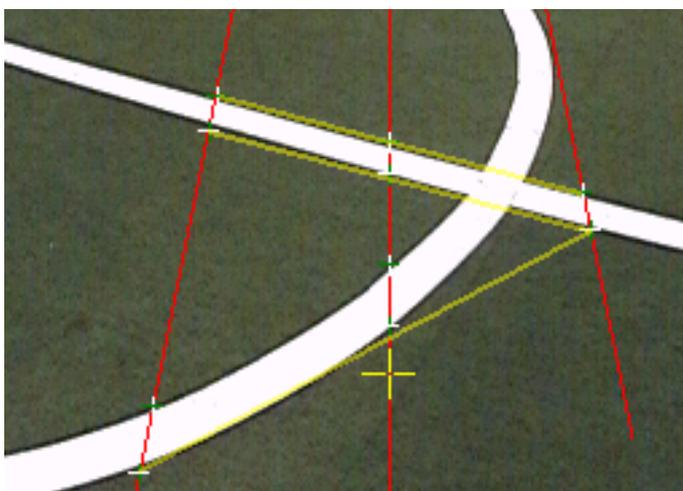


Figura 8.45 Transições entre cores detectadas ao longo das rectas e linhas fronteira detectadas (uma é incorrectamente assinalada)

Na figura 8.46 pode-se ver à esquerda a imagem obtida pela câmara do robot e à direita a classificação de cor conseguida. Esta mesma figura documenta ainda outra limitação importante do algoritmo utilizado que não encontra linhas rectas que não se cruzem com as 3 rectas de pesquisa definidas neste caso. Esta limitação diz respeito em concreto às rectas de pesquisa utilizadas já que um outro conjunto de rectas conseguiria detectar a linha recta presente nesta figura. As linhas de pesquisa devem reflectir a filosofia de pesquisa de fronteiras que se pretende implementar. É notório que as rectas de pesquisa definidas não se adequam a uma hipotética aplicação em que o robot segue alinhado com uma linha. Estas linhas de pesquisa são adequadas, no entanto, a um robot que circula dentro de uma zona delimitada por linhas. Uma possível solução seria validar apenas linhas com transições verde-branco e correspondente branco-verde determinado consistentemente. Considerou-se, no entanto, que tal limitação é excessiva e limitaria muito, na prática, o número linhas encontradas.

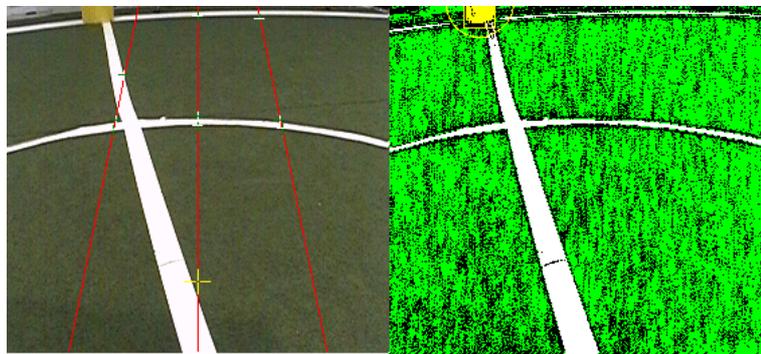


Figura 8.46 Uma linha que não é detectada

A utilização destas linhas fronteira de transição de cor não se limitam, no entanto, à detecção de linhas. Os conceitos apresentados podem ser utilizados para a determinação de fronteiras entre quaisquer cores calibradas no sistema. Uma hipotética utilização no âmbito do futebol robótico seria por exemplo determinar a linha fronteira do verde do campo para a cor do fundo da baliza. Tais dados raramente estariam acessíveis dado que existe elevado nível de ocultação de cores junto às balizas no jogos de futebol robótico. Numa outra aplicação genérica poder-se-ia utilizar diversas cores para identificar qual a linha que está a ser visualizada.

Para o funcionamento da aplicação é necessário listar todos os segmentos existentes no campo. Actualmente é desprezada a informação relativa à espessura da linha e considera-se que não há precisão suficiente para ser vantajoso listar separadamente transições verde-branco separadamente das transições branco verde. A espessura das linhas poderia servir para distinguir algumas das linhas entre si já que as regras impõem essa situação.

O reconhecimento de qual a linha avistada é feito através do conhecimento da localização anterior. Este método faz correcções locais mas a incorrecta classificação de qual a linha observada leva possivelmente à divergência do filtro

e à falha da localização. É essencial que sejam verificados os limites das linhas para aferir da verosimilhança da medida obtida. Os valores limiares utilizados devem fazer uma medida conservadora, isto é, se uma dada linha medida não se enquadrar próxima de linha alguma (que a nível de ângulo quer a nível de distância esperada à linha) a medida não deve fazer correcções ao estado do sistema.

8.5.3 Análise Resultados

Os histogramas da figura 8.47 mostram medidas que são reveladoras de um dos problemas deste tipo de medida, que é o da moda não estar centrada na medida. Uma elevada quantidade de experiências demonstra este problema. A necessária aproximação de considerar a medida gaussiana mantém-se ainda válida em aproximação uma vez que o erro é pequeno.

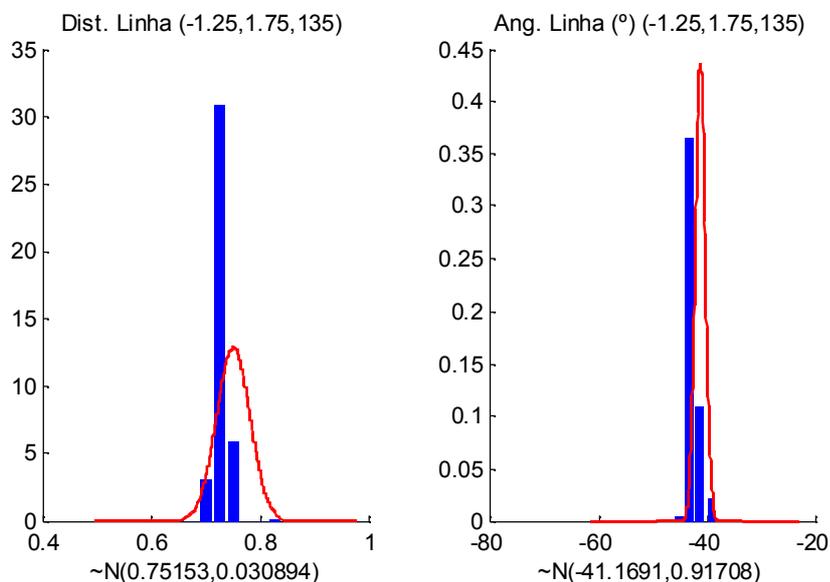


Figura 8.47 Histogramas na medida de linhas (todas as distâncias em metros)

As figuras 8.50, 8.48 e 8.49 resumem alguns dados estatísticos relativos à dispersão das medidas. Os desvios padrão mais elevados aparentam aparecer nas distâncias maiores no entanto o acompanhamento experimental levou a concluir que o ruído é muito dependente da situação particular da medida. O detalhe em concreto que afecta a dispersão da medida é a estabilidade da classificação de cor em torno da linha fronteira detectada. Tal facto é evidenciado por outras experiências muito próximas das anteriores mas que resultam em desvios padrão muito menores.

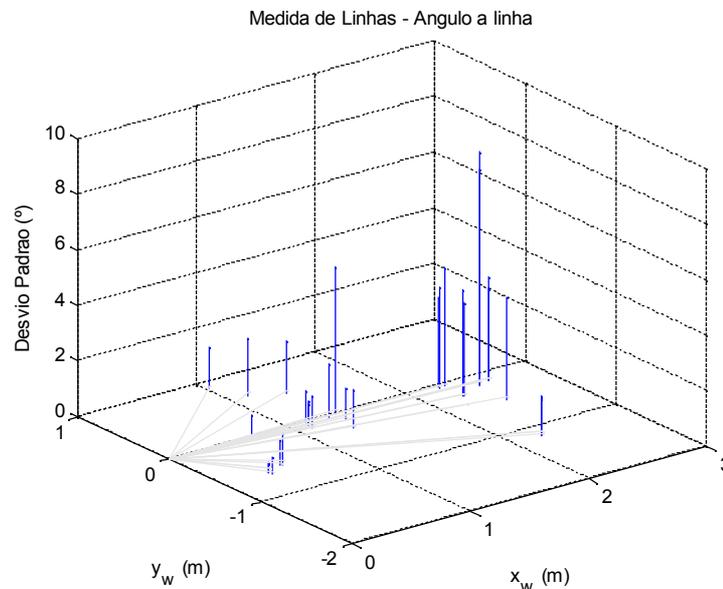


Figura 8.48 Histograma bidimensional da dispersão na medida do ângulo à linha

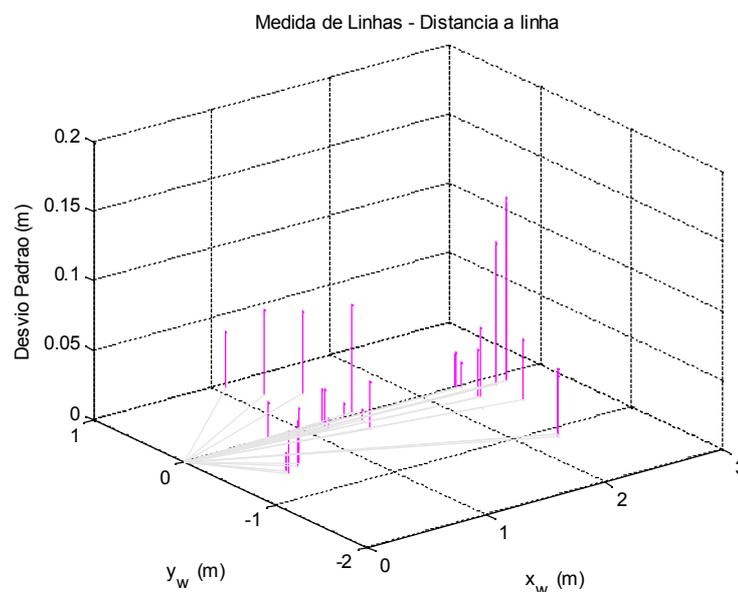


Figura 8.49 Histograma bidimensional da dispersão na medida da distância à linha

A figura 8.51 resume graficamente as medidas efectuadas. Os dados do relativos ao comportamento dos erros da medida estão podem ser vistos na tabela 8.3. Na figura 8.51 e na tabela 8.3 pode-se então comprovar diversos fenómenos descritos como por exemplo a moda ser não centrada com a média das medidas e o facto de o erro médio ser não nulo. Estas situações justificam que o valor utilizado para a covariância da media na implementação seja fixo em $(0.3\text{m})^2$, independente da distância e do ângulo da medida.

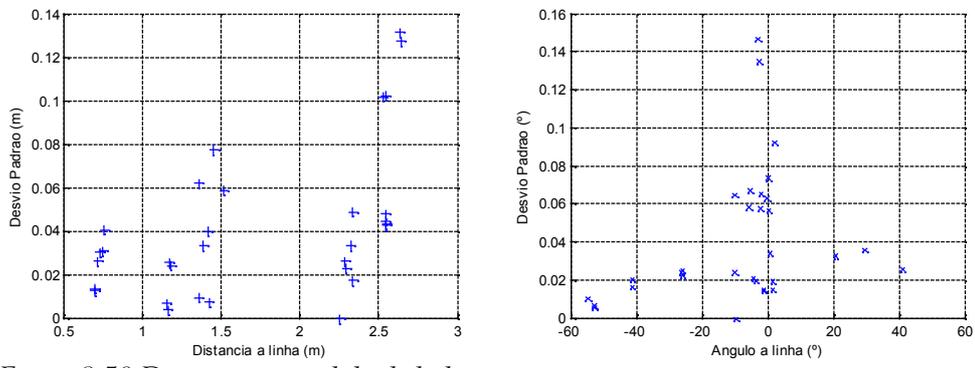


Figura 8.50 Dispersão na medida de linhas

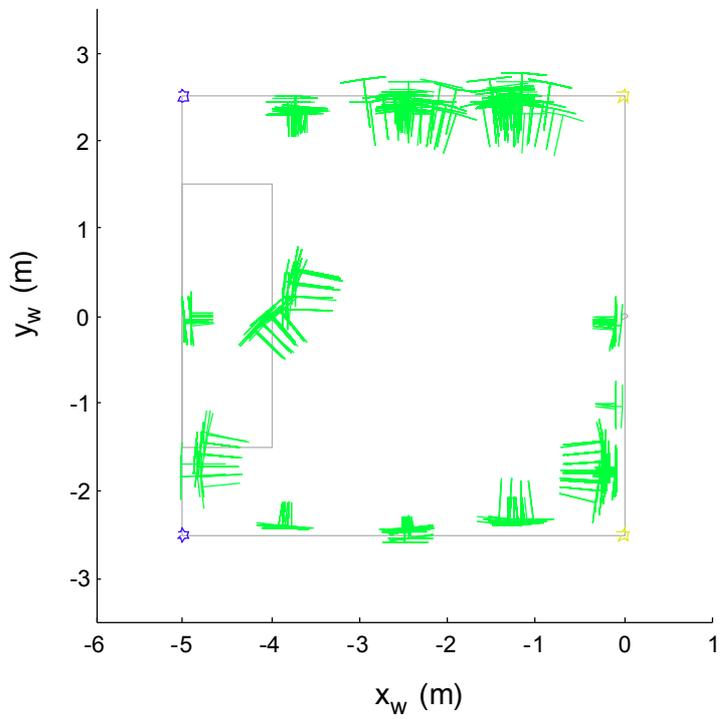


Figura 8.51 Disposição no campo das medidas de linhas

Tabela 8.3 Resumo das experiências relativas à medição de Linhas

<i>Distância à Linha (m)</i>	<i>Ângulo à Linha (°)</i>	<i>Erro (m)</i>	<i>Desvio Padrão (m)</i>
2.28	-0.117	0.21	0.030
2.55	-10.164	-0.22	0.065
2.29	0.367	0.21	0.025
2.54	-26.163	-0.03	0.041
2.55	-25.907	-0.04	0.039
2.54	-26.096	-0.03	0.045
2.63	-1.925	-0.13	0.134
2.64	-2.052	-0.13	0.128
1.18	1.583	-0.07	0.024
1.16	-1.182	0.09	0.004
1.36	0.927	-0.13	0.010
1.17	-1.072	0.08	0.026
1.39	-10.144	0.10	0.034
1.51	20.981	-0.01	0.060
1.36	29.635	0.13	0.062
1.42	-3.705	-0.07	0.009
0.73	-54.615	0.02	0.030
0.75	-41.169	0.00	0.031
0.76	-41.019	-0.00	0.041
1.45	2.198	0.05	0.072
1.15	-4.369	0.09	0.007
2.34	0.239	-0.16	0.014
0.69	-52.575	-0.05	0.012
0.69	-52.633	-0.06	0.013
0.71	1.418	-0.03	0.026
2.33	-5.510	-0.17	0.057
1.41	41.162	-0.08	0.040
2.52	-2.767	0.00	0.099
2.54	-2.402	-0.01	0.098

8.6 Localização por Postes

A utilização de postes para localização visual no RoboCup pode ser vista na liga F2000 e na liga *Sony Legged*. Este conceito tem também interesse para outros ambientes. A formulação apresentada de seguida não se restringe apenas a postes do ambiente RoboCup, antes diz respeito a qualquer tipo de marcador que indica uma direcção e poderá indicar com maior ou menor precisão a distância a que esse marcador se encontra. Uma hipotética realização simples deste tipo de marcadores pode ser simplesmente riscas coloridas pintadas numa parede.

Considere-se o problema de um robot que avista um marcador, esquematizado na figura 8.52. Considera-se que o marcador é redondo no sentido em que a informação que ele fornece não depende do ângulo entre robot e marcador³⁶.

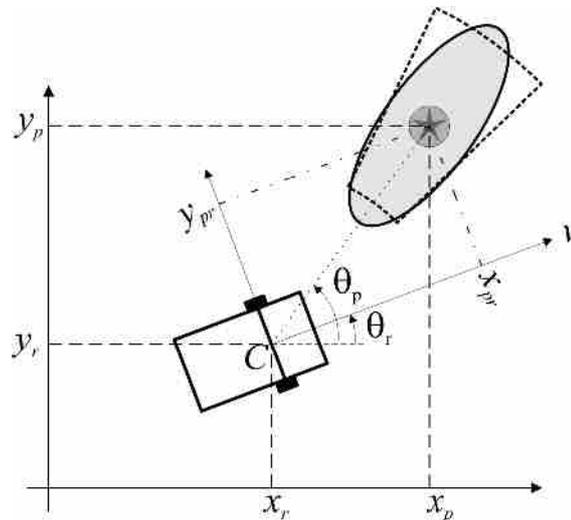


Figura 8.52 Esquema da localização por marcadores redondos/postes

Tal como referido anteriormente, a modelização de um sistema a ser estimado pelo método dos Mínimos Quadrados não deverá recorrer a funções periódicas. Por este motivo, na seguinte formulação, não aparece explicitamente o ângulo entre robot e marcador.

Tomando o esquema apresentado na fig. 8.52, pode então dizer-se que a medida de entrada deste algoritmos de auto localização é:

$$\begin{aligned}
 y^{Poste}(t_k) &= \begin{bmatrix} x_{pr}(t_k) \\ y_{pr}(t_k) \end{bmatrix} = \\
 &= \begin{bmatrix} (x_p - x_r(t_k)) \cos(\theta_r(t_k)) + (y_p - y_r(t_k)) \sin(\theta_r(t_k)) \\ -(x_p - x_r(t_k)) \sin(\theta_r(t_k)) + (y_p - y_r(t_k)) \cos(\theta_r(t_k)) \end{bmatrix} \quad (8.29)
 \end{aligned}$$

Atendendo a esta medida, é necessário calcular o jacobiano dela para que ele seja

³⁶ Um estudo deste tipo de marcadores é feito em [163] (ver página 64)

incluído na implementação no EKF. Omitindo-se a dependência do tempo de todos os elementos envolvidos:

$$H^{Poste} = \frac{\partial(x_{pr}, y_{pr})}{\partial(x_r, y_r, \theta_r)} = \begin{bmatrix} -\cos(\theta_r) & -\sin(\theta_r) & y_{pr} \\ \sin(\theta_r) & -\cos(\theta_r) & -x_{pr} \end{bmatrix} \quad (8.30)$$

Esta informação será posteriormente integrada no EKF para fusão de todos os tipos de informação disponíveis.

8.6.1 Modelo do erro na medida

O modelo do erro da medida é baseado em testes feitos na plataforma experimental. Na modelização obtida encontrou-se fortes diferenças para o erro na avaliação da distância robot-marcador e na medida do ângulo robot-marcador. Propõe-se assim que o modelo do erro na medida da distância robot-marcador seja $D_{Poste} \sigma_{Poste_d}$, onde D_{Poste} é a distância robot-marcador e σ_{Poste_d} é o desvio padrão para uma medida a 1 m. O erro associado à medida angular não depende da distância e é modelizado por uma constante σ_{Poste_a} .

$$R'' = \begin{bmatrix} (D_{Poste} \sigma_{Poste_d})^2 & 0 \\ 0 & \sigma_{Poste_a}^2 \end{bmatrix} \quad (8.31)$$

A matriz R'' não é válida no sistema de coordenadas dentro do robot. Ela é válida para o sistema de coordenadas cilíndricas com origem no centro do robot. A forma preconizada por (8.31) configura um sector de um anel e não uma elipse. O referido sector de anel é mostrado a tracejado na figura 8.52. A elipse que aproxima esta figura é:

$$R' = \begin{bmatrix} (D_{Poste} \sigma_{Poste_d})^2 & 0 \\ 0 & D_{Poste} \sigma_{Poste_a}^2 \end{bmatrix} \quad (8.32)$$

Para a utilização num EKF é necessário proceder tal como mostrado em 5.53 pág. 89 e 5.57 pág. 90 e assim é necessário obter rotação que relaciona os dois sistemas de coordenadas:

$$\theta_p = \text{tg}^{-1} \left(\frac{y_p - y_{re}}{x_p - x_{re}} \right) - \theta_{re} \quad (8.33)$$

onde (x_p, y_p) é a posição do marcador no mundo e $(x_{re}, y_{re}, \theta_{re})$ é o vector de estado estimado.

A matriz de rotação será então:

$$W = \begin{bmatrix} \cos(\theta_p) & \sin(\theta_p) \\ -\sin(\theta_p) & \cos(\theta_p) \end{bmatrix} \quad (8.34)$$

O ruído da medida fica finalmente:

$$R^{Poste}(k) = W R' W^T \quad (8.35)$$

8.6.2 Filtro de Kalman Extendido

Considere-se a figura 8.53 onde se mostra o robot em movimento.

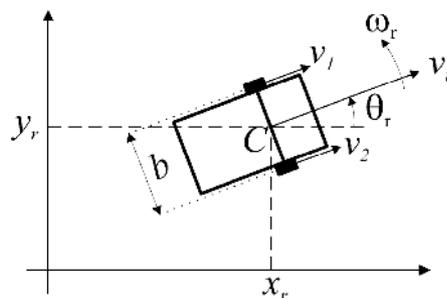


Figura 8.53 Ilustração do robot em movimento

O vector de estado escolhido para o sistema é:

$$X(t) = [x_r(t) \quad y_r(t) \quad \theta_r(t) \quad v_r(t) \quad \omega_r(t)]^T \quad (8.36)$$

onde (x_r, y_r) é a posição cartesiana do robot e θ_r é a sua direcção; v_r é a sua velocidade tangencial; considera-se tracção diferencial sem escorregamento lateral pelo que não se considera o valor da velocidade normal do robot que será sempre nulo; ω_r é a velocidade angular do robot.

O movimento é modelizado através da velocidade de cada roda, v_1 e v_2 e como tal verifica-se que:

$$\begin{cases} v_t = \frac{(v_1 + v_2)}{2} \\ \omega_r = \frac{(v_1 - v_2)}{b} \end{cases} \quad (8.37)$$

onde b é a distância entre os pontos de tracção de cada roda, valor este que é frequentemente aproximado pela distância entre rodas.

O sistema de hometria fornece valores que são convertidos directamente em velocidades de cada roda v_1 e v_2 . Como os valores de v_r e ω_r são facilmente obtidos e apresentam variações lentas pode considerar-se que estes são entradas

de comando e assim a equação da cinemática do robot no tempo contínuo é:

$$\frac{d}{dt} \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} v(t) \cos(\theta(t)) \\ v(t) \sin(\theta(t)) \\ \omega(t) \end{bmatrix} \quad (8.38)$$

No sentido de obter a equação de actualização do sistema em tempo discreto considera-se que os sinais de controlo são constantes entre cada instante de amostragem. De acordo com as equações 5.45 pág 87 e 5.62 pág 90 a dinâmica em tempo discreto do sistema é:

$$\begin{aligned} X(t_k) &= A X(t_{k-1}) + B u(t_{k-1}) \\ y(t_k) &= H x(t_k) \end{aligned} \quad (8.39)$$

E a matriz de estado na sua versão discreta fica:

$$A_k = \begin{bmatrix} 0 & 0 & -v(t_k) \sin(\theta(t_k)) \\ 0 & 0 & v(t_k) \cos(\theta(t_k)) \\ 0 & 0 & 0 \end{bmatrix} \quad (8.40)$$

$$A = \exp(A^*(k)(t(k) - t(k-1)))$$

8.6.3 Implementação

Os postes utilizados nas experiências estão de acordo com as normas propostas para a liga F2000 do RoboCup no ano de 2003. Todos os postes são cilíndricos e constituídos por 3 partes iguais, também cilíndricas mas apenas 2 cores, tal como mostrado na figura 8.54. O poste mede 20 cm de diâmetro e 1.00 m de altura, cada parte do cilindro terá 33 cm. Cada parte de poste pode ter cor amarela ou azul. Existem 2 postes com 2 partes amarelas (postes amarelos) e 2 postes com 2 partes azuis (postes azuis). Para o estudo experimental aqui apresentado, os postes amarelos foram colocados nas intersecções da linha central com as laterais. Os postes azuis foram colocados nos cantos do campo na extremidade negativa do eixo dos xx do campo (onde fica a baliza azul no campo de futebol robótico).

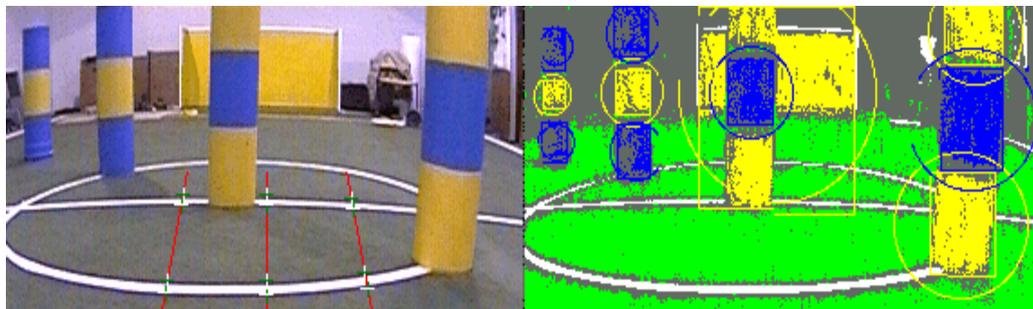


Figura 8.54 Encenação para classificação de cor relativamente a 4 postes em frente da baliza amarela (imagem rodada daquela que o robot realmente visualiza)

Na figura 8.54 pode-se ver uma situação em em que a calibração da cor verde do campo está activa. Na figura 8.55 pode-se ver outra situação em que se avista os quatro postes. Nesta figura pode-se ver da esquerda para a direita a imagem, a classificação de cor e manchas de cor transmitidas à aplicação de Decisão. É a partir destas manchas de cor que todo o processamento relativo a postes é feito. Na figura 8.54 a cor verde do campo está calibrada e aparece na imagem da direita; na figura 8.55 aparece a preto significando que esta cor não foi calibrada e logo é uma cor desconhecida – no entanto refira-se que as manchas de cor verde não são transmitidas pela rede para a decisão (a calibração da cor verde serve apenas para a detecção de transições).

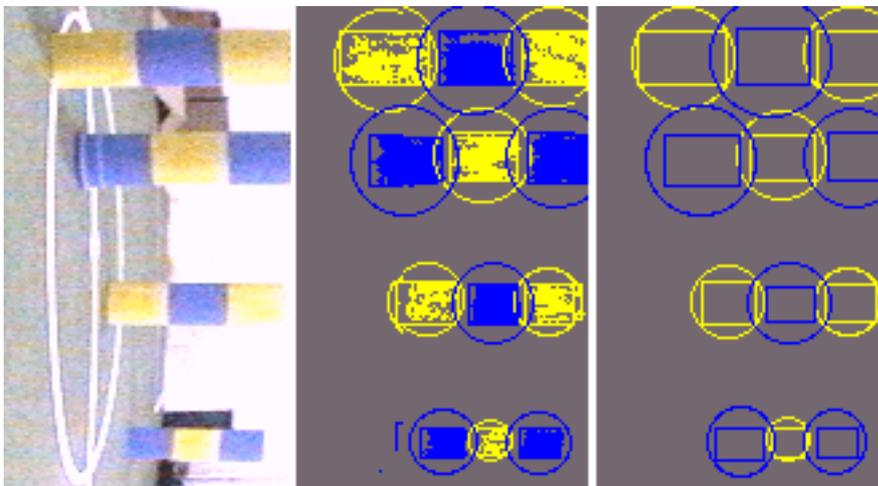


Figura 8.55 Imagem de situação encenada em que são visíveis: à esquerda a imagem com quatro postes; ao centro a classificação de cor conseguida (o verde do campo não está calibrado); à direita, os dados recebidos pela decisão

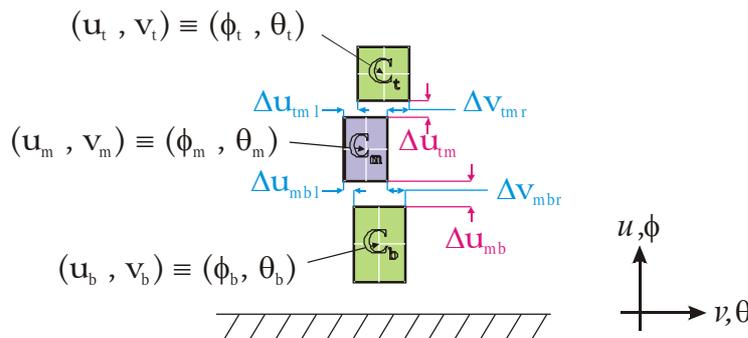


Figura 8.56 Resumo das dimensões associadas à medida de postes

Na implementação actual, todas as manchas de cor amarela são testadas para a viabilidade de pertencerem a um poste. Para tal operação defina-se C_i como a mancha de cor de ordem i que pode ter cor amarelo ou azul: $C_i = \{\text{Amarelo}, \text{Azul}\}$. A mancha C_i tem o centro nas coordenadas (u_i, v_i) a que corresponde por meio da calibração da câmara as coordenadas angulares (ϕ_i, θ_i) .

Numa primeira fase são listados todos os candidatos a postes. Cada candidato a poste é um conjunto de 3 manchas de cor $\{C_t, C_m, C_b\}$, respectivamente mancha

de cor de cima (*top*), do meio (*middle*) e de baixo (*bottom*)³⁷ que cumpre os todos requisitos listados de seguida:

1. $C_t = C_b \neq C_m$, $t \neq m \neq b$
2. $\max(\theta_t - \theta_m, \theta_t - \theta_b, \theta_m - \theta_b) < \theta_{\max}$ o parâmetro θ_{\max} define o desvio angular (horizontal) máximo para os centros das machas de cor
3. $\max(\phi_t - \phi_m, \phi_t - \phi_b, \phi_m - \phi_b) < \phi_{\max}$ separação máxima angular entre os centros das manchas de cor limitada ao valor do parâmetro ϕ_{\max} (sobre a vertical)
4. $\max(\Delta u_{tml}, \Delta u_{tmr}, \Delta u_{mbl}, \Delta u_{mbr}) < \Delta u_{\max}$ o parâmetro Δu_{\max} limita o desalinhamento máximo das bordas³⁸ sobre a vertical dos postes válidos
5. $\max(\Delta v_{tml}, \Delta v_{tmr}, \Delta v_{mbl}, \Delta v_{mbr}) < \Delta v_{\max}$ separação máxima entre machas de cor sobre a horizontal limitada ao valor máximo imposto pelo parâmetro Δv_{\max}
6. $\left\{ \begin{array}{l} d_m = UVZ2XY(u_m, v_m, h_m) \\ d_b = UVZ2XY(u_b, v_b, h_b) \end{array} \right\}$ obedecendo a $\left\{ \begin{array}{l} d_{\min} < d_m < d_{\max} \\ d_{\min} < d_b < d_{\max} \end{array} \right\}$ garantindo-se assim que postes muito longe (mais longe que d_{\max}) são invalidados; também postes que estão demasiado perto e não poderiam ser vistos pelo robot (mais perto que d_{\min}) devem ser eliminados (ver figura); h_m e h_b são as alturas a que os centros das partes dos postes devem estar na realidade, actualmente $h_m = 0.5$ m e $h_b = 0.165$ m.

No final é feita uma verificação adicional: se existirem candidatos a poste que partilhem manchas de cor, então eliminar todos os candidatos em causa. Postes que partilhem manchas de cor são fruto de ocultação, ruído ou má classificação de cor pelo que não se deve tirar conclusões em qualquer destas situações anómalas.

Observando a figura 8.57, pode-se verificar que na configuração geométrica apresentada não é possível ver postes a uma distancia inferior a d_{\min} , uma vez que a mancha de cor superior deixa de ser visível (confrontar com a figura 8.54). Um outro problema pode também acontecer que é deixar de ver a totalidade da mancha inferior o que poderia levar a alguns problemas de falta de precisão.

Tomando como referência figura 8.54 que representa 4 postes, o poste situado à frente da baliza amarela não será classificado como tal uma vez que as bordas da mancha de cor amarela são muito distantes das bordas das manchas de cor azul. Admitidamente, esta é uma limitação que advém da estratégia utilizada, que, sendo parametrizável, é facilmente ultrapassável. Esta regra faz sentido na aplicação experimental e na situação do campo de jogo do futebol robótico pois

³⁷ Os termos utilizados **dizem** respeito à realidade (figura 8.54): em cima, no meio e em baixo **não** dizem respeito à imagem que robot vê (figura 8.55)

³⁸ Operações relativas aos centros utilizam relações angulares; limites exteriores das manchas de cor utilizam pixels

se o robot estiver dentro do campo, um poste nunca aparece à frente de uma baliza. A estratégia da colocação dos postes nos cantos da zona de movimentação do robot parece sensata noutras realidades que transcendam a aplicação do futebol robótico.

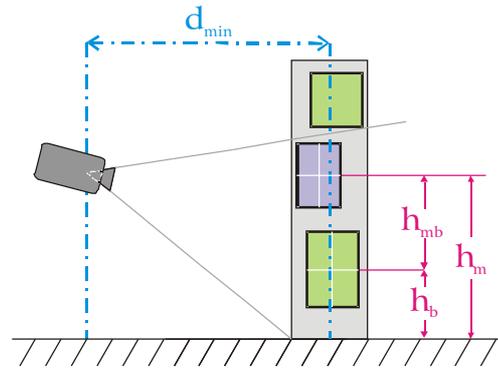


Figura 8.57 Distância mínima para um poste ser visível

A medida do poste propriamente dita é calculada da seguinte forma (ver figura 8.57):

$$\begin{cases} \theta_{Poste} = (\theta_t + \theta_m + \theta_b) / 3 \\ D_{Poste} = \frac{h_{mb} \cdot d}{u_m - u_b} \end{cases} \quad (8.41)$$

O cálculo de D_{Poste} é uma fórmula fechada que advém do modelo *pin-hole* da câmara; é utilizado o parâmetro intrínseco d da câmara (definido na secção 4.2.1, página 71) para calcular a distância ao alvo a partir de uma diferença de pixels dos dois centros das manchas de cor do meio e de baixo. Os centros das manchas de cor são a fonte segura de informação relativa a este processamento pois não só têm resolução sub-pixel (média de muitos pixels) como são relativamente estáveis face a uma classificação de cor ruidosa. A utilização da diferença entre os 2 centros de cor não só fornece uma variância menor como faz este cálculo fique independente do ângulo da câmara que pode ser falseado por irregularidades no pavimento onde o robot se movimenta.

8.6.4 Análise de Resultados

Os resultados dos testes realizados são mostrados nas figuras 8.58, 8.59 e 8.60. Estes resultados permitem estimar as covariâncias associadas às medidas relativas às posições dos postes.

Na figura 8.61 pode-se ver a colocação das medidas efectuadas no campo de ensaios.

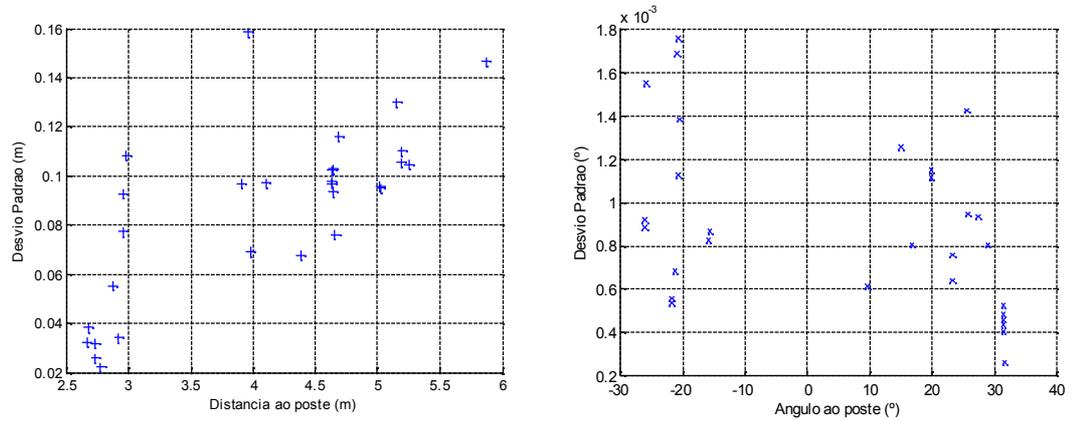


Figura 8.58 Dispersão na medida de Postes em forma de gráfico

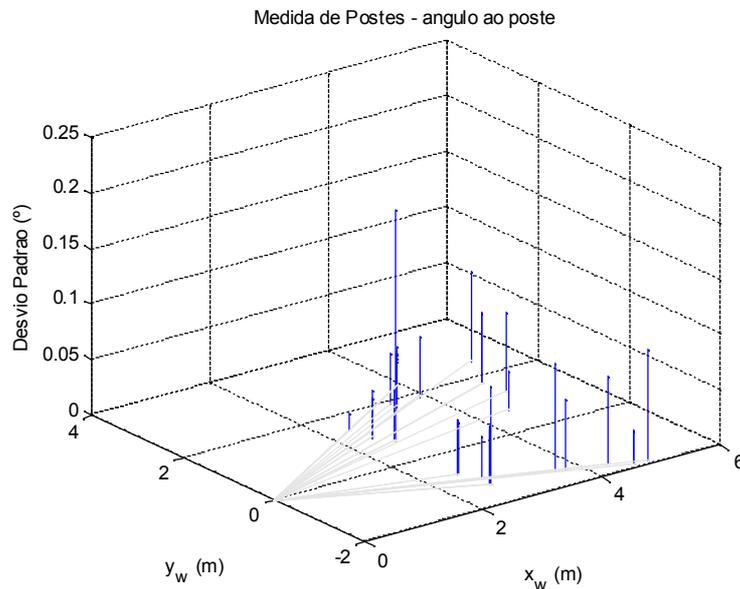


Figura 8.59 Diagrama da dispersão do ângulo na medida de postes

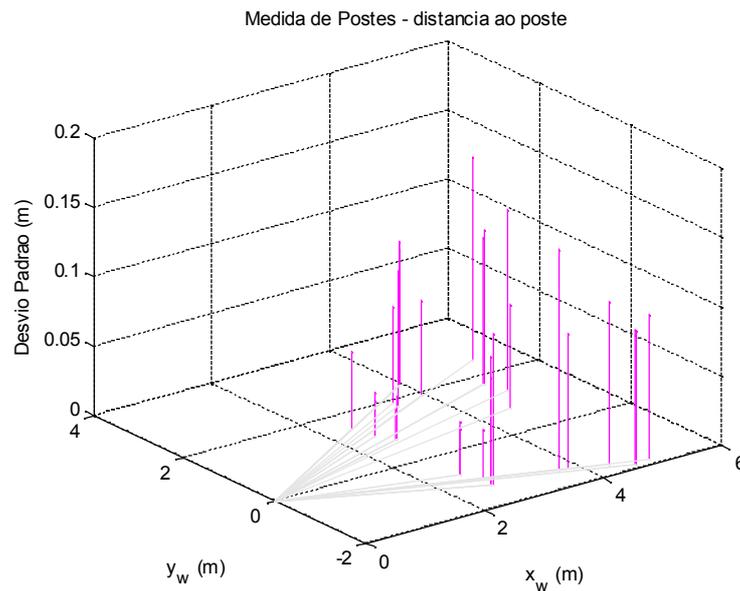


Figura 8.60 Diagrama da dispersão da distância na medida de postes

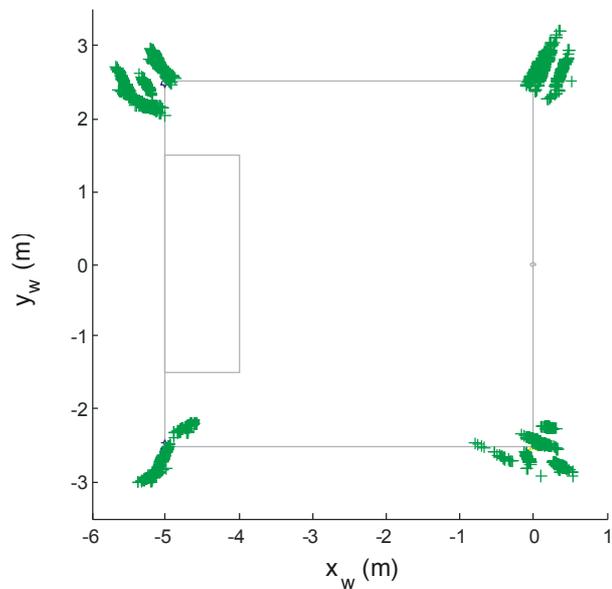


Figura 8.61 Medidas de postes representadas no campo

Tabela 8.4 Resumo das experiências relativas à medição de Postes

<i>Distância ao Poste (m)</i>	<i>Ângulo ao Poste (°)</i>	<i>Erro médio na distância (m)</i>	<i>Erro médio no ângulo (°)</i>	<i>Desvio Padrão Distância (m)</i>	<i>Desvio Padrão Ângulo (°)</i>
2.67	-15.6	0.06	1.0	0.03	0.0
4.64	31.5	0.13	5.2	0.09	0.0
4.63	31.5	0.12	5.2	0.10	0.0
4.64	31.5	0.13	5.2	0.10	0.0
4.63	31.5	0.12	5.2	0.10	0.0
4.63	31.5	0.12	5.2	0.10	0.0
4.66	9.7	0.15	-1.5	0.08	0.0
2.95	-26.0	0.16	0.5	0.09	0.0
2.95	-26.0	0.16	0.5	0.08	0.0
4.10	-20.6	0.06	1.1	0.10	0.0
4.68	-20.4	0.25	-4.0	0.12	0.0
4.38	25.8	0.34	4.0	0.07	0.0
3.96	-20.8	0.24	-1.1	0.16	0.1
5.87	25.6	0.28	7.1	0.15	0.1
2.87	31.6	0.08	5.0	0.06	0.0
2.98	-25.7	0.18	0.8	0.11	0.1
3.74	-26.4	-0.3	-4.6	0.23	0.1
5.25	-20.5	0.12	-3.5	0.10	0.1
3.91	27.4	0.19	7.7	0.10	0.1

<i>Distância ao Poste (m)</i>	<i>Ângulo ao Poste (°)</i>	<i>Erro médio na distância (m)</i>	<i>Erro médio no ângulo (°)</i>	<i>Desvio Padrão Distância (m)</i>	<i>Desvio Padrão Ângulo (°)</i>
3.98	29.0	0.15	5.3	0.07	0.1
2.92	-21.2	0.12	-2.7	0.03	0.0
2.68	-15.8	0.07	0.8	0.04	0.1
5.02	-21.7	0.09	-7.2	0.10	0.0
5.01	-21.7	0.08	-7.2	0.10	0.0
5.19	19.8	0.26	-0.6	0.11	0.1
5.20	19.8	0.27	-0.5	0.11	0.1
5.15	15.1	0.22	0.6	0.13	0.1

Pela análise dos dados anteriores pode-se verificar que as leituras do ângulo ao poste são consistentes e com valor absoluto do erro médio baixo. A medida da distância é, porém, repetidamente errada, o que se traduz num erro médio por vezes apreciável (mais de 30cm). Desta forma optou-se por valorizar mais a medida do ângulo ao poste e menos a medida da distância ao poste.

Os valores numéricos são:

$$\begin{aligned}\sigma_{Poste_d} &= 0.3 \text{ m} \\ \sigma_{Poste_a} &= 7^\circ = 0.12 \text{ rad}\end{aligned}\tag{8.42}$$

8.7 Hodometria

A calibração da Hodometria foi feita em 2 passos. Num primeiro conjunto de 5 experiências rectilíneas de 4 metros, obteve-se erros de localização inferiores a 2 cm, resultando numa precisão de 5 mm por cada metro. Num segundo conjunto de 5 missões de 1 uma volta completa sobre si próprio (360°), obteve-se um erro angular inferior a 3°, significando uma precisão superior a 0.008 grau sexagesimal por cada grau sexagesimal.

8.8 Ciclo de Fusão de Informação

O sistema dispõe de vários tipos de medidas. Atendendo ao facto que nem todas as medidas estão sempre disponíveis, a sequência de operações necessárias para o EKF pode ser resumida da seguinte forma:

- 1 Obter estimativas iniciais X_0 e P_0 ;
- 2 Executar em ciclo:
 - 2.1 Executar a previsão do EKF propagando $X(k^-)$ e $P(k^-)$;
 - 2.2 Se existir uma medida BC e se a medida for verosímil então actualizar essa medida;
 - 2.3 Se existir uma medida $Line$ e se a medida for verosímil então actualizar essa medida;
 - 2.4 Se existir uma medida $Poste$ e se a medida for verosímil então actualizar essa medida.

8.9 Simulação da Auto-Localização

Para avaliar o funcionamento global do sistema foi realizada uma série de simulações. Estas simulações utilizam valores realísticos para procurar avaliar a viabilidade do método proposto. São modelizados de uma forma realística todos os parâmetros provenientes da calibração do robot e das modelizações dos erros associados a cada tipo de medida.

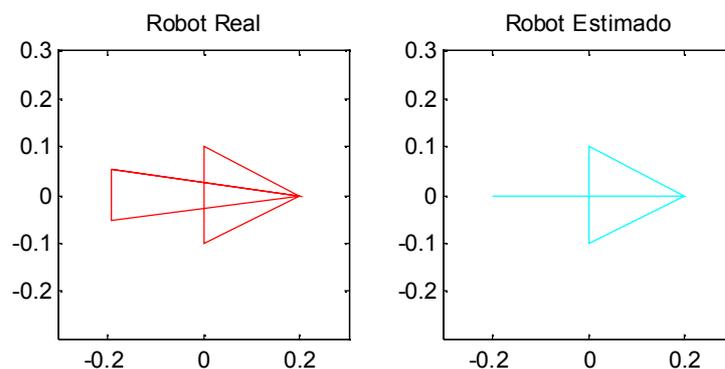


Figura 8.62 Detalhe dos robots real e estimado, representados em $x=0$, $y=0$, $\theta=0$

Nas simulações apresentadas de seguida o robot real é representado a vermelho numa seta larga. O robot simulado aparece como uma seta azul mais simples. Em torno do robot simulado está uma aproximação da elipse de covariância de

tal forma que a probabilidade do robot estar dentro da região assinalada é de 0.5. Como se pode confirmar na figura 8.62, o centro dos robots está no centro das setas para dar cobertura à noção de que quando o robot se move sobre ele próprio, as setas não se mexem também.

Na figura 8.63 está o traçado das referências dadas ao robot para as simulações seguintes. Nestas simulações pode-se apreciar a contribuição de cada estratégia de localização. Sempre que existe alguma informação de localização, tal facto é assinalado ao lado do robot estimado. Sempre que é fornecida informação de localização proveniente do primeiro poste, assinalado com uma estrela de 5 pontas, aparece um pequeno marcador ao lado do robot com essa forma. O mesmo se passa para as outras estratégias de localização, tal como se pode ver na legenda de cada figura. O segundo poste é marcado com uma estrela de 6 pontas e a localização proveniente da linha horizontal com um triângulo apontando para cima. Para a linha vertical é utilizado um triângulo que aponta para a direita. Quando existe informação BC, isso é assinalado através de um marcador em forma de \times .

Na figura 8.64 dado que não existem dados de localização, pode-se verificar como a estimativa não se aproxima do robot real.

Nas figuras 8.68 e 8.69, a informação de localização que é possível recolher das linhas não é suficiente para fazer convergir a simulação. A utilização de todos os elementos de localização de forma alternada mostrada na figura 8.71 assinala no entanto que esta informação é válida desde que o robot simulado esteja próximo do robot real.

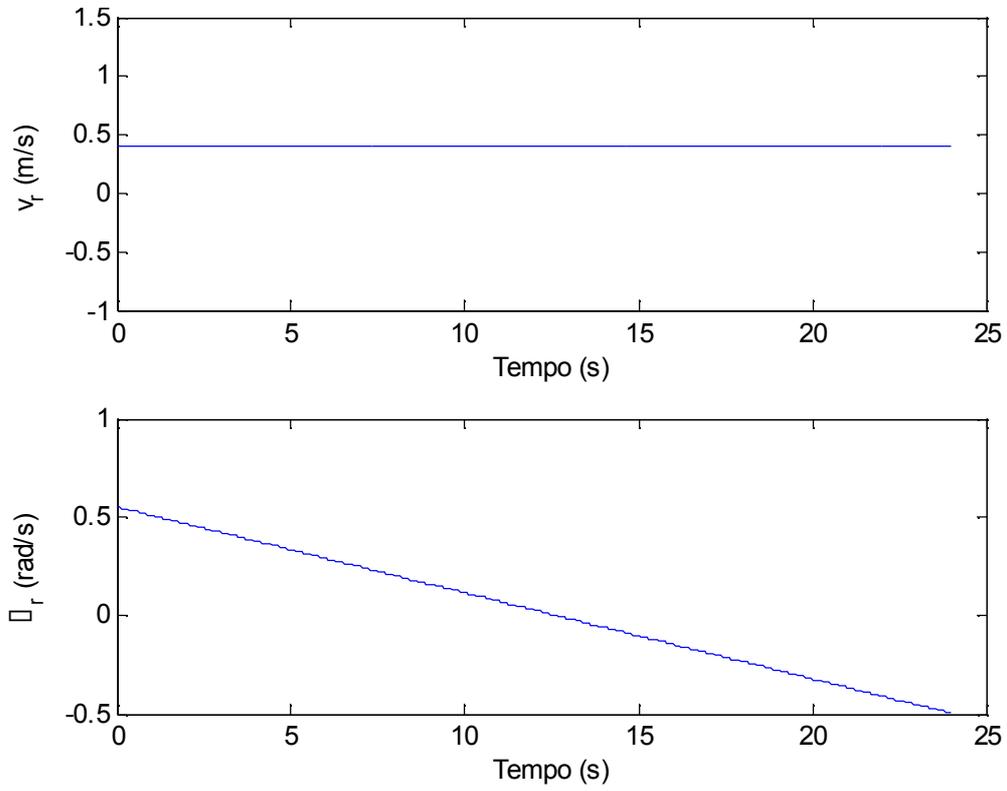


Figura 8.63 Traçados de v_r e ω_r

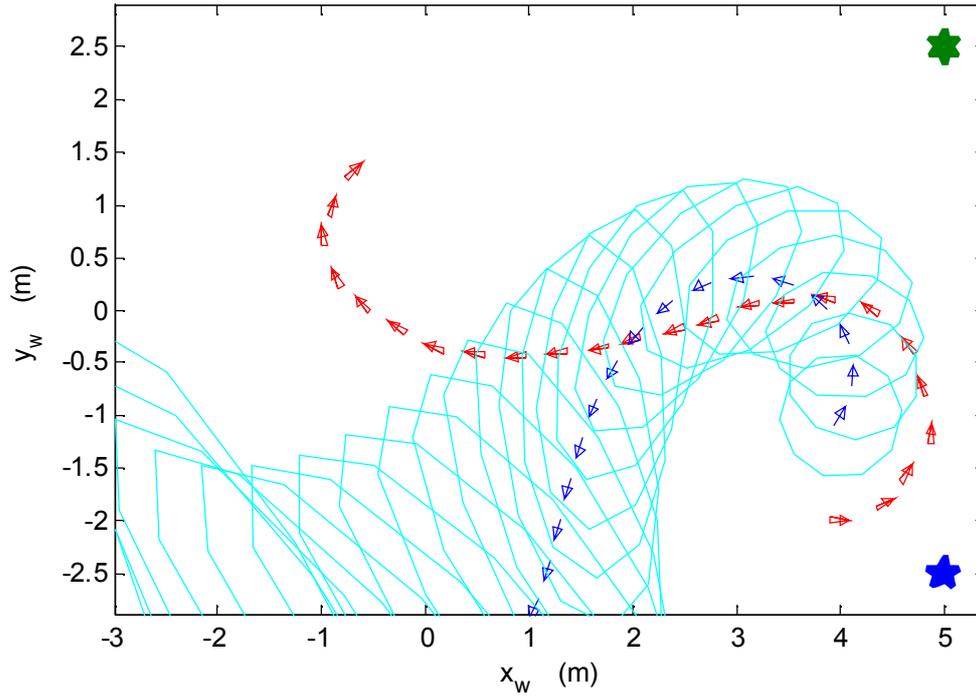


Figura 8.64 EKF sem nenhuma informação de localização

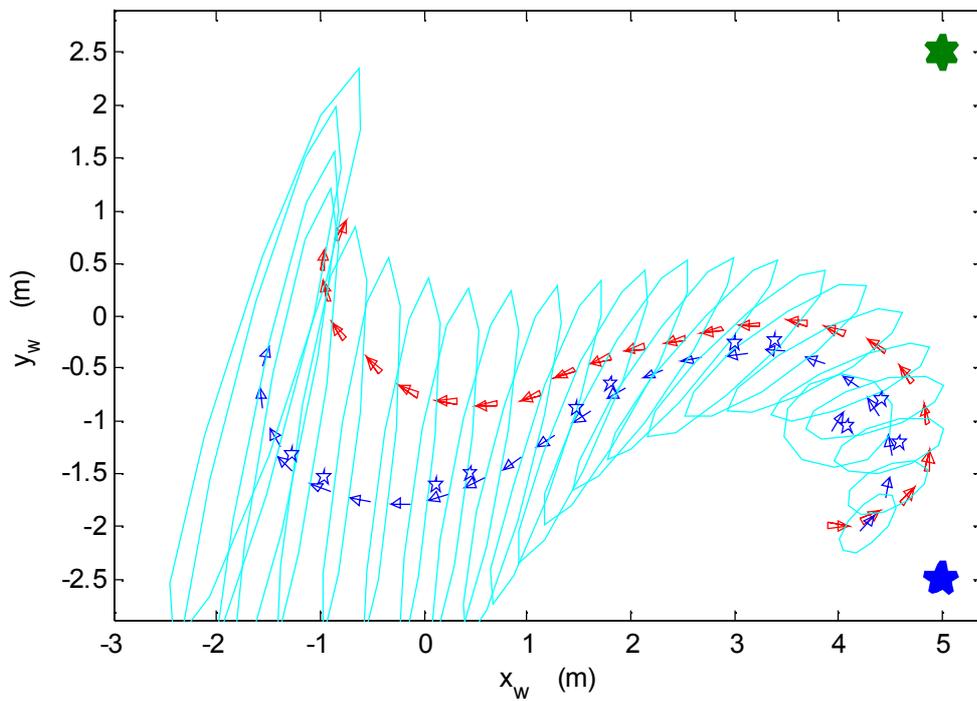


Figura 8.65 Simulação do robot localizando-se através do poste 1 (assinalado por uma estrela de 5 pontas)

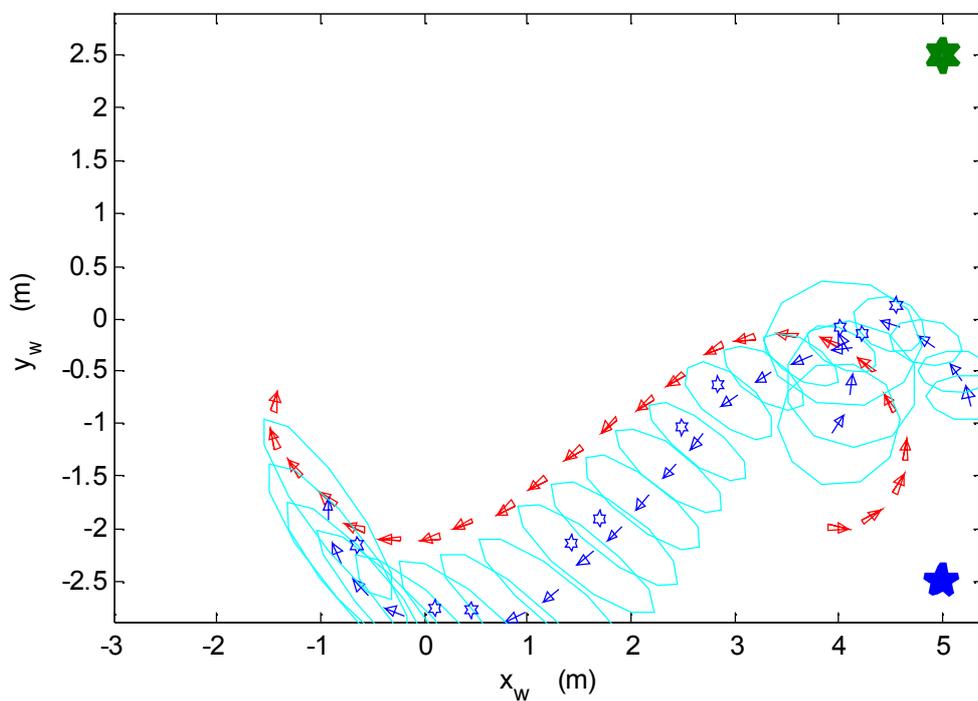


Figura 8.66 Simulação do robot localizando-se através do poste 2 (assinalado por uma estrela de 6 pontas)

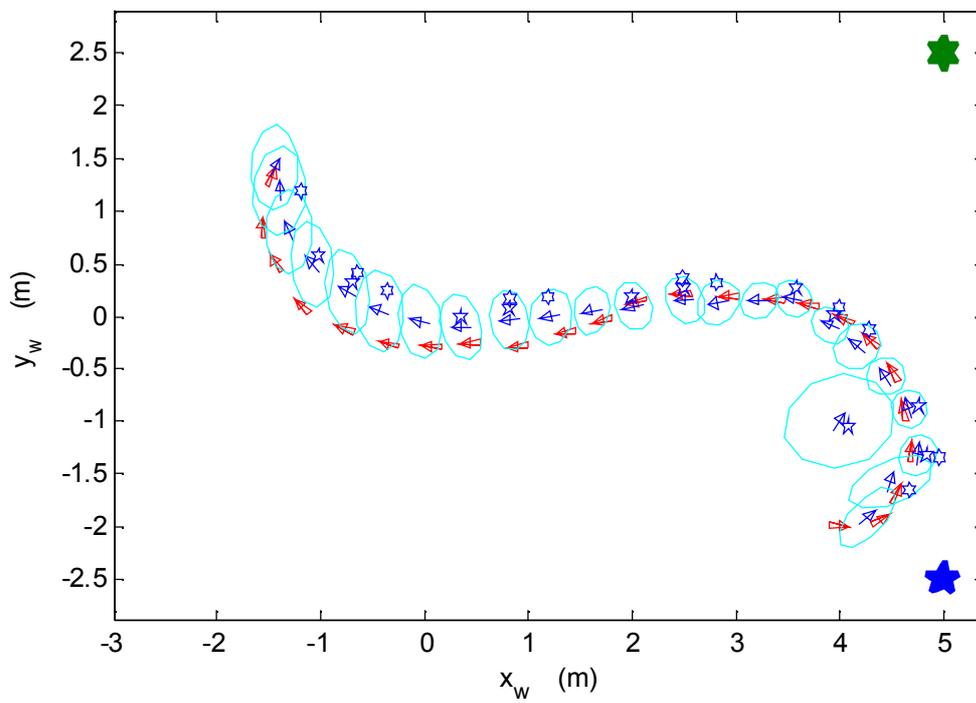


Figura 8.67 Simulação do robot localizando-se através dos postes 1 e 2

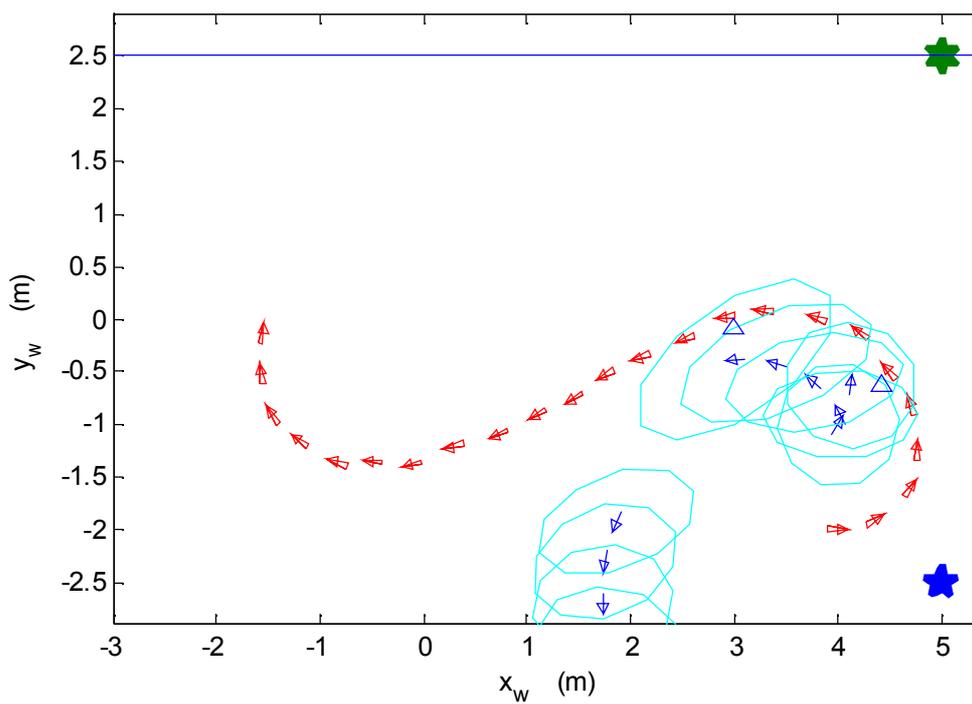


Figura 8.68 Simulação do robot localizando-se através da linha horizontal

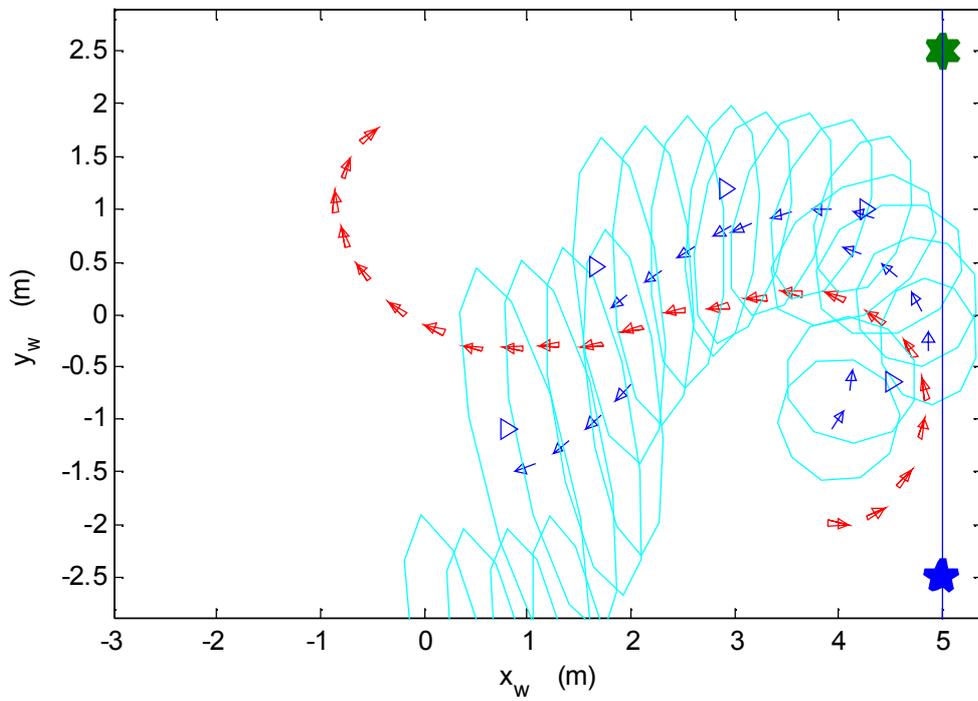


Figura 8.69 Simulação do robot localizando-se através da linha vertical

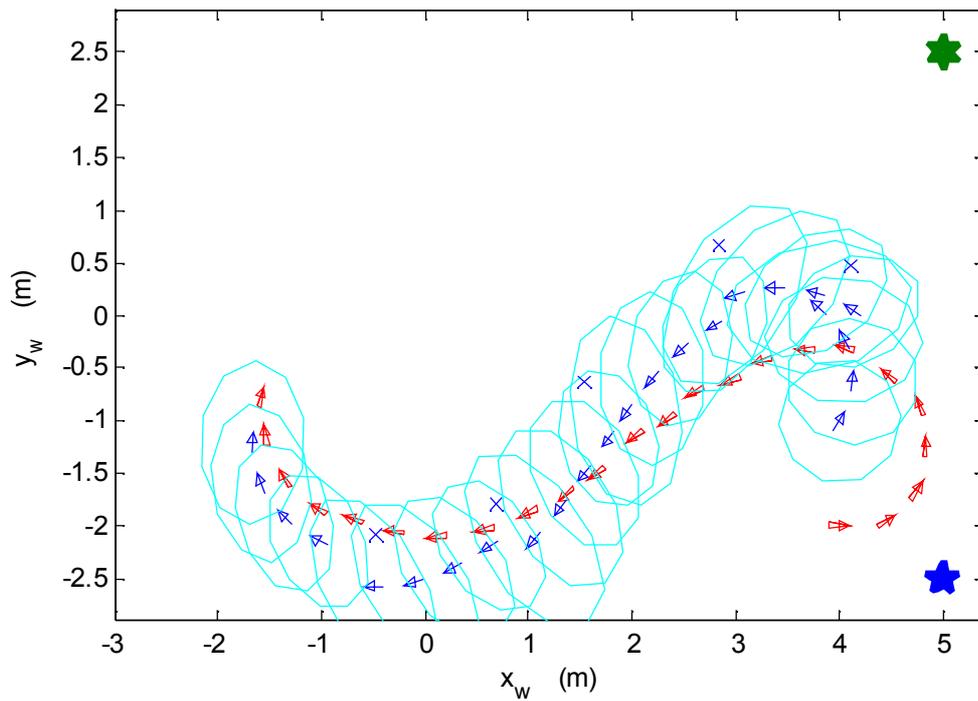


Figura 8.70 Simulação do robot localizando-se através de um CBU

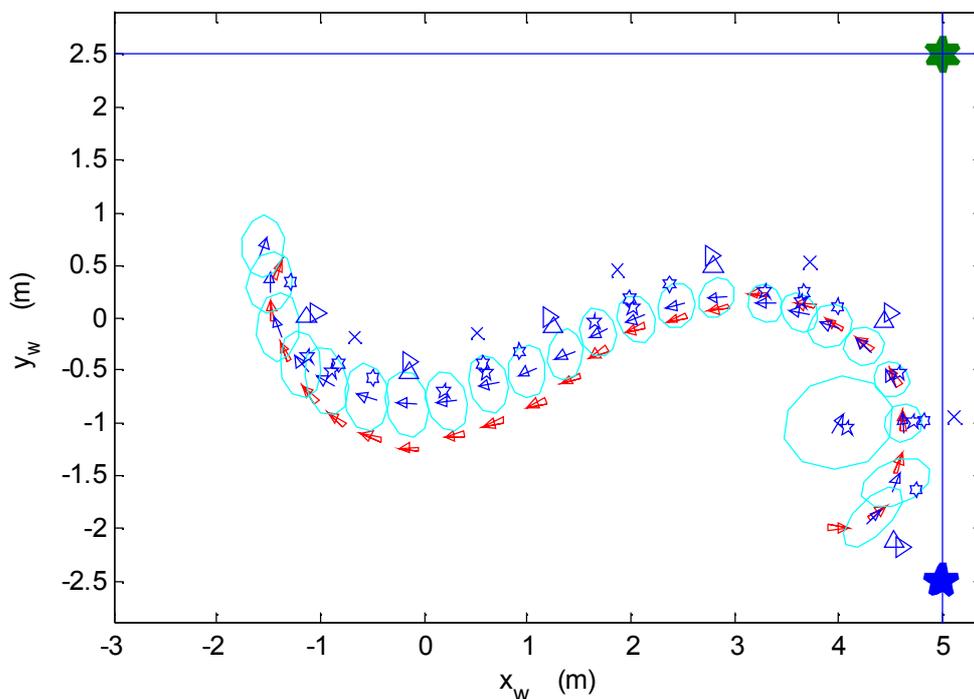


Figura 8.71 Simulação do robot localizando-se através da combinação de: 2 postes, 2 linhas e CBU

8.10 Testes Dinâmicos

Neste conjunto de testes pretende-se mostrar o funcionamento das estratégias de localização locais. Os diagramas mostrados são desenhados *off-line*, a partir de informação guardada durante a missão. Os testes dinâmicos decorrem numa parte do campo do futebol robótico, utilizado como ambiente estruturado para as experiências a realizar.

O desenho dos robots é tal como mostrado na figura 8.62, pág. 166. As medidas são, em geral, representadas a cor verde. A medida BC é um robot a verde (ver figura 8.42, pág. 147). A medida de linhas apresenta um “T” a verde (figura 8.51, pág. 154) e a linha a que foi afectada a medida é assinalada com um traçado preto largo. Algumas medidas de linhas não são afectadas a linha alguma e essas medidas são representadas com um “T” a linha interrompida de cor cinzenta. A posição dos postes é sempre assinalada no campo a traço fino. A medida de postes postes azuis é representada com um hexagrama e a medida de postes amarelos com um pentagrama e o poste correspondente é assinalado a preto com linha carregada.

Nos diagramas das figuras seguintes representa-se a covariâncias do robot no estado inicial e final. A trajectória do robot real aparece a traço vermelho claro

mas largo e a trajectória do robot estimado a traço largo, de cor ciano.

8.10.1 Localização através da medida de Linhas

Tal como referido anteriormente, as correcções que se podem efectuar com este tipo de medidas são limitadas. A quantidade de informação para as correcções melhora muito se forem avistadas num curto espaço de tempo linhas de direcção diferente, de preferência linhas perpendiculares.

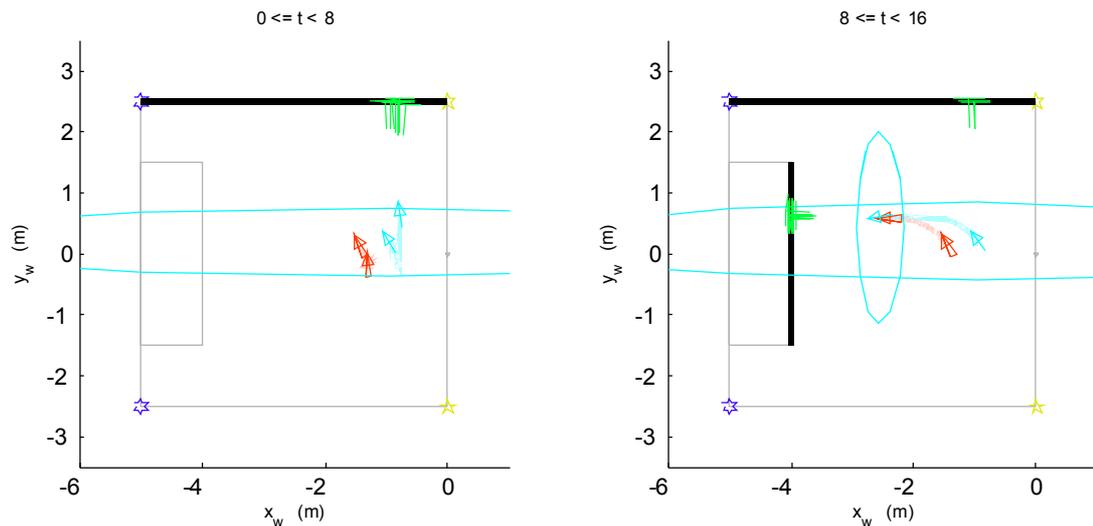


Figura 8.72 Teste dinâmico na localização por Linhas (tempos em segundos)

Na figura 8.72 pode-se ver o robot a partir de uma localização, a fazer medidas da linha lateral esquerda e o estado do sistema aproxima-se da posição yy correcta. Posteriormente é avistada a linha da área e a posição xx é também corrigida. A covariância do estado assinala a natureza da medida que existe e existiu recentemente e cresce na direcção em que não há informação.

8.10.2 Localização através da medida de Postes

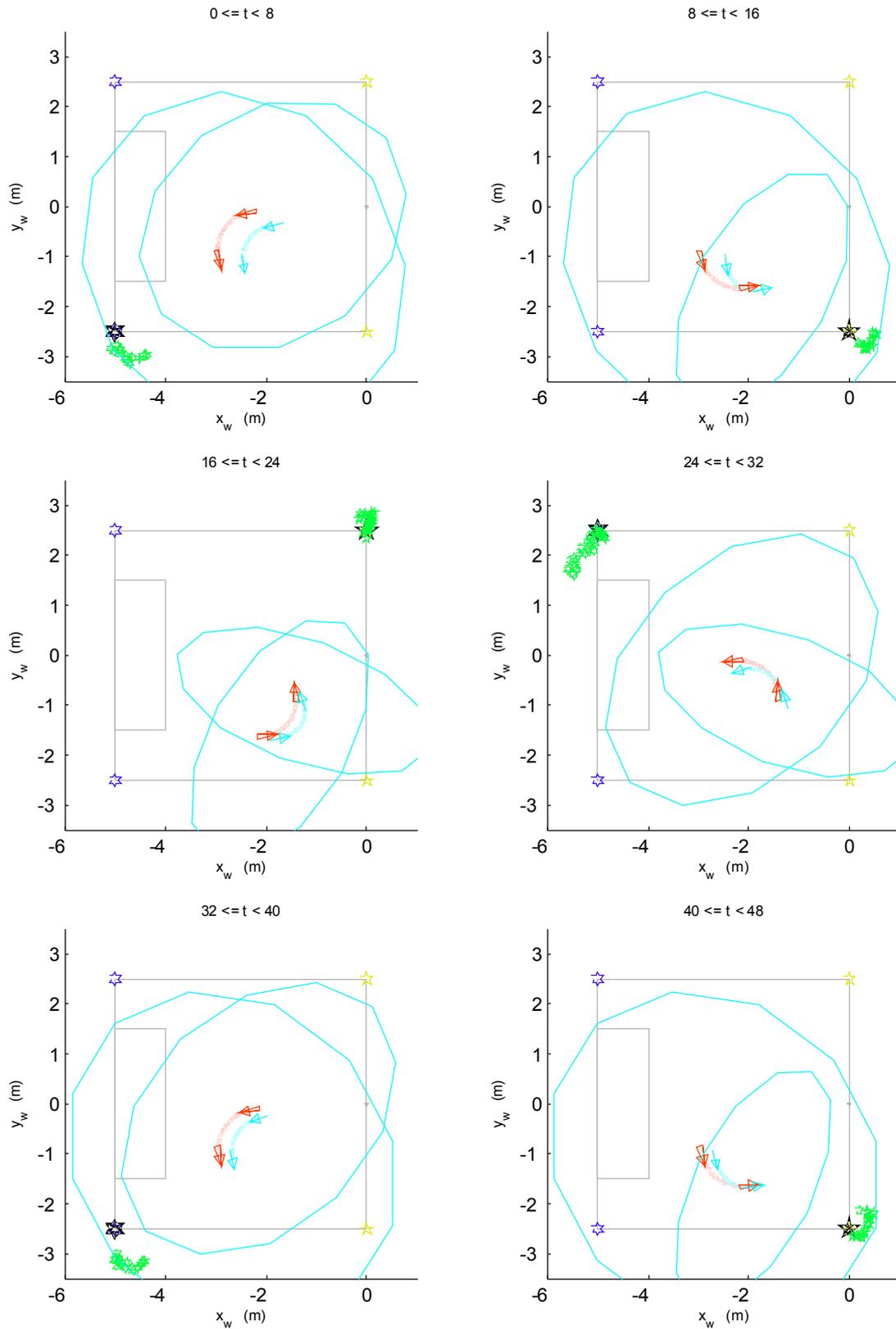


Figura 8.73 Teste dinâmico na localização por Postes (tempos em segundos)

A localização por medida de postes é também local. Devido à modelização feita da medida, a medida apresenta um conteúdo importante apenas na correcção da direcção em que é visto o posto (sendo que a distância ao poste não se demonstrou muito exacta).

Na figura 8.73 pode-se como o robot posto a rodar num circulo observa os diferentes postes alternadamente. As correcções angulares vão sendo feitas, a estimativa vai convergindo lentamente e ao fim de cerca de 48 segundos, o erro de localização é já muito pequeno e a covariância é baixa.

8.10.3 Localização completa

A medida por código de barras fornece a única estimativa global da posição. A experiência relatada pelo diagrama da figura 8.74 relata uma situação em que a estimativa inicial do robot está largamente divergente face à sua posição real.

Acompanhando a evolução da referida figura, veja-se como o robot (real) visualiza a linha de área da baliza azul e devido à sua errada estimativa pensa ter observado a linha de meio campo. Esta linha estaria no entanto a uma distância diferente e portanto a medida é inverosímil e conseqüentemente não é feita correcção alguma. As oscilações na trajetória são causadas pelas correcções relativas a ter sido visto o poste amarelo da direita do campo, o que não chega para localizar o robot. Posteriormente é avistada a linha lateral direita do campo e são feitas as correcções possíveis e a estimativa converge em y , mantendo elevado erro em x . Dos 8 segundos em diante, o robot real avista a linha central e a estimativa sofre correcções como se estivesse a ser avistada a linha final da baliza amarela. Não existe portanto convergência e esta situação é grave no sentido em que a covariância da localização diminui quando afinal o filtro está divergente. Aos 16 segundos de missão e até aos 24, há uma série de medidas de linhas que são classificadas de inverosíméis e uma série de medidas relativas ao poste amarelo da esquerda (também inverosíméis). Na sequência seguinte antes dos 32 segundos de missão, é avistado o código de barras de janela unívoca e a estimativa converge rapidamente sobre a medida BC, muito perto do robot real. À medida que o robot continua a rodar, deixa de ver o código de barras e passa a ver o poste azul esquerdo, coincidindo até as suas medidas com a posição real. São ainda feitas algumas medidas relativas à linha da área azul. No período até aos 40 segundos são feitas uma série de medidas de linhas que causam alguma instabilidade na estimativa uma vez que há alguma “confusão” relativamente a estar a ser avistada a linha lateral da área ou a linha lateral do campo – as medidas relativas ao poste azul direito fornecem pouca informação de distância e assim são fraco contributo para a convergência do filtro. Na sequência final, dos 40 segundos em diante há uma série de medidas de elevada qualidade que fazem o sistema convergir com pequeno erro. Neste espaço de tempo existe a medida do poste amarelo direito que faz pequenas correcções na direcção do robot. Existem ainda medidas relativas à linha lateral direita e à linha de meio

campo – uma vez que estas linhas são perpendiculares, temos uma elevada quantidade de informação de localização o que contribui para a pequena covariância que o estado do sistema apresenta no instante final da sequência apresentada.

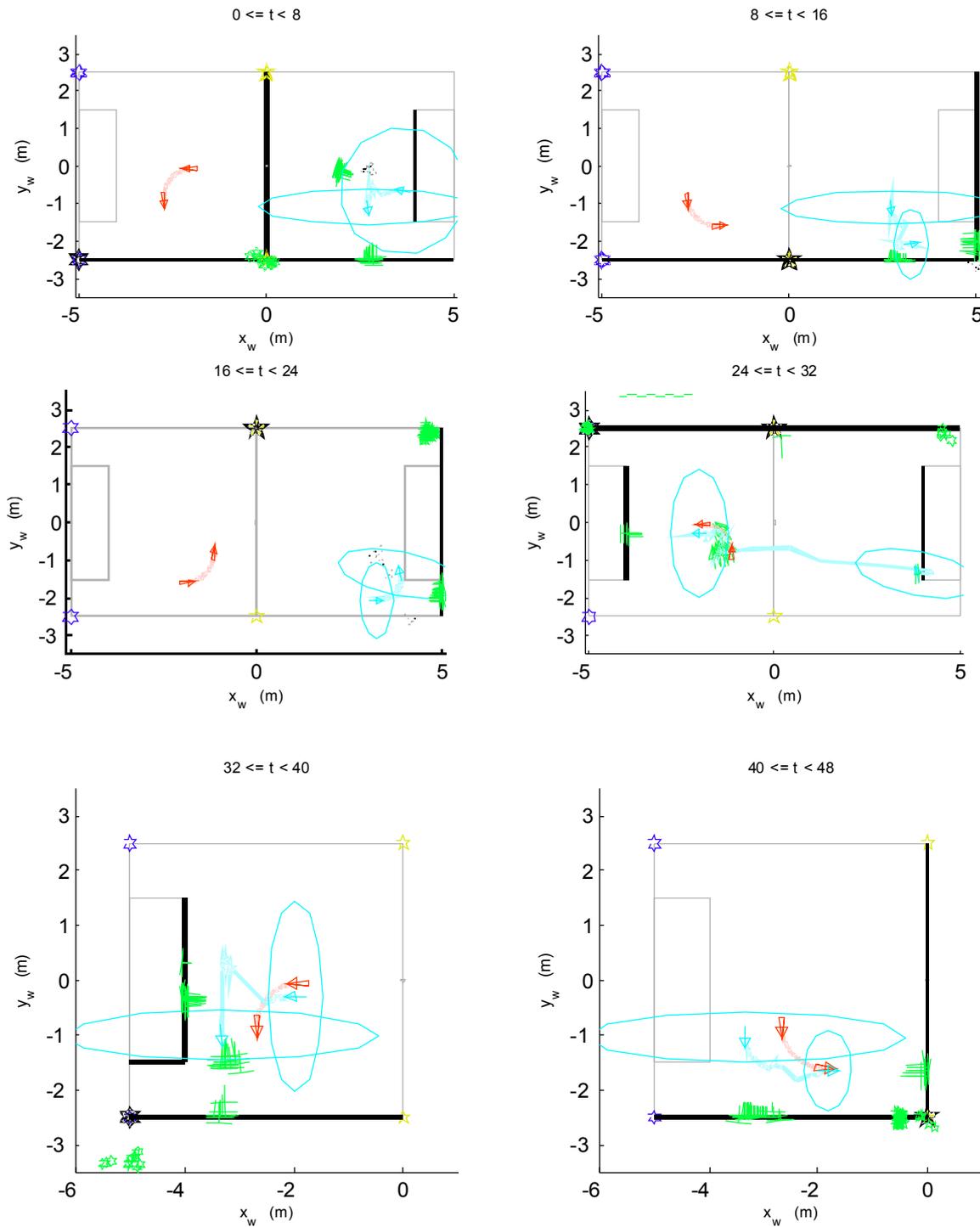


Figura 8.74 Localização recorrendo a BC, Postes e Linhas (tempos em segundos)

Tabela 8.5 Valores finais da sequência da figura 8.74

	$x (m)$	$y (m)$	$\theta (rad)$	$\theta (^\circ)$
VHC	-1.85	-1.61	-0.04	-2.4
EKF	-1.82	-1.66	-0.01	-0.8
Erro	0.03	-0.06	0.03	1.6

8.11 Conclusões

Este capítulo reúne o trabalho experimental realizado na área da localização.

Foi trabalhada a calibração da câmara para localização externa e foram caracterizados os dados fornecidos por este sistema.

De seguida foi apresentada a calibração da câmara embarcada no robot.

Vários métodos de auto-localização foram abordados de seguida: por reconhecimento de linhas do chão; por reconhecimento de postes e por Códigos de Barras Unívocos (BC). A localização por BC é um método inovador que permite a localização (x, y, θ) do robot através do processamento de visão normal que o robot partilha com o resto dos subsistemas do robot.

De seguida foi apresentado o EKF como elemento de fusão de informação que reúne todos os dados acessíveis num dado momento e devolve uma estimativa da localização do sistema.

Finalmente foram mostradas simulações que demonstram as vantagens e as limitações inerentes aos métodos apresentados.

A implementação da medida de linhas foi descrita e o erro da medida analisado. Esta medida parece interessante para a actual configuração da câmara no robot que permite ver linhas a uma distância não muito curta. Seria no entanto necessário melhorar a implementação da medida que neste momento não é suficientemente robusta a oclusões parciais. Para o ambiente do futebol robótico, este tipo de medidas parece interessante e outras aplicações podem tirar partido da elevada configurabilidade do sistema e assim melhorar a medida à custa de maior peso computacional. Seria assim possível detectar mais linhas que actualmente não são detectadas e também fazer descer a probabilidade de detecção de linhas em situações em que elas não existem (falsos positivos).

Relativamente à localização por postes, a medida conseguida é fiável no que diz respeito à direcção do poste mas não tanto relativamente à sua distância. Foi realizada uma modelização matemática da medida adequada à situação identificada. Mesmo com estas limitações, a localização é conseguida. Relativamente ao ambiente do futebol robótico, um sistema que permitisse ver mais vezes postes seria mais interessante que o actual em que é possível o robot não veja postes durante uma porção elevada da sua hipotética missão. Noutras aplicações os postes parecem um conceito interessante por exemplo para indicar direcções que o robot deve seguir.

Relativamente à medida por BCs foi feito o estudo da medida e ela apresenta excelente qualidade para aplicações fora do ambiente do futebol robótico (uma vez que as actuais regras não permitem este tipo de medida). A implementação conseguida apresenta alguma tolerância à ocultação parcial mantendo baixo baixo custo computacional e uma localização completa é realizada a cada 40 ms sem depender de dados anteriores. Está prevista a possibilidade futura de haver uma série de trechos de sequências unívocas espalhada pelo mapa conhecido do robot. Os códigos de barras utilizados utilizam visão a preto e branco para maior robustez de localização.

É admitido que em todas estas medidas (linhas, postes e BCs) é possível mas muito pouco provável existir ruído catastrófico, que pode inclusivamente ser indetectável como tal. Algum grau de tolerância ao ocultamento parcial foi, no entanto, conseguido mas pode ser melhorado.

A localização em ambiente estruturado é conseguida com recurso ao EKF para fusão de hodometria, de reconhecimento de linhas, de postes e de BCs. Tal filtro não apresenta garantias de convergência mas esta verifica-se, na prática.

O erro de localização no final da sequência experimental mais completa é de cerca de 10 cm e 2°, num mapa de cerca de 5 m por 5 m, o que constitui um muito bom resultado para o cenário experimental analisado.

Capítulo 9

Conclusões

Esta dissertação aborda o projecto de sistemas robóticos. A complexidade e multi-disciplinaridade deste tipo de projectos levam à necessidade de especial cuidado na sua execução. Foram discutidas escolhas de projecto e de implementação para diversas questões que vão desde a forma de locomoção de um robot até ao seu sistema de decisão a um elevado nível de abstracção. Foi também mencionada a filosofia que o projecto deve manter e foi citado um percurso típico de projecto deste tipo de sistemas. A perspectiva mostrada reveste-se de importância pela sua abrangência e aplicabilidade prática.

A autonomia em robótica, associada com a mobilidade, conduz à necessidade de localização. O tema da localização robótica foi discutido e foram caracterizadas diversas técnicas frequentemente utilizadas. Foram citados trabalhos relevantes na área e foi discutida a respectiva aplicabilidade.

Apresentou-se a formalização relativa ao conceito inovador de *codificador de janela deslizante* e correspondente *sequência constituída por janelas unívocas*. Trata-se de uma extensão dos conceitos relativos a codificadores e a códigos de barras. Exemplos ilustrativos foram dados para mostrar as aplicações a que este conceito dá lugar.

Os dois mais recentes projectos robóticos em que autor esteve envolvido são apresentados e discutidos na perspectiva que foi advogada anteriormente. São

robots futebolistas para a liga F180 e F2000 do RoboCup. As decisões de projecto são discutidas e justificadas. O trabalho experimental utiliza a plataforma destinada à liga F2000.

Os aspectos mais relevantes relativos à aplicação prática desenvolvida são:

- Construção e caracterização de um sistema de localização externo (VHC), de elevada precisão – serve de controlo para o resto da experimentação;
- Implementação e caracterização de um sistema de reconhecimento de linhas rectas em visão a cores;
- Implementação e caracterização de um sistema de reconhecimento de postes coloridos para navegação visual;
- Implementação e caracterização de um sistema de localização baseado em código de barras. Este sistema utiliza o conceito anteriormente formalizado do codificador de janela deslizante. A localização conseguida é global (x, y, θ) e é unicamente baseada em visão;
- Projecto, simulação e implementação de um Filtro de Kalman Extendido (FKE) para fusão sensorial dos dados disponíveis. A convergência deste tipo de filtros não é assegurada. As limitações inerentes a cada tipo de medida foi referida e foram confirmados os benefícios da utilização de vários tipos de medida para a convergência do filtro. Foi demonstrado experimentalmente que, em situação realista e utilizando todas as medidas disponíveis, a convergência se verifica. A correcção de cada medida depende, no entanto, de uma série de testes de verosimilhança da medida que por sua vez afectam a convergência global do filtro;
- Para facilitar a localização recomenda-se:
 - Utilização de linhas perpendiculares – o conjunto fornece uma boa informação de localização;
 - Evitar linhas curvas – este caso não foi tratado;
 - Marcar pontos de referência importantes com postes, de preferência na periferia da zona onde o robot se movimenta – os postes dão boa informação de direcção e alguma informação de distância;
 - Os códigos de barras dão boa informação de localização mas precisam de visibilidade adequada;
- É admitido que em todos os casos é possível, se bem que muito pouco provável, existir ruído catastrófico que pode inclusivamente ser indetectável como tal;
- Demonstrou-se a viabilidade prática do sistema de localização montado, numa situação próxima da que existe na liga F2000 do futebol robótico do RoboCup mas que não se restringe apenas a essa situação;

- Todos os algoritmos de localização propostos são adequados a desempenho em Tempo Real em sistemas com poder de cálculo limitado.

De forma resumida, pode dizer-se que este trabalho explora várias áreas da robótica, apresentando as mais importantes questões de projecto. Uma vez estabelecida a importância da localização em robótica, este assunto é trabalhado a nível experimental, utilizando conceitos inovadores. A experimentação realizada resulta numa elevada precisão de localização, inferior a 0.10 m em cada direcção e inferior a 2 graus sexagesimais na orientação, para um mapa de cerca de 5 por 5 metros.

9.1 Trabalho futuro

A nível dos projectos robóticos referidos, os melhoramentos mais relevantes a nível científico são no âmbito do Sistema Robótico destinado à Liga F2000:

- Estudar a mudança para um sistema omni-direccional de visão – um sistema omni-direccional de visão tornará, possivelmente, o sistema de localização por visão mais robusto;
- A arquitectura do sistema tem-se revelado limitativa – a actual partição entre HAL e DEC impede iterações de processamento adicionais sobre a imagem. Seria mais natural que a aplicação HAL passasse a reconhecer objectos e enviasse essa interpretação do mundo para a aplicação DEC.

A nível do sistema de localização desenvolvido, vários melhoramentos são possíveis:

- O sistema de medida de linhas necessita de melhoramentos que podem ser feitos a dois níveis:
 - Ao nível mais imediato, pode-se estudar o suporte para um maior número de rectas em diversas posições para retirar mais informação da imagem;
 - Ao nível mais profundo, é necessário um novo sistema de detecção de linhas rectas na imagem que percorra a imagem ou regiões da imagem, a ser realizado dentro da aplicação HAL;
- O sistema de detecção de postes deve ser também melhorado, no entanto, alguns melhoramentos implicam que a detecção de postes passe a ser realizada dentro da aplicação HAL;
- A localização por códigos de barras tem grande margem para desenvolvimento:
 - Estudar suporte para códigos de barras no chão e no tecto;

- Estudar a influência da dimensão do código de barras na qualidade da localização;
- Avaliar a qualidade da localização a distâncias mais elevadas;
- Implementar diversos trechos em diversas posições do mapa do robot;
- Comparar a precisão obtida com a situação equivalente mas com o padrão regular separado do código unívoco;
- Estudar a implementação de localização BC multi-resolução, isto é, ter vários tamanhos de código de barras para fazer a localização;

Os trabalhos descritos nesta dissertação não se encerram com a apresentação do mesmo e grande parte das questões referidas estão já a ser preparadas no âmbito da equipa 5dpo de futebol robótico da FEUP.

Capítulo 10

Referências Bibliográficas

- [1] **Devdas Shetty, Richard Kolk**, "*Mechatronics System Design*", Published by PWS Publishing (1998) 0-534 95285-2
- [2] **Kevin Craig**, "*Is Anything Really New in Mechatronics Education ?*", IEEE Robotics and Automation Magazine, Vol 8, No 2, Junho 2001 12-19 (2001)
- [3] **J. Borenstein, H. Everett, L. Feng**, "*Where am I ? Systems and Methods for Mobile Robot Positioning*", A. K. Peters, Ltd, Wellesley, MA, EUA (1996)
<http://www.eng.yale.edu/ee-labs/morse/other/intro.html>
- [4] **Paulo José Costa**, "*Localização em Tempo Real de Múltiplos Robots num Ambiente Dinâmico*", Tese de Doutoramento, Faculdade de Eng^a Univ. Porto (1999)
- [5] **Ton Peijnenburg**, "*Philips CFT RoboCup Team Description*", Proceedings of the RoboCup 2003, Pádua, Italy (2003)
http://ais.gmd.de/robocup/msl2003/QualificationMSL2003/tdp_philips.pdf
- [6] **J Borenstein**, "*Compliant-linkage Kinematic Design for Multi-degree-of-freedom Mobile Robots*", Proceedings of the SPIE Symposium on Advances in Intelligent Systems, Mobile Robots VII, Boston, EUA, 1992 344-351 (1992)

- [7] **David P. Anderson**, "*nBot Balancing Robot*" <http://www.geology.smu.edu/~dpa-www/robo/nbot/>
- [8] **Brian Yamauchi***, **Polly Pook**, and **Amanda Gruber**, "*Bloodhound: A Semi-Autonomous Battlefield Medical Robot*", 23rd - Army and Science Conference - Dezembro 2002 (2002) <http://www.asc2002.com/summaries/n/NP-06.pdf>
- [9] **Shigeo Hirose**, "*Super Mechano-System: New Perspective for Versatile Robotic System*", Proceedings of ISER '00 - International Symposium On Experimental Robotics 2000 - Carnegie Mellon University Pittsburgh, Pennsylvania, EUA (2000) <http://www.ri.cmu.edu/events/iser00/papers/hiroseISER00.pdf>
- [10] http://www.superdroidrobots.com/wheels_omni.htm
- [11] <http://www.mrrobot.com/omni.html>
- [12] "*4cm omni-directional roller wheel*" <http://www.acroname.com/robotics/parts/R77-4CM-ROLLER-3.html>
- [13] **Grigoriy B. Reshko**, **Matthew T. Mason** and **Illah R. Nourbakhsh** , "*Rapid Prototyping of Small Robots*" , Tech Report CMU-RI-TR-02-11, Robotics Institute, Carnegie Mellon University (2002) http://www.ri.cmu.edu/pub_files/pub3/reshko_greg_2002_1/reshko_greg_2002_1.pdf
- [14] **Prabhuram Raghunathan**, **Oliver Purwin**, **Jin Woo Lee** and **Rafaello D'Andrea**, "*The Cornell BigRed Team Foci for RoboCup 2003*", Proceedings of the RoboCup 2003, Pádua, Italy (2003) <http://robocup.mae.cornell.edu/2003long.pdf>
- [15] **Fernando Ribeiro**, **Paulo Braga**, **Jorge Monteiro**, **Ivo Moutinho**, **Pedro Silva e Vitor Silva**, "*New improvements of MINHO Team for RoboCup Middle Size League in 2003*", Proceedings of the RoboCup 2003, Pádua, Italy (2003) http://ais.gmd.de/robocup/msl2003/QualificationMSL2003/tdp_minho.pdf
- [16] **Paulo Costa**, **Armando Sousa**, **Paulo Marques**, **Pedro Costa**, **Susana Gaio e António Moreira**, "*5dpo Robotic Soccer Team for Year 2003*", Proceedings of the RoboCup 2003, Pádua, Italy (2003)
- [17] **António Paulo Moreira**, **Paulo Costa**, **Armando Sousa e Luís Paulo Reis**, "*5dpo-2000 Team Description for Year 2003*", Proceedings of the RoboCup 2003, Pádua, Italy (2003)
- [18] **J Borenstein**, "*Omnipede*" http://www.engin.umich.edu/research/mrl/00MoRob_6.html
- [19] **João Alves e Jorge Dias**, "*Design of a Spherical Robot and its Control Electronics*", Proc. of Controlo 2002 - Conf. Automatic Control, Aveiro, Portugal, pp. 495-500 (2002)
- [20] **Aarne Halme**, **Torsten Schönberg** and **Yan Wang**, "*Motion Control of a Spherical Mobile Robot*", Proc. AMC'96- MIE (1996) citeseer.nj.nec.com/halme96motion.html
- [21] **PBS Online by WGBH** "*Bomb Squad - The Spiral Track Autonomous Robot (STAR)*" 2000 <http://www.pbs.org/wgbh/nova/robots/hazard/meetstar.html>
- [22] "*Micromechanical Flying Insect (MFI) Project*", University of California, Berkeley, EUA <http://robotics.eecs.berkeley.edu/~ronf/mfi.html>

-
- [23] **Brett Warneke, Matt Last, Brian Liebowitz and Kristofer S.J.Pister**, "*Smart Dust: Communicating with a Cubic- Millimeter Computer*" (2001)
http://citeseer.nj.nec.com/cache/papers/cs/25491/http.zSzzSzbsac.eecs.berkeley.edu/zSzpaperszSzfinal_layout.pdf/warneke01smart.pdf
- [24] **Gilles Caprari**, "*Autonomous MicroRobots Applications and Limitations*", PhD Thesis, EPFL, Suíça (2003)
http://asl.epfl.ch/aslInternalWeb/ASL/publications/uploadedFiles/Caprari_phd2753.pdf
- [25] **Y. Fujimoto and S. Obata and A. Kawamura**, "*Robust Biped Walking with Active Interaction Control between Foot and Ground*", Proc. IEEE Int. Conf. on Robotics and Automation, 1998, pp. 2030-2035 (1998)
citeseer.nj.nec.com/fujimoto98robust.html
- [26] **J. Hodgins**, "*Three-Dimensional Human Running*", Robotics and Automation (1996)
citeseer.nj.nec.com/71467.html
- [27] <http://www.honda-p3.com>
- [28] "*Sony Humanoid SDR 4X II*" <http://www.i4u.com/article255.html>
- [29] "*Biologically Inspired Robotics Group (BIRG)*", EPFL - École Polytechnique de Lausanne <http://slwww.epfl.ch/birg/humanoid.shtml>
- [30] <http://www.ai.mit.edu/projects/lelab/robots/robots.html>
- [31] <http://animatlab.lip6.fr/pages/LiensRecherchesAng>
- [32] "*The Scorpion Robot*" <http://ais.gmd.de/BAR/SCORPION>
- [33] <http://slwww.epfl.ch/birg/salamander.shtml>
- [34] "*Troody Walks Across a Table*" <http://www.ai.mit.edu/people/chunks/chunks.html>
- [35] **Bruce M. Blumberg**, "*Silas T. Dog*" <http://lcs.www.media.mit.edu/people/bruce>
- [36] "*Sony Aibo Site*" <http://www.aibosite.com>
- [37] "*NeCoRo*" http://www.omron.com/news/n_161001.html
- [38] <http://www.ai.mit.edu/projects/ants>
- [39] <http://www.sciencenews.org/20001111/bob1.asp>
- [40] **Gavin Miller**, "*Dr. Gavin Miller's Snake Robots*" <http://www.snakerobots.com/main.htm>
- [41] **Jun Nakanishi, Toshio Fukuda, Daniel E. Koditschek**, "*A Brachiating Robot Controller*", IEEE Transactions on Robotics and Automation 16, no. 2 - 2000 pp. 109-123 (2000)
- [42] **Robert G. Dennis, Hugh Her**, "*Biomechatronic Devices*"
http://www-personal.umich.edu/~bobden/biomechatronic_devices.html
- [43] "*Robotics Society of America - Robot Shark Takes Plunge*"
<http://www.robotics-society.org/html/modules.php?name=News&file=article&sid=15>
- [44] **Joseph Ayers**, "*Biomimetic Underwater Program*" (2002)
- [45] "*Robot Tuna at MIT*" <http://web.mit.edu/towtank/www/tuna/brad/tuna.html>
-

- [46] <http://islwww.epfl.ch/birg/lamprey.shtml>
- [47] **Ismet Erkmen, Aydan M Erkmen, Fumitoshi Matsuno, Rabajit Chatterjee and Tetsushi Kamegawa**, "Snake Robot to the Rescue - Serpentine Search Robot in Rescue Operations", IEEE Robotics & Automation Magazine, Vol 9, No 3 Setembro 2002 pp. 17-25 (2002)
- [48] **Rajiv S. Desai, Charles J. Rosenberg, Joseph L. Jones**, "Kaa : An Autonomous Serpentine Robot Utilizes Behavior Control", IROS'95 IEEE/RSJ International Conference on Intelligent Robots and Systems, Pittsburgh, EUA (1995)
- [49] **L. Sebastião, C. Silvestre, A. Pascoal**, "Infante project Report 3" (1999)
<http://dsor.isr.ist.utl.pt/Projects/Infante/Report3/index.html>
- [50] **Author Poole, Patrick K. Clower, Penn**, "A systems approach to autonomous underwater vehicle (AUV) propulsion design", Source Marine Technology Society Journal. v. 30 p. 8-16. (1996)
- [51] **SubSea Propulsion A Division of DynaSea International Inc. Newburyport, Mass 01950**, "U.S.A. SubSea - AUV/ROV Propulsion Thrusters"
<http://www.roworld.com/promote/subseaprop/Ssp1.pdf>
- [52] **Fernando Lobo Pereira**, "Laboratório de Sistemas e Tecnologia Subaquática"
<http://www.fe.up.pt/~lsts/>
- [53] **Mikell P. Groover**, "Automation, Production and Computer Integrated Manufacturing", Prentice Hall (1987) 0-13-054652-6
- [54] **U. Rembold and R. Dillmann**, "Computer Aided Design and Manufacturing", Springer-Verlag - Berlin (1984) 3-540-16321-2
- [55] **R. Brooks and S. Iyengar**, "MultiSensor Fusion", Prentice Hall PTR (1998) 0-13-901653-8
- [56] "**PTC - Parametric Technology Corporation**" <http://www.ptc.com/products/index.htm>
- [57] **D. Dzung, C. Apneseth, G. Scheible**, "Wireless Sensor Communications and Powering for Real-Time Industrial Applications", WiP Proceedings of the 2002 IEEE International Workshop on Factory Communication Systems, August 28-20, 2002, Vasteras, Sweden (2002)
- [58] **Pedro Lima, Luís Custódio, Miguel Arroz, Marco Barbosa, Tiago Borrvalho, Cláudio Cardoso, Hugo Costelha, Bruno Damas, João Estilita, João Frazão, Miguel Garção, Cláudio Gil, Paulo Gomes, Bruno Machado, Gonçalo Neto, Pedro Pinheiro, Vasco Pires, João Santos, Rodrigo Ventura, Hans Lausen Jakob, Michael Nielsen**, "ISocRob 2003: Team Description Paper", Proceedings of the RoboCup 2003, Pádua, Italy (2003)
- [59] "Batteries for Mobile Robots" <http://robocup.mi.fu-berlin.de/buch/chap6/06robocup-battery.pdf>
- [60] **Isidor Buchmann**, "Batteries in a Portable World" (2001) <http://www.buchmann.ca/faq.asp>
- [61] **Isidor Buchmann**, "Do and don't battery table" (2001)
<http://www.batteryuniversity.com/print-partone-21.htm>

-
- [62] **Graham Prophet**, "*SuperCaps for SuperCaches*", EDN, 9 de Janeiro de 2003
<http://www.e-insite.net/ednmag/contents/images/268379.pdf>
- [63] <http://www.e-insite.net/ednmag/contents/images/268379f1.pdf>
- [64] **DaimlerChrysler**, "*Energy for the Future - Fuel Cell Bus*", DaimlerChrysler High tech Report 2/2002
- [65] <http://www.fuelcells.org>
- [66] **Michael Riezenman**, "*Energy Storage - Mighty Mites - Fuel Cell's First Big Market Success May be Alcohol Fed*", IEEE Spectrum - Junho 2003
- [67] **Alan Burns and Anfy Wellings**, "*Real Time Systems and Programming Languages*", Pearson Education Limited (2001) 0 201 72988 1
- [68] **Richard Dorf, Editor**, "*Electrical Engineering Handbook*", CRC Press (1993)
- [69] **John G. Webster (Editor)**, "*The Measurement, Instrumentation and Sensors Handbook*", CRC Press & IEEE Press (1999)
- [70] "*The Basics: Laser Radar – Lidar*"
https://peoiewswwebinfo.monmouth.army.mil/JPSD/RTV/LIDAR_basics.pdf
- [71] **Paulo Costa, Armando Sousa, Paulo Marques, Pedro Costa, Susana Gaio, António Moreira**, "*5dpo Team Description*", in RoboCup 2001: Robot Soccer World Cup V Editor: Andreas Birk, Silvia Coradeschi, Stoshi Tadokoro- Springer-Verlag Berlin/Heidelberg New York 2002 (2001) ISBN: 3-540-43912-9
- [72] **J. Almeida, A Martins, E. Silva, J. Baptista, A. Patacho, L. Lima, V. Serqueira, C. Almeida, R. Picas**, "*ISePorto Robotic Soccer Team for Robocup 2003*", Proceedings of the RoboCup 2003, Pádua, Italy (2003)
- [73] **G. De Micheli (editor)**, "*Special Issue on Hardware/Software Co-design*", Proceedings of IEEE, Vol 85, No. 3, Março 1997
- [74] **Jean Claude Latombe**, "*Robot Motion Planning*", Kluwer Academic Publishers (1991)
- [75] **J. Borenstein, H. Everett, L. Feng, D. Wehe**, "*Mobile Robot Positioning - Sensors and Techniques*", Journal of Robotic Systems Vol 14, No. 4 Special issue on Mobile Robots 231-249 (1997)
- [76] **Ronald Arkin**, "*Behaviour Based Robotics*", The MIT Press, London, England (1998)
- [77] "*EPFL Collective robotics - BIRG Group*" <http://slwww.epfl.ch/birg/collective.shtml>
- [78] **L. Wald**, "*Definitions and Terms of Reference in Data Fusion*", International Archives of Photogrammetry and Remote Sensing, Vol. 32, Part 7-4-3 W6, Valladolid, Spain, 3-4 June, 1999 pp. 2-6. (1999) citeseer.nj.nec.com/wald99definitions.html
- [79] **Geert De Cubber**, "*Integration of sensors on a mobile robot*", PhD Thesis - Free University of Brussels, Bélgica, 2001
-

- [80] **Ian Reid**, " *Applied Estimation Notes 2002*"
<http://www.robots.ox.ac.uk/~ian/Teaching/Estimation/estimation2.pdf>
- [81] **J. Almeida Costa e A. Sampaio e Melo**, " *Dicionário de Língua Portuguesa - 7ª Edição*",
Porto Editora (1994) 972 0 05001 2
- [82] **B. Barshan and H. F. Durrant-Whyte**, " *Inertial navigation systems for mobile robots*",
IEEE Transactions on Robotics and Automation, vol. 11, no. 3 328-342
(1995)
- [83] **Charles Cohen and Frank Koss**, " *A comprehensive study of three-object triangulation*",
In SPIE Mobile Robots VII, 1992
- [84] **João Sena Esteves, Adriano Carvalho, Carlos Couto**, " *Generalized Geometric
Triangulation Algorithm for Mobile Robot Absolute Self-Localization*", Proceedings of
the IEEE International Symposium on Industrial Electronics 2003, June 9-11,
Rio Janeiro Brasil (2003)
- [85] **M. Betke and K. Gurbits**, " *Mobile robot localization using landmarks*", Proceedings of
the IEEE International Conference on Robotics and Automation, volume 2,
May 1994. pp. 135-142 (1994) citeseer.nj.nec.com/betke97mobile.html
- [86] " *Trimble - All about GPS* " <http://www.trimble.com/gps>
- [87] " *Sick* " [Positioning at 240 m with millimetre accuracy](http://www.sick.de/de/products/categories/auto/lasermeasurementsystemsindoor/lms200indoor/en/toolboxpar.0006.file.tmp/PILMS22E.pdf)
- [88] " *Sick Product Information - LMS200 / LMS220 2001*"
<http://www.sick.de/de/products/categories/auto/lasermeasurementsystemsindoor/lms200indoor/en/toolboxpar.0006.file.tmp/PILMS22E.pdf>
- [89] **Patric Jensfelt**, " *Approaches to Robot Localization in Indoor Environments*", Royal
Institute of Technology, Estocolmo, Suécia (2001) 91-7583-135-9
- [90] **Maria Isabel Ribeiro, João G. M. Gonçalves**, " *Natural Landmark based Localization of
Mobile Robots Using Laser Range Data*", Proceedings EUROBOT 1996
- [91] **Artur Arsénio e M. Isabel Ribeiro**, " *Absolute Localization of Mobile Robots using
Natural Landmarks*", 5th IEEE International Conference on Electronics,
Circuits and Systems (ICECS'98), Lisbon, September 7-10, 1998
- [92] **João Gomes-Mota e Maria Isabel Ribeiro**, " *Mobile robot localization on reconstructed
3D models*", Journal of Robotics and Autonomous Systems 31 (1-2) pp. 17-30,
Elsevier Science, Abril 2000
- [93] **Michelle Marie Simi**, " *A Real Time Positioning System for Mobile Robots Based on Laser
Triangulation*", Msc. Thesis Northwestern University, Mechanical Engineering ,
EUA (2000)
- [94] **Danaher Motion Särö AB** (formerly NDC) <http://www.ndc.se/>
- [95] <http://www.lazerway.com/>
- [96] " *Lazer Way Equipment Buyers Guide*" http://www.ndc.se/products/pdf/buyers_guide2003.pdf

-
- [97] **R. Want, A. Hopper, V. Falcao, J. Gibbons**, "The active badge location system", ACM Transactions on Information Systems - vol. 10, no. 1, Jan. 1992 91-102 (1992) citeseer.nj.nec.com/want92active.html
- [98] **N. Bulusu, J. Heidemann, D. Estrin**, "GPS-less Low Cost Outdoor Localization For Very Small Devices", IEEE Personal Communications Magazine (2000)
- [99] "RFTechnologies – PinPoint"
<http://www.rftechnologies.com/pinpoint>
<http://www.rftechnologies.com/pinpoint/systems.htm>
- [100] **Jeffrey Hightower, Gaetano Borriello and Roy Want**, "SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength", University of Washington CSE Technical Report 2000-02-02 (2000)
- [101] **Andrew M. Ladd, Kostas E. Bekris, Guillaume Marceau, Algis Rudys, Dan S. Wallach and Lydia E. Kavradi**, "Using Wireless Ethernet for Localization", Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland (2002)
- [102] **Paramvir Bahl and Venkata N. Padmanabhan**, "RADAR: An In-Building {RF}-Based User Location and Tracking System", IEEE Infocom'00, Tel Aviv The Conference on Computer Communications, Volume 2 775-784 (2000) 0-7803-5880-5
<http://citeseer.nj.nec.com/cache/papers/cs/11606/http://zSzzSzwww.research.microsoft.comzSz%7EpadmanabzSzpaperszSzinfocom2000.pdf/bahl00radar.pdf>
- [103] **Paramvir Bahl and Venkata N. Padmanabhan**, "User Location and tracking in an in-Building Radio Network", Technical Report MSR-TR-99-12., Microsoft Research, Redmonton, WA, USA (1999)
- [104] "Comunicado do Governo Português: e-U (Campus Virtuais) é um sucesso a nível internacional" http://www.unic.pcm.gov.pt/UMIC/Media/Sala+de+Imprensa/eu_microsoft.htm
- [105] "e-U (universidade electrónica), Campus Virtual" <http://www.e-u.pt>
- [106] **Grayham French**, "Where You Are", IEEE Spectrum July 2003 pp. 21-25 (2003)
- [107] **Fatima Toor**, "Ultra-wideband (UWB) Technology" <http://cs.smith.edu/~ftoor/uwb.htm>
- [108] **Robert Fleming and Cherie Kushner**, "Integrated Ultra-Wideband Localizers", UWB Conference, 1999, Washington DC, EUA (1999)
http://www.aetherwire.com/Aether_Wire/Integrated_Ultra-Wideband_Localizers.pdf
- [109] "Aether Wire and Location" <http://aetherwire.com/>
- [110] **Kai-Wei Chiang, Aboelmagd Noureldinn Naser El-Sheimy**, "Multisensor integration using neuron computing for land-vehicle navigation", Springer-Verlag 2003 (2002)
- [111] **Maury Wright**, "Time, Position, Velocity? Ask Your GPS", EDN Europe, 3 de Março 1997, pp. 50-66 (1997)
- [112] "Office of the Press Secretary" http://www.ostp.gov/html/0053_2.html
- [113] <http://www.igeb.gov/sa/>
-

- [114] **Hydraulics Lab, Scripps Institution of Oceanography - Center for Coastal Studies**, "GPS STUFF 2001" <http://hydraulicslab.ucsd.edu/gps/garmin1.gif>
- [115] **Alfred Leick**, "DGPS and WADPGS", ACSM Bulletin March/April (1995)
<http://www.spatial.maine.edu/~leick/pub25.htm>
- [116] "USNO NAVSTAR Global Positioning System" <http://tycho.usno.navy.mil/gpsinfo.html>
- [117] "Russian Ministry of Defense - Scientific Information Center"
<http://www.glonass-center.ru/int.html>
- [118] "Ashtec GG24 GPS+Glonass Receiver"
http://products.thalesnavigation.com/assets/techpapers/6_GPSGlonassTech_GG24.pdf
- [119] **James G. Murphy, William V. Cottrell**, "Airborne Testing of GPS+GLONASS Positioning"
http://products.thalesnavigation.com/assets/techpapers/4_AirborneTestingofGPS.pdf
- [120] "A GPS GIS Systems Source GPS-GIS Newsletter"
http://www.rlageosystems.com/gps_news.htm
- [121] **J. Benedicto, S.E.Dinwiddy, G. Gatti, R. Lucas, M. Lugert**, "GALILEO: Satellite System Design and Technology Developments", European Space Agency (2000)
http://ravel.esrin.esa.it/docs/galileo_world_paper_Dec_2000.pdf
- [122] <http://www.mlit.go.jp/koku/ats/e/mts/role/01.html>
- [123] **Hugo Fruehauf**, "WAAS, EGNOS, and WAAS, EGNOS, and MSAS for Telecom MSAS for Telecom Applications Applications", (2001)
http://www.time-frequency.co.uk/appnotes/WAAS_Briefing_12-01.PDF
<http://GPS.FAA.GOV/Programs/WAAS/waas.htm>
- [124] **Dan Cotta**, "Lockheed Martin Selected For Japan Government's Global Navigation Satellite System Program" (1997)
<http://www.missionsystems.lockheedmartin.com/announce/releases/msas2.html>
<http://qbnwww01.qbg.ms.lmco.com/internet/newscntr/releases/>
- [125] **Apostolia Karamali, Andrew Wilson**, "Galileo: The European Programme for Global Navigation Services", European Space Agency Publications Division, ESTEC, PO Box 299, 2200 AG Noordwijk, The Netherlands (2003) ISBN 92-9092-730-5 ISSN 0250-1589 <http://ravel.esrin.esa.it/docs/GalileoBrochure.pdf>
- [126] "ESA Pay-as-you-go motoring just around the corner", ESA portal - 9 de Setembro 2003 http://www.esa.int/export/esaCP/SEMBC5ZO4HD_Benefits_0.html
- [127] "Galileo Presentation" http://europa.eu.int/comm/space/doc_pdf/galileo_presentation.pdf
- [128] **Wolfgang Werner, Theodor Zink, Erwin Löhnert, Jürgen Pielmeier**, "Galileo Integrity Performance Assessment (GIPA)" ION GPS 2001, 11-14 September 2001 1838-1850 (2001) http://www.ifen.com/publications/IONGPS2001_GIPA.pdf
- [129] **Zheng da Wu and James Larkin**, "Localization of mobile devices", Class Notes - INFT13/73 - 336 Mobile Networks and Computing - Chpt 6 (2003)
<http://www.it.bond.edu.au/inft336/>

-
- [130] "Federal Communication Commission - Enhanced 911" (2003)
<http://www.fcc.gov/911/enhanced/>
- [131] **M. Drumheller**, "Mobile Robot Localization Using Sonar", IEEE Transaction on Pattern Analysis and Machine Intelligence, vol 9, n.2 pp 352-332 (1987)
- [132] **O. Vijk, P. Jensfelt and H. Christensen**, "Triangulation based Fusion of Ultrasonic Sensor Data", Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-98), Leuven, Belgium, May 16-20, 1998. IEEE Computer Society, 1998, ISBN 0-7803-4301-8, Volume 4 3419-3424 (1998)
- [133] **R.N. Aguilar, G.C.M. Meijer**, "Low-Cost Ultrasonic Fusion Sensor for Angular Position", Proceedings of SeSens 2002, November 29, 2002 - Veldhoven, the Netherlands (2002) 90-73461-33-2 <http://www.stw.nl/sesens/proc2000/aguilar.pdf>
- [134] **L. Girod, D. Estrin**, "Robust Range Estimation Using Acoustic and Multimodal Sensing", International Conference on Intelligent Robots and Systems (IROS 2001)
- [135] **Ching-Chih Tsai**, "A localization System of a Mobile Robot by Fusing Dead-Reckoning and Ultrasonic Measurements", IEEE Instrumentation and Measurement Technology Conference, Minnesota, USA, 1998
- [136] **O. Wijk and H.I. Christensen**, "Localization and navigation of a mobile robot using natural point landmarks extracted from sonar data", Robotics and Autonomous Systems 31 (2000) pp. 31-42 (2000)
- [137] **Figueroa, F., Mahajan, A**, "A Robust Navigation System for Autonomous Vehicles using Ultrasonics", Control Engineering Practice, Vol. 2, No. 1, 1994. pp. 49-59 (1994)
- [138] **Andy Ward, Alan Jones, Andy Hopper**, "A New Location Technique for the Active Office", IEEE Personnel Communications, Vol 4 N.5, October 1997 pp. 42-47 (1997) <http://citeseer.nj.nec.com/ward97new.html>
- [139] **Greg L. Reid, Evangelos Milios**, "Active Stereo Sound Localization", Technical Report cs-1999-09, York University, Ontario, Canada, 1999
citeseer.nj.nec.com/reid99active.html
- [140] **Gen'ichi Yasuda and Hiroyuki Takai**, "Sensor-Based Path Planning and Intelligent Steering Control of Nonholonomic Mobile Robots", IECON'01: The 27th Annual Conference of the IEEE Industrial Electronics Society (2001)
- [141] **Armando Jorge Sousa, Paulo Gomes da Costa, António Paulo Moreira**, "Variable Resolution System in Mobile Robotics", Proceedings of Controlo 2002, 5th Portuguese Conference on Automatic Control, 2002 pp.489-494 (2002)
- [142] **Joseph A. Muratore**, "Illumination for Machine Vision", Dolan-Jenner Industries
<http://www.pinnaclevision.co.uk/illum02.htm>
- [143] **Lars Stenberg**, "The PSD School", SiTek Electro Optics <http://www.sitek.se/section1.htm>
-

- [144] **Ana Cristina Paiva, Jorge A. Silva, Aurélio Campilho**, "*Aquisição de Informação Tridimensional com Luz Estruturada Codificada*", Actas do 3º Encontro Nacional do Colégio de Engenharia Electrotécnica da Ordem dos Engenheiros pp. 139-146 (1997)
- [145] **StockerYale**, "*What is Structured Light*"
http://www.stockeryale.com/i/lasers/structured_light.htm
- [146] **N. Winters, J. Gaspar, G. Lacey, and J. Santos Victor**, "*Omni-directional vision for robot navigation*", Proc. IEEE Workshop on Omnidirectional Vision, South Carolina, EUA, June 2000 <http://citeseer.nj.nec.com/winters00omnidirectional.htm>
- [147] **C. Marques, P. Lima**, "*A Localization Method for a Soccer Robot Using a Vision-Based Omni-Directional Sensor*", in: P. Stone, T. Balch, G Kraetzschmar (Eds.), RoboCup 2000-Robot Soccer World Cup IV, Springer Verlag, Berlin, D, 2001 (2000)
- [148] **Nuno Gonçalves and Hélder Araújo**, "*Mirror shape recovery from image curves and intrinsic parameters: Rotationally symmetric and conic mirrors*", Proceedings of the Omnivis 2003: Workshop on Omnidirectional Vision and Camera Networks (2003)
- [149] **Pedro Lima, Andrea Bonarini, Carlos Machado, Fabio Marchese, Carlos Marques, Fernando Ribeiro, Domenico Sorrenti**, "*Omni-Directional Catadioptric Vision for Soccer Robots*", Robotics and Autonomous Systems, vol. 36 n. 2--3 (2001) pp. 87—102 (2001) citeseer.nj.nec.com/lima01omnidirectional.html
- [150] **Pedro Lima, Andrea Bonarini, Carlos Machado, Fabio Marchese, Carlos Marques, Fernando Ribeiro, Domenico Sorrenti**, "*Omni-Directional Catadioptric Vision for Soccer Robots*", Robotics and Autonomous Systems, vol. 36 n. 2--3 (2001) pp. 87—102 (2001) <http://citeseer.nj.nec.com/lima01omnidirectional.html>
- [151] **Joshua Gluckman, Member, IEEE Computer Society, and Shree K. Nayar, Senior Member, IEEE**, "*Rectified Catadioptric Stereo Sensors*", IEEE Transactions On Pattern Analysis and Machine Intelligence, Vol. 24, No. 2, Fev 2002 pp. 224-236 (2002)
- [152] **Giuseppina Gini, Francesco Amigoni, Andrea Bonarini, Vincenzo Caglioti, Marco Sornalvico**, "*Self Localization Of Mobile Robots In Indoor Environment*", In Taraglio, S.; Nanni, V. (editors); Enabling Technologies for the PRASSI Autonomous Robot, ENEA Robotics and Information Technology Division pp. 86-97 (2001) <http://citeseer.nj.nec.com/548813.htm>
- [153] **Hans P. Moravec**, "*Robot Spatial Perception by Stereoscopic Vision and 3D Evidence Grids*", CMU-RI-TR-96-34The Robotics Institute - Carnegie Mellon University, EUA - Setembro 1996
- [154] **Shishir Shah, J. K. Aggarwal**, "*Mobile Robot Navigation and Scene Modeling using Stereo Fish-eye Lens System*", Machine Vision and Applications (1997) Vol 10 - Springer - Verlag 1997 pp. 159-173 (1997)

-
- [155] **Shmuel Peleg, Moshe Ben-Ezra e Yael Pritch**, "*Omnistere: Panoramic Stereo Imaging*", IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 23, No. 3, Março 2001 pp. 279-290 (2001)
- [156] **R. Bunschoten and B. Kröse**, "*Range estimation from a pair of omnidirectional images*", Proc. IEEE Int. Conf. on Robotics and Automation, Seoul, Korea (2001) <http://citeseer.nj.nec.com/bunschoten01range.html>
- [157] **Fabien Launay, Akihisa Ohya and Shin'ichi Yuta**, "*Vision-Based Navigation of Mobile Robot using Fluorescent Tubes*", The 10th International Conference on Advanced Robotics (Aug. 2001) <http://www.roboken.esys.tsukuba.ac.jp/~ohya/pdf/RSJ2000-FAB.pdf>
- [158] **Fabien Launay, Akihisa Ohya and Shin'ichi Yuta**, "*A Corridors Lights based Navigation System including Path Definition using a Topologically Corrected Map for Indoor Mobile Robots*", Proceedings 2002 IEEE International Conference on Robotics and Automation, pp.3918-3923 (May 2002) <http://www.roboken.esys.tsukuba.ac.jp/~ohya/pdf/ICRA2002-FAB.pdf>
- [159] **S. Panzieri, F. Pascucci, R. Setola, G Ulivi**, "*A low cost vision based localization system for mobile robot*", MEDSYMP 2001
- [160] **Jon Howel and Keith Kotay**, "*Landmarks for absolute localization*", Technical Report TR2000-364 - Department of Computer Science - Dartmouth College, Hanover, NH 03755-3510, EUA (2000) <ftp://ftp.cs.dartmouth.edu/TR/TR2000-364.pdf>
- [161] **Toyama, K., and Hager, G. D.**, "*Tracker fusion for robustness in visualfeature tracking*", SPIE Photonics East, Philadelphia, PA, 2589 pp. 38-49 (1995) <http://cs-www.cs.yale.edu/homes/toyama/>
- [162] **Sinisa Segvic and Slobodan Ribaric**, "*Determining the Absolute Orientation in a Corridor Using Projective Geometry and Active Vision*", IEEE Transactions on Industrial Electronics, vol. 48, no. 3, june 2001
- [163] **G. Jang, S. Kim, W. Lee, I. Kweon**, "*Color Landmark Based Self-Localisation for Indoor Mobile Robots*", Proc. of the 2002 IEEE International Conference on Robotics & Automation, Washington DC, EUA ,Maio 2002 pp. 1037-1042 (2002)
- [164] **Evgeni Kiriy, Martin Buehler**, "*Three-state Extended Kalman Filter for Mobile Robot Localisation*", Tech. Report Centre for Intelligent Machines (CIM), McGill University, April 12, 2002
- [165] **Prof. Lynne E. Parker William Duncan**, "*Localisation and Exploration*", Class Notes Software for Intelligent Robotics - University of Tennessee - CS594 <http://www.cs.utk.edu/~parker/Courses/CS594-fall02/Lectures/Nov21.pdf>
- [166] **Hugh Durrant-Whyte**, "*Localization, Mapping and the Simultaneous Localisation and Mapping(SLAM) Problem*", SLAM Summer School 2002
-

- [167] **M. Dissanayake, Paul Newman, Steven Clark, Hugh F. Durrant-Whyte e M. Csorba**, "A Solution to the Simultaneous Localisation and Map Building (SLAM) Problem", IEE Transactions on Robotics and Automation, vol. 17, no. 3, Junho 2001
- [168] **Paul Newman**, "On the Structure and Solution of the Simultaneous Localisation and Map Building Problem", PhD Thesis, Australian Centre for Field Robotics, The University of Sydney, Austrália (1999)
- [169] **Michael Csorba**, "Simultaneous Localisation and Map Building", PhD Thesis University of Oxford, Inglaterra (1997)
- [170] **Helga Kolb, Eduardo Fernandez, Ralph Nelson, Bryan William Jones**, "WebVision 2003" <http://webvision.med.utah.edu/index.html>
- [171] **Rafael Gonzalez and Richard Woods**, "Digital Image Processing 2nd Ed.", Prentice Hall - Upper Saddle River NJ 07458 (2002) 0-20-118075-8
- [172] **David Forsyth and Jean Ponce**, "Computer Vision - A Modern Approach", Prentice-Hall, New Jersey, EUA (2003) 0-13-085198-1
- [173] "Color Vision - Spectral Selectivity" <http://www.photo.net/photo/edscott/vis00010.htm>
- [174] **A Gelb, J. Kasper, Raymond Nash, Charles Price, Arthur Sutherland**, "Applied Optimal Estimation" (1989)
0-262-57048-3
- [175] **R. E. Kalman**, "A New Approach to Linear Filtering and Prediction Problems", Transaction of the American Society of Mechanical Engineers —Journal of Basic Engineering, Março 1960 p. 3545 (1960)
- [176] **Eric Wan and Alex Nelson**, "Dual EKF Methods", Kalman Filtering and Neural Networks, Wiley Publishing, editors Simon Haykin, 2001 ISBN:0471369985 citeseer.nj.nec.com/345859.html
- [177] **Greg Welch and Gary Bishop**, "An Introduction to the Kalman Filter", Technical Report 95-041 - Department of Computer Science University of North Carolina at Chapel Hill, EUA (2002)
- [178] **Graham Goodwin and Robert Payne**, "Dynamic System Identification: Experiment Design and Data Analysis", Academic Press, Inc (1997) 0-12-289750-1
- [179] **Manuel Ferreira e Isabel Amaral**, "Probabilidades e estatística - Formulário - 5a Edição", Edições Sílabo (2001) 972-618-251-4
- [180] **R. Steven Rainwater**, "Robot Competition FAQ 2002" <http://robots.net/rcfaq.html>
- [181] "The RoboCup Federation" <http://www.robocup.org/>
- [182] **H Kitano, M Asada, Y. Kuniyoshi, I. Noda, E. Osawa**, "Robocup: The Robot World Cup Initiative", Proceedings of IJCAI'95 Workshop on Entertainment and AI/Alife, pp. 19-24, 1995

- [183] **Pedro Lima, Tucker Balch, Masahiro Fujita, Raul Rojas, Manuela Veloso, Holly Yanco**, "*RoboCup 2001 - Report on Research Issues that surfaced during the competitions and the Conference*", IEEE Robotics & Automation Magazine, Vol9, No 2 June 2002 20-30 (2002)
- [184] "*History Timeline of Robotics*" http://trueforce.com/Articles/Robot_History.htm 07-08-2002
- [185] "*History of robotics and robots*" <http://www.geocities.com/Eureka/7331/hrorobot.htm>
- [186] "*A Brief History of Robotics*" <http://robotics.megaqiant.com/history.html>
- [187] **Raymond Kurzweil**, "*The Age of Intelligent Machines: Chronology*" <http://www.kurzweilai.net/meme/frame.html?main=/articles/art0298.html>
- [188] "*BBC Timeline: Real robots 2001*" http://news.bbc.co.uk/1/hi/in_depth/sci_tech/2001/artificial_intelligence/1531432.stm
- [189] **Robert Bartelds**, "*Real Time Vision Based Self-Localization*" – Technical University of Delft, 4-4-2002

Anexo 1

Acrónimos e Siglas

- AGV – *Automated Guided Vehicle*
- ASIC – *Application Specific Integrated Circuit*
- CAD – *Computer Aided Design*
- CAE – *Computer Aided Engineering*
- CAM – *Computer Aided Manufacturing*
- COTS – *Components-Off-The-Shelf*
- DGPS – *Differential GPS*
- DSP – *Digital Signal Processor*
- EGNOS – *European Geostationary Navigation Overlay Service*
- EKF – *Extended Kalman Filter*
- FDP – *Função Densidade de Probabilidade*
- FEUP – *Faculdade de Engenharia da Universidade do Porto*
- GLONASS – *Global'naya Navigacionnaya Sputnikovaya Sistema – GLobal NAvigation Satellite System*
- GNSS - *Global Navigation Satellite System*

- GPS – *Global Positioning System*
- IFF – *Identify Friend or Foe*
- ILS – *Instrument Landing System*
- IR - *Infra Red*
- ISBN - *International Standard Book Number*
- ISM – *Industrial, Scientific and Medical*
- KF – *Kalman Filter* - Filtro de Kalman
- LIDAR – *Light Detection And Ranging*
- MCU – *Micro-Controller Unit*
- MEMS – *Micro Electromechanical Systems*
- MPU – *Micro Processor Unit*
- MTTF – *Mean time to Failure*
- MTTR – *Mean Time to Repair*
- NAVSTAR(-GPS) - *NAVigation System and Ranging*
- PAN – *Personal Area Network*
- PSD – *Position Sensitive Device*
- PWM – *Pulse Width Modulation*
- RADAR - *Radio Detection And Ranging*
- RF – *Rádio Frequência*
- RT – *Real Time*
- SA – *Selected Availability*
- SLAM – *Simultaneous Localization And Mapping*
- TWI – *Two Wire Interface*
- UNII – *Unlicensed National Information Infrastructure*
- UPS – *Uninterruptible Power Supply*
- USB – *Universal Serial Bus*
- UWB – *Ultra Wide Band*
- VHC – *Very High Camera*
- VOR – *VHF Omni-directional Radio Ranges*
- WADGPS – *Wide Area DGPS*

Anexo 2

Marcos Históricos da Robótica

A informação contida neste anexo foi recolhida a partir das referências [184] [185], [186], [187], [188].

- ~ 400 AC - O filósofo e matemático Archytas constrói uma pomba em madeira que consegue bater as asas e voar.
- ~ 332 AC - Aristóteles avança «Se uma ferramenta quando ordenada ou mesmo por sua própria vontade pudesse fazer o seu trabalho... não seriam necessários aprendizes para os mestres nem escravos para os seus senhores...».
- 1801 Joseph Jacquard inventa uma máquina têxtil comandada por cartões perfurados – um tear programável.
- 1921 Karel Čapek introduz a palavra “Robot” na novela “R.U.R. - Rossum's Universal Robots”. A palavra vem de “robota”, que significa trabalho entediante na língua checa.
- 1926 No filme "Metropolis", de Fritz Lang, "Maria", a fêmea robot do filme, é o primeiro robot a ser projectado na tela de cinema.
- 1936 Alan Turing introduz o conceito de um computador teórico, denominado de Máquina de *Turing*. Além de ser um progresso

- fundamental na lógica do computador, leva também ao aparecimento de novas escolas dentro da matemática.
- 1938 O primeiro robot industrial de pintura é projectado pelos Norte-Americanos Willard Pollard e Harold Roselund para a companhia DeVilbiss.
- 1942 Isaac Asimov publica o seu livro “Runaround”, onde define as três leis fundamentais da robótica.
- 1946 Aparecimento do computador: George Devol regista a patente de uma máquina genérica que repete operações anteriores. O dispositivo de controlo de funcionamento baseia-se em gravação magnética.
- 1954 George Devol projecta o primeiro robot programável e introduz o termo “*Universal Automation*”, plantando a semente do nome da sua futura companhia – Unimation.
- 1968 SRI fabrica o *Shakey*, um robot móvel com capacidade visual, controlado por um computador do tamanho de um quarto.
- 1997 A missão da NASA "*Mars PathFinder*" capta os olhos e imaginação do mundo como *PathFinder* da superfície de Marte e o robot *Sojourner* envia imagens das viagens realizadas no planeta distante.
- 1997 A *Honda* mostra o P3, que é o oitavo protótipo de robot humanóide num projecto iniciado em 1986.
- 1997 Primeira conferência e jogos *RoboCup* em Nagoya, Japão
- 1999 A *Sony* apresenta o animal doméstico robótico, denominado *Aibo*.

Base	Ja-nela	Dig. Reptd	Tama-nho	Tempo Process.	Sequência unívoca
5	5	5	3124	5.397	"0000100002000030000400011000120001300014000210002200023000240003100032000330003400041000420004300044001010010200103001040011100112001130011400121001220012..."
6	5	5	7774	7.52	"0000100002000030000400005000110001200013000140001500021000220002300024000250003100032000330003400035000410004200043000440004500051000520005300054000550010..."
5	6	5	15440 *	15.893	"000010000110000200001200002100003000013000022000031000040000140000230000320000410001010001110001210001310001410002010002110002210002310002400003300042000..."
6	6	5	46374 *	79.274	"00001000011000020000120000210000300001300002200003100004000014000023000032000041000050000150000240000330000420000510001010001110001210001310001410015100..."
4	2	4	17	0	"00102031121322330"
5	2	4	26	0	"00102030411213142232433440"
6	2	4	37	0.01	"0010203040511213141522324253343544550"
4	3	4	66	0.01	"000100200301101201302102202303103203311121131221231321332223233300"
5	3	4	127	0.01	"00010020030040110120130140210220230240310320330340410420430441112113114121231241321331341421431442232242332342432443334344400"
6	3	4	218	0	"0001002003004005011012013014015021022023024025031032033034035041042043044045051052053054055111211311411512212312412513213313413514214314414515215315415522322422523323423524324424525325425533343353443..."
4	4	4	244 *	3.585	"00010002000300011001200130021002200230031003200330101020103011101120113012101220123013101320133020203021102120213022102220223023102320233030311031203130321..."
5	4	4	614 *	3.856	"000100020003000400110012001300140021002200230024003100320033003400410042004300440101020103010401110112011301140121012201230124013101320133013401410142014..."
6	4	4	1282 *	4.146	"0001000200030004000500110012001300140015002100220023002400250031003200330034004004200430044010042004300440050011001200130014001500210022002300240025003100320033003400350041004200430044005005100520053005400550101020103010401050111011201130114011501160117011801190120012100012200..."
4	5	4	940 *	4.366	"00010001100020001200021000300013000220003100101001110012100131002010021102210023000320010200112001220013200202002120022200231003010031100321003300103001..."
5	5	4	2971 *	5.448	"00010001100020001200021000300013000220003100040001400023000320004100101001110012100131001410020100211002210023100240003300042001020011200122001320014200..."
6	5	4	7550 *	7.321	"00010001100020001200021000300013000220003100040001400023000320004100050001500024000330004200051001010011100121001310014100151002010021100221002310024100250..."
4	6	4	3700 *	6.369	"00010001010002000102000201000300010300020200030100100110001101001200011100021000120100130001120002110003100013010020010021010022000121000220100230012200..."
5	6	4	14745 *	14.251	"000100010100020001020002010003000103000202000301000400010400020300030200040100100110001101001200011100021000120100130001120002110003100013010014000113000..."
6	6	4	45001 *	81.588	"00010001010002000102000201000300010300020200030100040001040002030003020004010005000105000204000303000402000501001001100011010012000111000210001201013000..."
3	2	3	10	0	"0010211220"
4	2	3	17	0.01	"00102031121322330"
5	2	3	26	0.01	"00102030411213142232433440"
6	2	3	37	0.01	"0010203040511213141522324253343544550"
3	3	3	26	1.813	"00102212202112101201100200"
4	3	3	58 *	2.994	"0010020030110120130210220230310320331211311221323212313300"
5	3	3	114 *	2.955	"00100200300401101201301402102202302403103203303404104204304411211311412212312413213313414214314422322423324344400"
6	3	3	205 *	2.984	"001002003004005011012013014015021022023024025031032033034035041042043044045051052053054055112113114115122123124125132133134135142143144145152153154155223..."
3	4	3	54 *	3.485	"001001100200120021010112010201221102202022121021122011"
4	4	3	189 *	3.555	"00100110020012002100300130022003101011201020113010301210122012300320131021101320202102210223013302030213023030311023103120313032103220232032311211221131..."
5	4	3	515 *	3.746	"0010011002001200210030013002200310040014002300320041010112010201130103011401040121012201230124003300420131013201410211014202021022102230203021302240133013400430143023102320233024020402140234014402410..."
6	4	3	1126 *	4.075	"0010011002001200210030013002200310040014002300320041005001500240033004200510101120102011301030114010401150105012101220123012401250034004300520131013201410142015102110152020210221022302030213022402040..."
3	5	3	154 *	3.856	"001001010020010200201011001010120011200210012010210102102200121002110022011020112101122002210121101220120201212012210202021021121122110212202212122120220200"

Base	Ja- nela	Dig. Reptd	Tama- nho	Tempo Process.	Sequência unívoca
5	2	1	26	0.01	"00102030411213142232433440"
6	2	1	37	0	"0010203040511213141522324253343544550"
2	3	1	10	0.01	"0001011100"
3	3	1	29	0	"00010020110120210221112122200"
4	3	1	66	0	"000100200301101201302102202303103203311121131221231321332223233300"
5	3	1	127	0	"0001002003004011012013014021022023024031032033034041042043044111211311412 21231241321331341421431442232242332342432443334344400"
6	3	1	218	0	"0001002003004005011012013014015021022023024025031032033034035041042043044 04505105205305405511121131141151221231241251321331341351421431441451521531 5415522..."
2	4	1	19	0	"0000100110101111000"
3	4	1	84	0	"0000100020011001200210022010102011101120121012202021102120221022211112112 21212222000"
4	4	1	259	0.01	"0000100020003001100120013002100220023003100320033010102010301110112011301 21012201230131013201330202030211021202130221022202230231023202330303110312 031303..."
5	4	1	628	0	"0000100020003000400110012001300140021002200230024003100320033003400410042 00430044010102010301040111011201130114012101220123012401310132013301340141 0142014301440202030204021102120213021402..."
6	4	1	1299	0	"0000100020003000400050011001200130014001500210022002300240025003100320033 00340035004100420043004400450051005200530054005501010201030104010501110112 01130114011501210122012301240125013101320..."
2	5	1	36	0	"00000100011001010011101011011110000"
3	5	1	247	0	"0000010000200011000120002100022001010010200111001120012100122002010020200 21100212002210022201011010120102101022011020111101112011210112201202012110 1212012..."
4	5	1	1028	0	"0000010000200003000110001200013000210002200023000310003200033001010010200 10300111001120011300121001220012300131001320013300201002020020300211002120 021300..."
5	5	1	3129	0.02	"0000010000200003000040001100012000130001400021000220002300024000310003200 03300034000410004200043000440010100102001030010400111001120011300114001210 0122001..."
6	5	1	7780	0.03	"0000010000200003000040000500011000120001300014000150002100022000230002400 02500031000320003300034000350004100042000430004400045000510005200053000540 0055001..."
2	6	1	69	0	"00000010000110001010001110010010110011010011110101011101101111100000"
3	6	1	734	0.01	"0000001000002000011000012000021000022000101000102000111000112000121000122 00020100020200021100021200022100022200100100200101100101200102100102200110 100110..."
4	6	1	4101	0.03	"0000001000002000003000011000012000013000021000022000023000031000032000033 00010100010200010300011100011200011300012100012200012300013100013200013300 0201000..."
5	6	1	15630	0.07	"0000001000002000003000004000011000012000013000014000021000022000023000024 00003100003200003300003400004100004200004300004400010100010200010300010400 0111000..."
6	6	1	46661	0.251	"0000001000002000003000004000005000011000012000013000014000015000021000022 00002300002400002500003100003200003300003400003500004100004200004300004400 0045000..."

Anexo 4

Ficha Técnica

*Escrito em Open Office 1.0.3.1 sob Windows com sm641mi.dll corrigida
Fontes: Arrus BT, Dauphin , Verdana e Blue Highway
Muitas figuras desenhadas utilizando Corel Draw 10*

Number of Pages:	230
Number of Tables:	11
Number of Graphics:	174
Number of OLE Objects:	293
Number of Paragraphs:	4008
Number of Words:	47717
Number of Characters:	313310
Number of Lines:	7263

URL da página do Autor: <http://www.fe.up.pt/~asousa>
URL da página da equipa de Futebol Robótico da FEUP: <http://www.fe.up.pt/~robosoc>

*Armando Jorge Miranda de Sousa,
1 Setembro 2003*