

Localização em Tempo Real de Múltiplos Robots num Ambiente Dinâmico

Paulo José Cerqueira Gomes da Costa

UNIVERSIDADE DO PORTO

Faculdade de Engenharia

Departamento de Engenharia Electrotécnica e de Computadores

1999

Paulo José Cerqueira Gomes da Costa

**Localização em Tempo Real de Múltiplos Robots
num Ambiente Dinâmico**

Dissertação para Doutoramento em Engenharia Electrotécnica e de Computadores
na Faculdade de Engenharia da Universidade do Porto

Tese realizada sob a orientação do
Doutor Jorge Leite Martins de Carvalho
Professor Catedrático

UNIVERSIDADE DO PORTO
Faculdade de Engenharia
Departamento de Engenharia Electrotécnica e de Computadores
1999

⁰O autor foi, durante a realização desta Tese, bolsiro do PRODEP

aos meus Pais e à Joana

Agradecimentos

Ao meu orientador, o Professor Doutor Jorge Leite Martins de Carvalho, pela confiança, disponibilidade e apoio demonstrados.

Ao Paulo Marques pelo apoio no desenvolvimento e implementação da maior parte do software.

Ao Armando Sousa e ao resto da equipa de Futebol Robótico da FEUP: António Paulo, Pedro Costa e Susana Gaio cujo esforço foi crucial para o sucesso do projecto. Este agradecimento estende-se a todos os que contribuíram de muitas maneiras na construção da equipa.

A todos os meus colegas da FEUP que apoiaram e incentivaram este projecto.

À Joana que, entre muitas outras coisas, suportou os efeitos menos bons do meu esforço.

À minha família que me apoiou serenamente.

¹ Trough it all, silent witness to his torment, was the Hunter's gift. The ultimate temptation. Not power, but something far more subtle. Not sorcery, but something even richer. Knowledge.

C. S. Friedman

¹Durante tudo isto, testemunha silenciosa do seu tormento, estava a oferenda do Caçador. A tentação final. Não poder, mas algo muito mais subtil. Não magia, mas algo ainda mais rico. Conhecimento.

Resumo

Para que seja possível posicionar e manobrar um robot ou um grupo de robots num ambiente cujos obstáculos e objectivos seguem uma dinâmica própria, é necessário dispor de capacidades sensoriais capazes de extrair a informação necessária. Dada a dinâmica inerente à informação a recolher, esse processo deve ser efectuado em tempo real, o que requer o desenvolvimento de técnicas e de algoritmos capazes de actuarem com restrições temporais apertadas. Finalmente é importante que o sistema sensorial apresente uma série de características tais como a robustez à operação em condições subóptimas e seja ao mesmo tempo capaz de fornecer não só uma estimativa das grandezas observadas mas também da certeza associada a essa estimativa.

O futebol robótico fornece uma plataforma onde estes problemas são propostos, num ambiente standard, o que permite a comparação e análise das diferentes técnicas propostas e implementadas pela comunidade científica.

Desenvolvemos, implementámos e fundamentámos uma série de algoritmos capazes de, recorrendo a um esquema de visão, extrair a localização e estimar a velocidade dos robots e da bola envolvidos num desafio de futebol robótico. Este é um passo essencial para estabelecer um esquema de decisão e controlo capaz de possibilitar o preenchimento dos objectivos da equipa robótica.

O sistema proposto, recorrendo a uma classificação baseada em operações sobre conjuntos *fuzzy*, seguida de uma filtragem em que se recorre a um filtro de Kalman, mostrou dispor de características de robustez, versatilidade e rapidez.

Neste trabalho descrevemos e fundamentamos a nossa abordagem e apresentamos ainda uma série de exemplos onde ilustramos alguma das suas propriedades mais interessantes.

Abstract

To be possible to position and to maneuver a robot or a robot group in an environment whose obstacles and objectives follow their own dynamics, it is necessary to dispose of sensorial capabilities which are able to extract the necessary information. Given the dynamical properties of the information, that process should be completed in real time. That requires the development of techniques and algorithms capable of fulfilling the timing restrictions. Finally, it is important that the sensorial system presents a series of characteristics related with the robustness of operation in suboptimal conditions. It should also supply not only an estimate of the observed variables but also the certainty associated with that estimate.

The robotic soccer challenge supplies a platform where these problems are proposed in a standard environment. That allows the comparison and analysis of the proposed techniques and implementations by the scientific community.

Using vision as our sensorial input, we developed and implemented a series of algorithms capable of extracting the location and estimate the speed of the robots and the ball, during a game of robotic soccer. This is an essential step for establishing the decision and control strategies that fulfill the objectives of the robotics team.

The proposed system, based on a classification scheme supported by a sequence of operations on fuzzy sets and an extra filter step implemented by an Extended Kalman Filter, showed robustness, versatility and speed.

In this work we describe and fundament our approach and provide examples that illustrate some of its most interesting properties.

Résumé

Pour qu'il soit possible de positionner et de manier un robot ou un groupe de robots dans une ambiance dont les obstacles suivent une dynamique propre, il est nécessaire de développer des capacités sensorielles capables d'extraire l'information nécessaire. Étant donnée la dynamique inhérente à l'information à recueillir, ce procédé doit être effectué en temps réel, ce qui exige le développement de techniques et d'algorithmes capables de fonctionner avec des restrictions temporelles serrées. Finalement, il est important que le système sensoriel présente une série de caractéristiques rapportées avec la robustesse à l'opération en condition "suboptimales" ainsi qu'il soit capable de fournir non seulement un calcul des grandeurs observées mais aussi de la certitude associée à ce calcul.

Le football robotique fournit une plate-forme où ces problèmes sont proposés, dans une ambiance standard, ce qui permet la comparaison et l'analyse des différentes techniques proposées et exécutées par la communauté scientifique.

Nous développons, nous exécutons et nous fondons une série d'algorithmes capables de, en recourant à un schéma de vision, extraire la localisation et estimer (évaluer) la vitesse des robots et de la balle engagés dans un match de football robotique. Celui-ci est un pas essentiel pour un schéma de décision et de contrôle capable de rendre possible le remplissage des objectifs de l'équipe robotique.

Le système proposé, en recourant à une classification basée sur des opérations sur ensemble "fuzzy" suivie d'un filtrage dont on se sert d'un filtre de Kalman, a montré de posséder des caractéristiques de robustesse, de polyvalence et de rapidité.

Dans ce travail, nous décrivons et nous fondons notre approche et nous présentons encore une série d'exemples où nous illustrons quelques-unes de ses propriétés les plus intéressantes.

Índice

Lista de Figuras	14
Lista de Tabelas	16
1 Introdução	17
1.1 Contexto e Motivação	17
1.2 Objectivos e Contribuições	18
1.3 Estrutura da Tese	18
2 Definição do Problema	20
3 Conjuntos <i>Fuzzy</i>	24
3.1 Definição e Propriedades de Conjuntos <i>Fuzzy</i>	24
3.2 Operações com Conjuntos <i>Fuzzy</i>	26
3.2.1 Intersecção	27
3.2.2 Reunião	27
3.2.3 Complemento	28
3.2.4 Subconjuntos	28
3.3 Cardinal, Entropia e outros Conceitos	28
3.3.1 Distância de dois conjuntos	28
3.3.2 Cardinal	29
3.3.3 Entropia	29
3.3.4 Teorema da Entropia <i>Fuzzy</i>	30
3.4 Proposições <i>Fuzzy</i>	30
3.4.1 Extensão <i>Fuzzy</i> do Conceito de Subconjunto	30
3.5 <i>Fuzzy</i> versus Probabilidade	31
4 Processos Estocásticos e o Filtro de Kalman	35
4.1 Variáveis Aleatórias e Processos Estocásticos	35
4.2 O Filtro de Kalman	40
4.2.1 Filtro de Kalman discreto	42
4.2.2 Filtro de Kalman Extendido	45
5 Sistema de Interpretação de Imagem	48
5.1 Dos Pixeis Para o Estado do Sistema	51
5.2 Sistema de Coordenadas	52

5.3	Estado do Sistema	53
5.4	Mapeamento Imagem - Mundo	55
5.4.1	A Definição da Escala	57
5.5	Correcção da paralaxe	57
5.6	Classificação das Cores	59
5.7	Estimação dos Conjuntos Associados aos Marcadores	60
5.7.1	Construção do Histograma	61
5.7.2	Estimação de um Conjunto Associado à Cor	62
5.7.3	Medida da Qualidade da Definição dos Conjuntos Associados às Cores	63
5.8	Seleccção da Área Activa	67
5.9	Classificação e Ordenação	69
5.10	Agregação e Afecção dos Pixeis a Marcadores	71
5.11	Qualidade de um Marcador	72
5.12	Afecção Marcador-Robot e Marcador-Bola	74
5.13	Marcadores Adicionais e Correspondente Validação das Observações	74
5.13.1	Algumas Definições Úteis	75
5.13.2	Versão 98	76
5.13.3	Versão 99	79
6	Filtragem da Medida do Estado da Bola	82
6.1	Dinâmica da bola	82
6.2	Identificação dos Parâmetros do Modelo	86
6.3	Filtro de Kalman	87
7	Filtragem da Medida do Estado dos Robots	89
7.1	Dinâmica do Robot	89
7.1.1	Robots Controlados	90
7.1.2	Robots Adversários	93
8	Implementação e Resultados	95
8.1	Optimizar?	95
8.1.1	Profundidade de cor	96
8.1.2	Implementação do algoritmo de Classificação	96
8.2	Resultados	98
8.2.1	Caso 1 - Uma Bola Parada	101
8.2.2	Caso 2 - A Bola em Movimento	106
8.2.3	Caso 3 - O Robot 1 Estacionário	111
8.2.4	Caso 4 - O Robot 2 em Movimento	115

9	Conclusões	120
9.1	Trabalho Desenvolvido	120
9.2	Trabalho Futuro	121
9.2.1	Múltiplas Câmaras	121
9.2.2	Generalização do Algoritmo de Visão	122
9.2.3	Processamento Extra sobre a Imagem da Bola	123
9.2.4	Estudo da Robustez e Optimização do Estimador da Posição de um Marcador	123
9.2.5	Introdução de um Mecanismo de Adaptatividade no Mapeamento das Cores	124
	Bibliografia	124
A	Descrição do Hardware	128
B	Desenvolvimento do Software	129
B.1	Ferramentas de Desenvolvimento	129
B.2	Sistema Desenvolvido	129

Lista de Figuras

2.1	Arquitetura Típica do Sistema	22
3.1	Conjunto <i>fuzzy</i> com a representação gráfica do seu núcleo, suporte e corte-0.4	26
3.2	Um quadrado?	33
3.3	Quão vermelho é o <i>vermelho</i> 2?	34
5.1	O sistema de captura de imagem	49
5.2	O erro de Paralaxe para alturas da câmara diferentes	49
5.3	O Problema da Ocultação da Bola - Na vertical da câmara	50
5.4	O Problema da Ocultação da Bola - A 60 <i>cm</i> da vertical da câmara	50
5.5	O campo com o respectivo sistema de coordenadas	52
5.6	A imagem capturada pela câmara colocada sobre campo	55
5.7	Os pontos de calibração do mapeamento de coordenadas	57
5.8	A compensação da distorsão em barril	58
5.9	O Espaço <i>rgb</i> e algumas regiões desse espaço	59
5.10	Representação dos Conjuntos F_0, \dots, F_3	64
5.11	Representação dos Conjuntos F_0, \dots, F_3 segundo outro ângulo	65
5.12	Representação dos Conjuntos F_0, \dots, F_3 evidenciando a sepa- ração amarelo-laranja	66
5.13	Imagem Capturada completa	68
5.14	Parte Activa da Imagem Capturada	69
5.15	Sequências de Processamento	70
5.16	Esquema dos marcadores - Versão 98	76
5.17	As configurações dos marcadores adicionais - Versão 98	77
5.18	Ambiguidade no emparelhamento dos marcadores - Caso A	77
5.19	Ambiguidade no emparelhamento dos marcadores - Caso B	78
5.20	A equipa de 98 durante um treino em Paris	80
5.21	Esquema dos marcadores - Versão 99	81
5.22	As configurações dos marcadores adicionais - Versão 99	81
7.1	Um Robot e respectivas coordenadas	91
7.2	O Controlador Local ao Robot	92

8.1	Os conjuntos $F_0...F_3$ usados nos exemplos seguintes	98
8.2	Os conjuntos segundo outro ângulo	99
8.3	Os conjuntos segundo uma vista de topo	100
8.4	Caso 1 - Posição observada e estimada da bola	102
8.5	Caso 1 - Trajectória observada e estimada da bola	102
8.6	Caso 1 - Histograma da distribuição de x	103
8.7	Caso 1 - Histograma da distribuição de y	103
8.8	Caso 1 - Velocidades estimadas da bola	104
8.9	Caso 1 - Cardinal de $M(G_{bola}c_0)$	105
8.10	Caso 2 - Histograma da distribuição de x (instantes iniciais) .	107
8.11	Caso 2 - Histograma da distribuição de y (instantes iniciais) .	107
8.12	Caso 2 - Posição observada e estimada da bola	108
8.13	Caso 2 - Trajectória observada e estimada da bola	108
8.14	Caso 2 - Velocidade estimada segundo x e estimativa da res- pectiva covariância	109
8.15	Caso 2 - Velocidade estimada segundo y e estimativa da res- pectiva covariância	109
8.16	Caso 2 - Cardinal de $M(G_{bola}c_0)$	110
8.17	Caso 2 - Velocidades e posições estimadas	110
8.18	Caso 3 - Posição observada e estimada do Robot 1	112
8.19	Caso 3 - Trajectória observada e estimada do Robot 1	112
8.20	Caso 3 - Histograma da distribuição de x	113
8.21	Caso 3 - Histograma da distribuição de y	113
8.22	Caso 3 - Histograma da distribuição da atitude	114
8.23	Caso 3 - $M(G_{mp}c_1)$	114
8.24	Caso 4 - Posição observada e estimada do Robot 2	115
8.25	Caso 4 - Trajectória observada e estimada do Robot 2	116
8.26	Caso 4 - $M(G_{mx}c_2)$	116
8.27	Caso 4 - Posição observada e estimada segundo x	117
8.28	Caso 4 - Posição observada e estimada segundo y	117
8.29	Caso 4 - Atitude medida e estimada do Robot 2	118
8.30	Caso 4 - Estimativa da velocidade tangencial e respectiva co- variância	118
8.31	Caso 4 - Estimativa da velocidade tangencial e respectiva co- variância	119

Lista de Tabelas

8.1	Caso 1 - Estatísticas da Posição xy	101
8.2	Caso 1 - Qualidade das observações	101
8.3	Caso 1 - Estatísticas da Velocidade	104
8.4	Caso 2 - Qualidade das observações	106
8.5	Caso 3 - Estatísticas da Posição xy	111
8.6	Caso 3 - Qualidade das observações	111
8.7	Caso 4 - Qualidade das observações	115

Capítulo 1

Introdução

1.1 Contexto e Motivação

Uma das componentes mais importantes em robótica é a componente sensorial. Quando referimos a componente sensorial não estamos apenas a considerar a construção e disposição dos sensores mas também aquisição e tratamento dos sinais disponíveis de modo a extrair a informação que procuramos. Muitas vezes é este último passo que pode barrar o caminho à obtenção de resultado viáveis.

Nos últimos anos assistiu-se a um crescente interesse no recurso a sistemas com visão como um meio sensorial primário e autónomo. Vários factores se conjugaram para esse fenómeno:

- A evolução no poder de computação e de armazenamento começou a tornar viável o processamento de imagens a cores e com resoluções razoáveis.
- A versatilidade deste tipo de sensores, pois a partir de uma imagem é possível a extracção de medidas que requereriam uma grande quantidade de sensores de outro tipo.
- A progressiva disponibilidade de câmaras e sistemas de aquisição a preços significativamente baixos.
- Uma câmara pode mostrar-se um sensor essencialmente não intrusivo e pode apresentar uma maior versatilidade na sua colocação.
- Para uma série de problemas, por exemplo a navegação, é o método preferido pela maioria dos sistemas biológicos. Por vezes, os muitos milhões de anos de evolução indicam-nos soluções muito elegantes.

Por todos estes factores, o recurso a sistemas baseados em visão para o controlo em robótica tem motivado crescente investigação nesta área.

1.2 Objectivos e Contribuições

Procurámos atacar o problema de localizar múltiplos robots assim como o de estimar certos parâmetros do ambiente recorrendo a um sistema de visão. Se, pelo menos local e temporariamente, o problema da localização de um robot pode ser resolvido recorrendo a sensores locais já a localização de objectos externos se torna muito mais complexa. O recurso a uma ou mais câmaras externas que localizam os itens de interesse é uma solução cada vez mais viável e que nos pareceu extremamente interessante quer do ponto de vista teórico quer prático.

Recorrendo ao problema do futebol robótico, pudemos projectar, implementar e testar um sistema que ataca um conjunto de questões num ambiente que já se tornou um padrão. Assim a comparação do desempenho e das áreas a explorar e melhorar foi significativamente melhorada dada a existência de uma comunidade científica que ataca o problema sob condições semelhantes. Consequentemente, desenvolvemos um sistema que permite seguir, em tempo real, a posição de uma variedade de marcadores. Este problema torna-se consideravelmente mais difícil devido ao requisito de tempo real. As restrições temporais excluem, actualmente, o recurso a técnicas de processamento de imagem mais complexas.

O sistema proposto permite, graças a uma classificação *fuzzy* dos pixels da imagem, reduzir consideravelmente a dimensão dos dados a processar, para extrair a informação necessária para o cálculo da posição dos marcadores.

Este sistema permite também extrair uma medida da incerteza associada à estimativa da posição dos marcadores.

O esquema de calibração apresentado, permite ainda um ajuste incremental que facilita o refinamento progressivo da qualidade da classificação. Torna-se possível, interactivamente, verificar e ajustar a calibração que está a ser efectuada.

O recurso a um esquema de filtragem baseado num filtro de Kalman permite alcançar um nível extra de precisão pois permite introduzir conceitos de memória e de plausibilidade na nossa estimativa.

1.3 Estrutura da Tese

No capítulo 2 apresentamos o problema, a sua motivação e discutimos as características que o distinguem. São também apresentadas e discutidos os requisitos a impor a este sistema de visão. Indicamos também quais as justificações para esses requisitos.

Nos capítulos 3 e 4 apresentamos sucintamente as ideias base relativas às ferramentas teóricas empregues. Temos assim no capítulo 3 a definição de conjuntos fuzzy e a apresentação das operações que iremos considerar.

Introduzimos também a operação *corte- α fuzzy*. No capítulo 4, depois de uma breve revisão dos conceitos de probabilidade, variáveis aleatórias e processos estocásticos, apresentamos uma versão em tempo discreto do filtro de Kalman e uma versão mista, isto é recorrendo a um modelo contínuo mas com observações em tempo discreto, do Filtro de Kalman Extendido. Estes capítulos permitem introduzir a notação e realçar os pressupostos necessários.

O capítulo 5 apresenta e justifica, segundo uma perspectiva teórica, a sequência de algoritmos que processam a imagem do campo e extraem a localização dos robots e da bola. Também é apresentado o método de calibração deste sistema e são discutidas as várias alternativas empregues para identificar cada um dos robots. São apresentadas métricas que permitem caracterizar a calibração e operação do sistema.

Nos capítulos 6 e 7 apresentamos os modelos, a sua estimação e as considerações necessárias ao recurso a uma filtragem que refina os resultados do sistema de visão. Como esse sistema fornece não só uma observação das posições como uma indicação da incerteza associada, podemos introduzir essa informação no Filtro de Kalman. Perante situações em que temos observações de muito baixa qualidade podemos recorrer à informação filtrada para evitar uma degradação acentuada da qualidade da nossa estimativa do estado. No capítulo 6 recorreremos a um filtro de Kalman em tempo discreto tal como foi apresentado no capítulo 4. Já no capítulo 7, dada a não linearidade do modelo associado aos robots apresentamos o recurso a um Filtro de Kalman Extendido. No fim destes capítulos expomos os resultados da aplicação deste esquema de filtragem para alguns casos.

No capítulo 8 discutimos certos pormenores de implementação que permitem a operação eficiente dos algoritmos apresentados. As operações mais críticas são analisadas detalhadamente e os esquemas de representação mais eficientes dos conjuntos *fuzzy* são apresentados. São também analisados certos parâmetros do sistema que podem ser ajustados para obter um balançamento correcto entre a precisão e a velocidade de execução. São também apresentados os resultados do sistema para uma variedade de situações de modo a evidenciar as propriedades mais interessantes do nosso sistema.

Capítulo 2

Definição do Problema

A investigação em robótica móvel e na generalidade dos problemas associados: mecânica, controlo, localização e todo o problema sensorial, decisão e interacção com o ambiente, encontra como obstáculo a falta de *benchmarks* adequados à análise dos progressos efectuados. Dada a multiplicidade de configurações e possíveis definições das tarefas a efectuar por um ou mais robots, torna-se por vezes difícil comparar os resultados de diferentes investigadores e avaliar a relevância de diferentes abordagens. Daí o sucesso do RoboCup. O esforço da *RoboCup Federation* de criar e regular uma série de encontros em que equipas de robots se defrontam realizando partidas de Futebol, tem vindo a ser correspondido por uma adesão crescente de investigadores. Ao introduzir Ligas onde as dimensões e outras características das equipas de robots estão bem definidas, permite que as soluções e as técnicas empregues sejam mais facilmente comparadas. Mais ainda, a promoção de encontros anuais permite um contacto frequente entre as equipas que desenvolvem os robots facilitando a troca de experiência e soluções. Nesta tese abordaremos apenas problemas que foram tratados durante a construção e evolução da equipa de robots que participou nos campeonatos mundiais do RoboCup em 1998 e 1999 na Liga F-180. Tal como nas outras ligas, não há intervenção humana durante o jogo, conseqüentemente, os robots devem ser capazes de evoluir em campo de uma forma perfeitamente autónoma. Apesar das regras[25] estarem em evolução constante, podemos listar as principais características desta liga:

- Uma equipa pode ter até cinco robots em campo apresentando cada robot uma área que não deve exceder 180 cm^2 .
- A maior diagonal de um robot não deve exceder 18 cm .
- O campo de jogos tem as dimensões duma superfície de Ténis de Mesa e as balizas 50 cm de largura.
- Joga-se com uma bola de Golfe côr de laranja.

- O campo está rodeado de paredes com 10 *cm* de altura.
- Cada robot deve apresentar uma bola de Ténis de Mesa colocada no seu topo para identificação da equipa a que pertence. Uma das equipas usa bolas amarelas enquanto que a outra recorre a bolas azuis.
- De um modo geral, o contacto entre robots é penalizado.
- Os robots podem comunicar com um ou mais computadores externos que, naturalmente, também devem operar sem intervenção humana. Nota: esta regra é válida enquanto o jogo decorre, durante as paragens é admitida a operação manual até ao reinício do jogo.
- Cada jogo consiste em duas partes de 10 min cada.

Estas características levam quase intuitivamente ao uso de sistemas externos de visão para localizar os robots e a bola. Temos assim o recurso a uma ou mais câmaras que, colocadas sobre o campo, possibilitam a localização da bola e dos robots. Com o uso de marcadores adicionais é possível extrair também informação acerca da atitude dos robots.

Este problema apresenta uma série de particularidades que, face aos problemas de visão mais tradicionais, por um lado o simplificam mas por outro trazem dificuldades extra. Há factos que diminuem a sua complexidade:

- A câmara pode estar fixa e perpendicular ao plano de jogo.
- As condições de iluminação podem ser mantidas extremamente estáveis.
- São conhecidas a forma e a cor dos marcadores. Estas não se alteram ao longo de um jogo.
- As diferentes cores dos marcadores são suficientes para obter grande parte da informação necessária sem que seja necessário a identificação de padrões.

Porém, há um factor que traz dificuldades acrescidas. O processamento da imagem deve ser feito num espaço de tempo extremamente apertado. A maior parte das técnicas mais comuns tipicamente revelam um peso computacional excessivo, para o tempo disponível. Isto porque, dado o carácter extremamente dinâmico do jogo, rapidamente a informação acerca das posições se torna obsoleta. Não é viável o recurso a imagens de resolução demasiado baixa pois os marcadores e a própria bola são relativamente pequenos face à área que deve ser observada. Certas equipas com sistemas de visão que demoravam tempos na ordem dos 100 *ms* tiveram a desagradável surpresa de ter, numa imagem, a bola à frente do robot que servia de guarda redes e de na imagem seguinte não encontrarem a bola pois ela já estava

dentro da baliza. Se queremos lidar com robots e bolas a moverem-se a velocidades da ordem de $1-2 \text{ m.s}^{-1}$ é importante ter tempos de processamento, pelo menos, da ordem¹ de 20 ms .

Outra questão delicada é o papel crucial que este sistema tem sobre o desempenho de todo o sistema de decisão e controlo da equipa. Se por algum motivo a informação disponibilizada pelo sistema de visão não apresenta a precisão requerida, rapidamente teremos a equipa de robots incapaz de funcionar adequadamente. Não conhecer a localização da bola ou dos robots é, naturalmente, completamente impeditivo de esboçar qualquer acção útil.

O problema a resolver é o de conseguir um sistema de visão capaz de, em tempo real, obter a posição e atitude dos robots da nossa equipa, a posição dos robots da equipa adversária e a posição da bola. Será também conveniente ter um mecanismo de avaliar e validar essa informação assim como alguma espécie de filtro capaz de a completar. Este aspecto é muito importante pois um filtro correctamente implementado permite extrair informação que é crucial para os níveis de decisão e controlo dos robots. Mais concretamente, uma estimativa da velocidade da bola, que geralmente não pode ser extraída de uma imagem parada, é crítica para a decisão das acções a pedir à equipa. Também importante, para o desempenho do sistema de controlo, é o conhecimento da velocidade dos robots.

A arquitetura típica para resolver este problema está presente na figura 2.1.

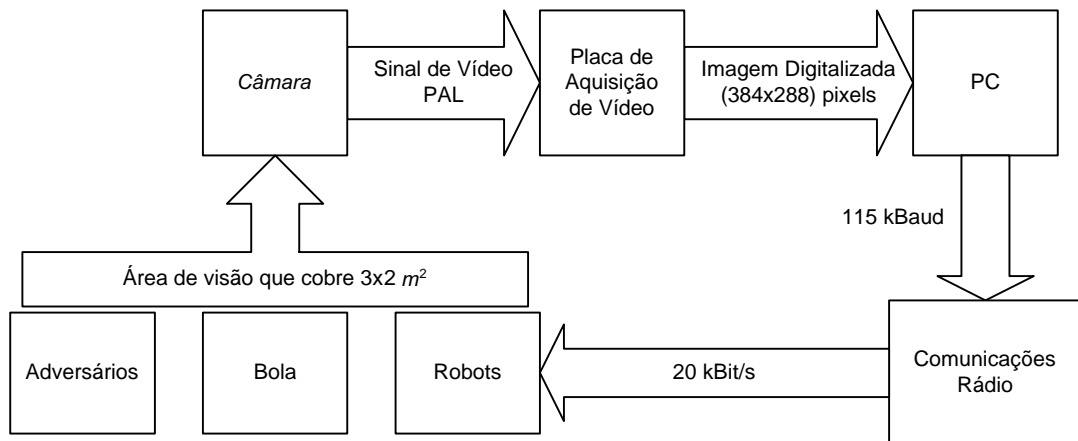


Figura 2.1: Arquitetura Típica do Sistema

Naturalmente, iremos debruçar-nos essencialmente sobre a componente sensorial que, embora represente apenas uma parte do problema global, é

¹Para uma bola a deslocar-se a 2 m.s^{-1} e com um tempo de processamento de 20 ms é fácil verificar que, quando o controlo chega ao robot, a bola já pode estar a 4 cm do ponto onde foi avistada.

uma componente crucial para o seu funcionamento.

Capítulo 3

Conjuntos *Fuzzy*

[21] **fuzzy** [-i] adj. ondulado, frisado, crespo (cabelo); impreciso, indistinto; com algodão; esfiado; com defeito ocasionado pela utilização de fibras partidas no fio têxtil, o que origina uma aparência felpuda.

3.1 Definição e Propriedades de Conjuntos *Fuzzy*

Neste capítulo apresentamos as noções fundamentais associadas a conjuntos *fuzzy*. Considera-se Zadeh[30] a referência original no que toca à teoria relativa a conjuntos *fuzzy*. Desde então muitos outros autores descreveram e expandiram essa teoria. Neste capítulo, seguiremos a notação e as ideias apresentadas em [16]. Podemos considerar a teoria de conjuntos *fuzzy* uma generalização da teoria existente para conjuntos clássicos. Partindo de um universo U , podemos representar um conjunto clássico como um vector, com tantas componentes como elementos de U , onde cada componente pode tomar valores no conjunto $\{0, 1\}$. Naturalmente, o valor zero indica que o elemento do universo correspondente não pertence ao conjunto, enquanto um 1 indica que o elemento pertence ao conjunto. Para um conjunto *fuzzy* admitimos que o grau de pertença de um elemento a um conjunto pode ser superior a zero mas ao mesmo tempo inferior a 1. Assim, o vector que define o conjunto passará a ter as suas componentes a tomar valores no intervalo $[0, 1]$.

Estamos a considerar a dualidade conjunto-vector assumindo que está evidente a associação elemento do universo com a respectiva coordenada do vector. Esta associação é clara para o caso de universos numeráveis onde esteja bem definida uma ordenação já que se pode fazer implicitamente corresponder o i -ésimo elemento do universo com a i -ésima componente do vector. Já para os outros casos será mais correcto definir um conjunto A recorrendo a uma função m_A que indica o grau de pertença de cada elemento

x do universo U ao conjunto A :

$$A = \{(x, m_A(x)) : x \in U, m_A(x) : U \rightarrow [0, 1]\} \quad (3.1)$$

Porém, nos casos em que é possível empregar o conceito de vector para representar o conjunto *fuzzy* isso permite-nos empregar certos conceito geométricos para visualizar algumas das suas operações e propriedades. Por exemplo, tomemos todos os subconjuntos de U , no caso clássico. Sendo vectores cujas componentes são zero ou um, apontam sempre para um dos vértices do hipercubo unitário $I^n = [0, 1]^n$, onde n é o número de elementos de U . Já no caso em que temos os subconjuntos *fuzzy* de U , o que obtemos é todo o hipercubo unitário.

Um conjunto *fuzzy* pode ser caracterizado por uma série de propriedades que não se encontram definidas para conjuntos clássicos pois não são muito interessantes quando aplicadas a essa classe de conjuntos. No entanto, quando usadas em conjuntos *fuzzy* servem de base à extensão de certos conceitos tais como a convexidade de um conjunto. Para um conjunto *fuzzy* A definido por uma função m_A temos:

A *altura* de um conjunto *fuzzy* define-se como:

$$height(A) = \sup_{x \in U} m_A(x) \quad (3.2)$$

Um conjunto *fuzzy* com altura igual a um diz-se *normal*. Qualquer conjunto clássico, com a excepção do conjunto vazio ϕ , tem altura igual a um.

O *núcleo* de um conjunto *fuzzy* é o conjunto clássico definido por:

$$core(A) = \{x \in U : m_A(x) = 1\} \quad (3.3)$$

O *suporte* de um conjunto *fuzzy* é o conjunto clássico tal que:

$$support(A) = \{x \in U : m_A(x) > 0\} \quad (3.4)$$

Podemos, generalizando os conceitos de núcleo e de suporte definir o *corte- α* :

$$ac(A, \alpha) = \{x \in U : m_A(x) \geq \alpha\} \quad (3.5)$$

e a versão com a desigualdade estrita, o *corte- α forte*:

$$\overline{ac}(A, \alpha) = \{x \in U : m_A(x) > \alpha\} \quad (3.6)$$

Podemos usar o *corte- α* e o *corte- α forte* para definir o núcleo e o suporte de um conjunto:

$$core(A) = ac(A, 1) \quad (3.7)$$

$$support(A) = \overline{ac}(A, 0) \quad (3.8)$$

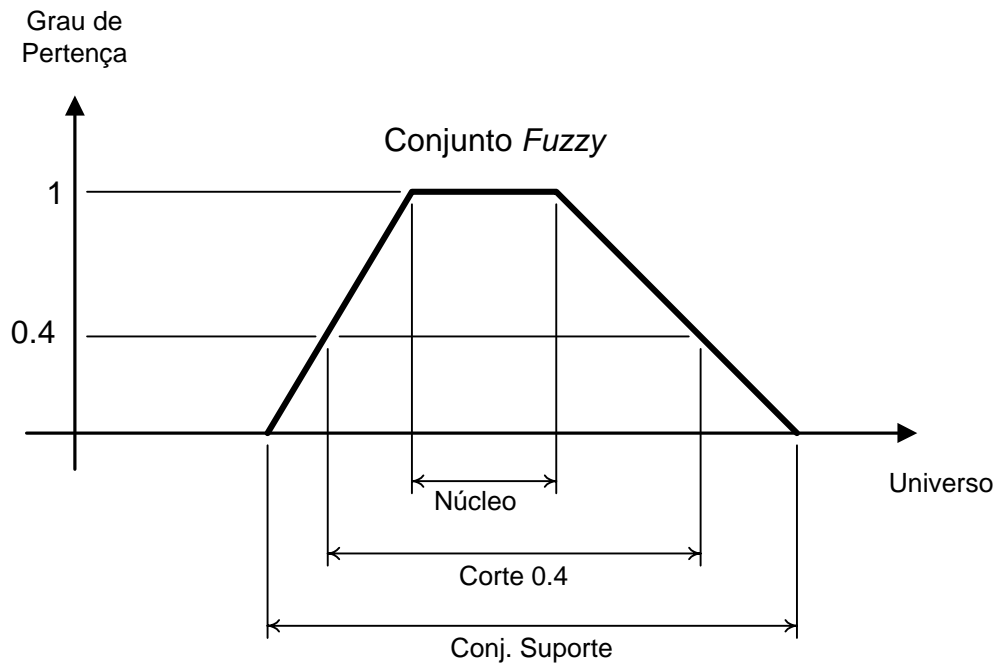


Figura 3.1: Conjunto *fuzzy* com a representação gráfica do seu núcleo, suporte e corte-0.4

A figura 3.1 ilustra os conceitos mencionados.

Para $U = \mathfrak{R}^n$, um conjunto *fuzzy* será convexo se e só se todos os seus *corte- α* forem convexos.

Será vantajoso definir uma variante do *corte- α* . Tal servirá para caracterizar certas operações que iremos realizar sobre conjuntos *fuzzy*. Temos então o *corte- α fuzzy* do qual resulta, não um conjunto clássico, mas sim um conjunto *fuzzy*:

$$B = fac(A, \alpha) = \left\{ (x, m_A(x)) : x \in U, m_B(x) = \begin{cases} m_A(x) & \leq m_A(x) \geq \alpha \\ 0 & \leq m_A(x) < \alpha \end{cases} \right\} \quad (3.9)$$

Esta operação permite retirar de um conjunto *fuzzy* os elementos abaixo de um dado nível α sem perder a informação acerca do grau de pertença associada aos elementos não retirados.

3.2 Operações com Conjuntos *Fuzzy*

Para tratarmos os conjuntos *fuzzy* como uma extensão dos conjuntos clássicos devemos definir as operações intersecção, reunião e complemento. Essas

operações devem ser definidas de uma forma coerente para poderem ser vistas como uma generalização das operações originais.

3.2.1 Intersecção

Para definir a intersecção de dois conjuntos *fuzzy* A e B recorreremos a uma função \frown :

$$\frown : [0, 1] \times [0, 1] \rightarrow [0, 1] \quad (3.10)$$

Essa função, normalmente chamada T-Norma deverá possuir algumas propriedades extra[8], que garantem a consistência da operação resultante.

Teremos assim:

$$A \cap B = \{(x, m_{A \cap B}(x)) : x \in U, m_{A \cap B}(x) = m_A(x) \frown m_B(x)\} \quad (3.11)$$

Das muitas funções propostas, a função $\min(a, b)$ será a utilizada de aqui para a frente como a operação que induz a intersecção de conjuntos *fuzzy*. Consequentemente a equação 3.11 poderá ser escrita como:

$$A \cap B = \{(x, m_{A \cap B}(x)) : x \in U, m_{A \cap B}(x) = \min(m_A(x), m_B(x))\} \quad (3.12)$$

Ficamos assim com cada elemento do conjuntos obtido da intersecção de dois conjuntos *fuzzy* com um grau de pertença dado pelo mínimo do seu grau de pertença a cada um dos conjuntos originais. Naturalmente, esta operação mantém as mesmas características da intersecção clássica pois $\forall \alpha \in [0, 1], \min(0, \alpha) = 0$ e $\min(1, 1) = 1$ acontece no único caso em que o grau de pertença de um elemento do conjunto resultado da intersecção é um.

3.2.2 Reunião

Para definir a reunião de dois conjuntos *fuzzy* A e B recorreremos a uma função \oplus :

$$\oplus : [0, 1] \times [0, 1] \rightarrow [0, 1] \quad (3.13)$$

Tal como para o caso de uma T-Norma, essa função, chamada S-Norma deverá possuir algumas propriedades que a qualificam como tal[8].

Teremos assim:

$$A \cup B = \{(x, m_{A \cup B}(x)) : x \in U, m_{A \cup B}(x) = m_A(x) \oplus m_B(x)\} \quad (3.14)$$

Das muitas funções propostas, a função $\max(a, b)$ será a utilizada de aqui para a frente como a operação que induz a reunião de conjuntos *fuzzy*. Consequentemente a equação 3.14 poderá ser escrita como:

$$A \cup B = \{(x, m_{A \cup B}(x)) : x \in U, m_{A \cup B}(x) = \max(m_A(x), m_B(x))\} \quad (3.15)$$

Ficamos assim com cada elemento do conjunto obtido da reunião de dois conjuntos *fuzzy* com um grau de pertença dado pelo máximo do seu grau de pertença a cada um dos conjuntos originais. Mais uma vez, esta operação mantém as mesmas características, neste caso, da reunião clássica pois verifica-se que $\forall \alpha \in [0, 1], \max(1, \alpha) = 1$ e $\max(0, 0) = 0$ acontece no único caso em que o grau de pertença de um elemento do conjunto resultado da reunião é zero.

3.2.3 Complemento

O complemento de um conjunto *fuzzy* A será dado por:

$$A^c = \{(x, m_{A^c}(x)) : x \in U, m_{A^c}(x) = 1 - m_A(x)\} \quad (3.16)$$

Mais uma vez, realce-se, trata-se da extensão do conceito já existente para conjuntos clássicos.

3.2.4 Subconjuntos

Outro conceito presente na teoria de conjuntos clássica é o de subconjunto. Diz-se que um conjunto A é um subconjunto de B se todos os seus elementos estiverem presentes em B . Esse conceito pode ser generalizado para abranger os conjuntos *fuzzy*:

$$A \subset B \text{ sse } \forall x \in U, m_A(x) \leq m_B(x) \quad (3.17)$$

Basta substituir o "estiverem presentes em B " por "possuírem um grau de pertença a A inferior ou igual ao grau de pertença a B ". Podemos visualizar esta condição verificando se o gráfico de $m_A(x)$ se encontra sempre por baixo do de $m_B(x)$.

3.3 Cardinal, Entropia e outros Conceitos

Os conceitos seguintes serão definidos para o caso de conjuntos *fuzzy* num universo U finito[16].

Consideremos então o universo $U = \{x_1, x_2, \dots, x_n\}$.

3.3.1 Distância de dois conjuntos

Recorramos outra vez ao conceito de conjunto definido como um vector com dimensão n cujas componentes são zero ou 1. Todos os possíveis conjuntos, *fuzzy* ou não, estarão associados a um vector contido no hipercubo unitário $I^n = [0, 1]^n$. Podemos considerar a distância l^p , entre os pontos definidos

por dois conjuntos A e B :

$$l^p(A, B) = \sqrt[p]{\sum_{i=1}^n |m_A(x_i) - m_B(x_i)|^p} \quad (3.18)$$

para $1 \leq p \leq \infty$.

3.3.2 Cardinal

O cardinal de um conjunto *fuzzy* pode ser definido por:

$$M(A) = \sum_{i=1}^n m_A(x_i) \quad (3.19)$$

Mais uma vez podemos verificar que esta operação generaliza o conceito de cardinal para o caso de um conjunto clássico.

Alternativamente podemos usar o conceito de distância entre conjuntos para definir o cardinal:

$$M(A) = l^1(a, \phi) \quad (3.20)$$

3.3.3 Entropia

A noção de entropia, neste caso, mede quão *fuzzy* um conjunto é. À partida, um conjunto clássico, sendo completamente não *fuzzy*, deverá apresentar uma entropia zero. Como sabemos, os conjuntos clássicos podem ser encontrados nos vértices do hipercubo unitário. Recorrendo à distância que um conjunto apresenta em relação a um conjunto não *fuzzy* podemos derivar uma expressão para a entropia desse conjunto. Definamos:

$$A_{perto} = \min_{X \in U} (l^1(A, X)) \quad (3.21)$$

como o conjunto não *fuzzy* mais próximo de A e:

$$A_{longe} = \max_{X \in U} (l^1(A, X)) \quad (3.22)$$

como o conjunto não *fuzzy* mais distante de A .

Temos então a entropia de A como:

$$E(A) = \frac{l^1(A, A_{perto})}{l^1(A, A_{longe})} \quad (3.23)$$

Este valor é zero para todos os conjuntos clássicos. Para conjuntos fuzzy atinge o valor de 1, como o seu máximo, para o conjunto que tem todos os elementos com um grau de pertença $\frac{1}{2}$. Podemos considerar que este é o caso em que há um mínimo de informação. Não estamos a trazer nenhuma informação acerca da pertença de um elemento ao conjunto pois estamos a pôr o seu grau de pertença igual ao grau de não pertença.

3.3.4 Teorema da Entropia *Fuzzy*

Este teorema, apresentado em [16] apresenta uma formulação interessante para o valor da entropia de um conjunto:

$$E(A) = \frac{M(A \cap A^c)}{M(A \cup A^c)} \quad (3.24)$$

À partida pode parecer uma formulação estranha pois para conjuntos clássicos verifica-se *sempre* que $A \cap A^c = \phi$ e que $A \cup A^c = U$ e logo teremos $E(A) = \frac{M(\phi)}{M(U)} = \frac{0}{n} = 0$. Esse é o valor correcto da entropia para conjuntos clássicos mas para conjuntos *fuzzy* $A \cap A^c \neq \phi$ assim como $A \cup A^c \neq U$.

De notar que este teorema apenas é válido quando de usa como T-Norma a função $\min(a, b)$ e como S-Norma a função $\max(a, b)$. Isso liga a definição de cardinal, baseada na norma l^1 , às funções \min e \max como T-Norma e S-Norma.

3.4 Proposições *Fuzzy*

Podemos considerar, classicamente, que se uma proposição p pertence ao Conjunto das Proposições Verdadeiras U_v então o seu valor lógico $t(p) = 1$. Por oposição, se p não pertence a U_v então $t(p) = 0$ e diz-se que temos uma proposição falsa. Aceitando que U_v pode ser um conjunto *fuzzy* temos então proposições s que podem pertencer a U_v num grau inferior a um e superior a zero. Genericamente, $0 \leq t(s) \leq 1$ e para as proposições s em que $0 < t(s) < 1$ temos uma proposição estritamente *fuzzy*. Estamos no caso em que, não podendo dizer se uma proposição é falsa ou verdadeira podemos ainda indicar quão está próxima de um desses extremos. Esta ferramenta pode ser muito útil para modelizar situações em que a incerteza não provém do nosso desconhecimento acerca de um dado acontecimento (aí o recurso a considerações estocásticas é a ferramenta preferida) mas sim do facto que o acontecimento contém em si um certo grau de ambiguidade.

3.4.1 Extensão *Fuzzy* do Conceito de Subconjunto

O conceito apresentado anteriormente decidia $t(A \subset B)$ num contexto clássico, isto é, indicava se essa proposição era verdadeira ou falsa. Porém consideremos o seguinte caso:

Dado o universo:

$$U_{10} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\} \quad (3.25)$$

e os conjuntos *fuzzy*

$$A = [0.5 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0.3 \ 0.2] \quad (3.26)$$

e

$$B = [0.4 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1] \quad (3.27)$$

podemos verificar que $A \subset B$ é uma proposição falsa. Para o primeiro elemento temos $m_A(x_1) = 0.5 > m_B(x_1) = 0.4$ e isso viola a regra que define se um conjunto é um subconjunto de outro. Porém A *quase* é um subconjunto de B e essa informação é irremediavelmente perdida ao atribuir o valor zero à proposição $A \subset B$. Podemos então definir a operação $S(A, B)$ que atribui um valor lógico entre zero e um à proposição $A \subset B$, interpretando-a como uma proposição *fuzzy*.

$$S(A, B) = 1 - \frac{\sum_{i=1}^n \max(0, m_A(x_i) - m_B(x_i))}{M(A)} \quad (3.28)$$

Assim, para o exemplo apresentado teríamos

$$M(A) = 5 \quad (3.29)$$

e

$$\sum_{i=1}^{10} \max(0, m_A(x_i) - m_B(x_i)) = 0.1 \quad (3.30)$$

logo

$$S(A, B) = \frac{49}{50} = 0.98 \quad (3.31)$$

Este valor, tão perto de um, indica que realmente A quase é um subconjunto de B .

Apresentamos também a relação, cuja demonstração, pode ser encontrada em [16], que exprime o conceito de subconjunto versão *fuzzy* recorrendo apenas ao cardinal de dois conjuntos:

$$S(A, B) = \frac{M(A \cap B)}{M(A)} \quad (3.32)$$

3.5 *Fuzzy* versus Probabilidade

Com a divulgação dos conceitos ligados a conjuntos e lógica *fuzzy* apareceu a controvérsia relativa aos méritos desta metodologia face ao paradigma aceite para representar fenómenos com incerteza. Tradicionalmente esse era o papel reservado às probabilidades e a toda a teoria relativa à descrição estocástica de fenómenos. Embora essa controvérsia já não esteja tão intensa, os seus ecos ainda obrigam à inclusão de uma secção onde esse problema é abordado.

À partida, uma ferramenta teórica que se mostre usável e capaz de descrever com precisão adequados fenómenos físicos, tem a sua existência perfeitamente justificada. Pode-se argumentar que se trata de uma particularização de uma teoria mais geral mas, mesmo que seja esse o caso, se essa particularização simplifica a operação e permite uma visualização mais directa das propriedades do sistema, então mantém ainda a sua razão de existir.

Não parece haver dúvidas que entre *fuzzy* e probabilidades estamos perante paradigmas diferentes, assentes em premissas diferentes e com propriedades diferentes. De um modo geral, a interpretação que podemos dar a cada uma das ferramentas assenta em diferentes fontes de incerteza. Tipicamente a incerteza modelizada probabilisticamente tem a ver com o facto de não dispormos de informação completa sobre os acontecimentos. Quando dizemos que o resultado de um lançamento de um dado tem uma probabilidade de $\frac{1}{6}$ de ser dois, estamos a fazer isso porque não vimos o resultado. É essa, a nossa fonte de incerteza. Quer porque estamos a fazer uma previsão e o dado ainda não foi lançado ou porque ele foi lançado e não temos acesso ao resultado. O mesmo acontece com uma medida. Tipicamente ela vem contaminada com algum ruído e não conhecemos a realização desse ruído. Modelizamos então esse valor com uma variável aleatória fornecendo uma probabilidade, adequadamente calculada, a cada um dos valores possíveis. (No caso de um valor contínuo fornecemos a probabilidade da medida estar num dado intervalo, ou mais concretamente a função densidade de probabilidade.) Isso é equivalente ao caso em que o dado foi lançado mas não conseguimos ver o resultado.

Estes são casos em que um modelo probabilístico se adequa extremamente bem. Mais ainda, não faz muito sentido recorrer a conjuntos *fuzzy* por causa deste género de incerteza. Porém há casos em que os papéis se invertem. Tipicamente, esses são casos em que a incerteza advém de não estar muito nítida a fronteira entre dois fenómenos. E de essa incerteza não poder ser desfeita com mais informação. Veja-se uma variante do problema proposto por Kosko[16]:

Temos na Figura 3.2 um quadrado? Suponhamos um contexto em que estamos a procurar identificar formas numa imagem e deparamos com a imagem apresentada na figura 3.2. Realmente não se trata de um quadrado: os lados não são rectos, a largura não é exactamente igual à altura, os cantos não formam ângulos de 90° . Porém, parece-se muito com um quadrado, se relaxarmos os pressupostos podemos aceitá-lo como um quadrado. Infelizmente perderemos a informação de que não se trata de um quadrado *exacto*. Pior ainda, era possível construir duas figuras, muito parecidas, em que uma delas seria considerada um quadrado pois o seu formato ainda era classificado dentro do nível escolhido enquanto que outra estava abaixo desse limiar. Classificar uma das imagens como um quadrado e outra não, iria resultar pouco aceitável dada a sua semelhança. Este é um problema semelhante ao

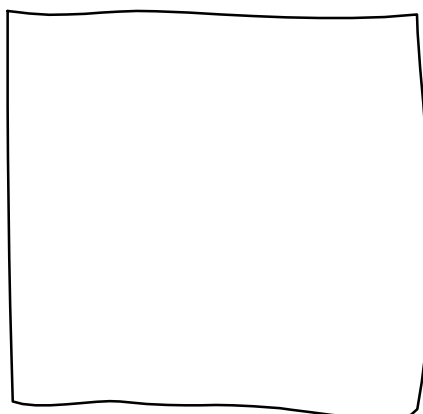


Figura 3.2: Um quadrado?

suposto Paradoxo da Duna:

1. Uma duna contém um número finito de grãos de areia.
2. Retirar um grão de areia a uma duna não faz com que deixe de ser uma duna.
3. Repetindo 2. um número finito de vezes conseguimos atingir o caso em que não há mais grãos de areia.
4. Consequentemente, um buraco pode ser uma duna!

Onde falha a sequência anterior?

Não podemos esquecer que temos de impor um limiar, por exemplo na altura, que separa uma duna de uma não-duna. Consequentemente haverá um instante em que a proposição 2. é falsa. Essa é a vez em que retirar um grão contribuirá para alterar a altura da duna para baixo do limiar proposto. Trata-se de um caso em que o nosso modelo tem um comportamento pouco intuitivo. Apesar disso, muitas vezes é um modelo perfeitamente aceitável pois este efeito não aparece de uma forma tão desagradável. Mas este problema seria resolvido de uma forma extremamente elegante caso consigamos atribuir um valor lógico entre zero e um à proposição s "Este monte de areia é uma duna". Assim podemos ter casos em que retirar um grão de areia baixará o valor lógico de s $t(s)$ para um valor $t(s) - \varepsilon$ que é ligeiramente inferior. Desse modo podemos construir uma transição suave de duna para não-duna, mantendo inclusive a informação de quanto estamos afastados desse arquétipo. Note-se que aqui temos um caso em que não há falta de informação, podemos ter conhecimento de quantos grãos de areia temos, de como eles estão dispostos, com toda a precisão. O que conta é o facto do conceito de duna ter alguma latitude que queremos que seja incorporada no



Figura 3.3: Quão vermelho é o *vermelho 2*?

modelo. Nestes casos a técnica de quantizar em verdadeiro ou falso, acima ou abaixo, para resolver a ambiguidade pode ser muito grosseira.

Um outro exemplo, particularmente adequado e bastante relacionado com o problema que estamos a tentar resolver, é o de classificar uma cor. Quão amarelo é um dado amarelo? Como classificar os vermelhos apresentados na figura 3.3? Parece evidente que alguns dos vermelhos são mais vermelhos que outros. Trata-se de mais um caso em que o uso de conjuntos *fuzzy* parece adequar-se ao problema de uma maneira impressionante.

Naturalmente, a interpretação a dar a cada uma das metodologias não é definitiva. Há ainda casos em que temos um pouco dos dois efeitos. A incerteza pode ser provocada por ambiguidade na classificação mas ao mesmo tempo existe o problema de as medidas aparecerem perturbadas por ruído. Porém conhecer os pontos fortes e fracos de cada um dos paradigmas permitirá a escolha do modelo adequado para cada caso.

Capítulo 4

Processos Estocásticos e o Filtro de Kalman

Neste capítulo apresentamos os conceitos básicos correspondentes à descrição estocástica dos fenómenos que queremos modelizar. Apresentamos também a formulação do filtro de Kalman e do Filtro de Kalman Extendido correspondentes às variantes que implementamos. Desta maneira fica a evidente a notação empregue assim como os pressupostos que consideramos necessários.

4.1 Variáveis Aleatórias e Processos Estocásticos

Quando construímos um modelo de um dado fenómeno estamos interessados em quantificar as grandezas envolvidas. Quer porque estamos a fazer previsões acerca dum acontecimento futuro, quer porque as medidas de que dispomos não são totalmente precisas, encontramos-nos muitas vezes na situação em que não é possível caracterizar, com toda a precisão essas grandezas. Não conseguir estabelecer o valor preciso de uma grandeza não é motivo para desermos para o outro extremo, onde manifestamos o mais completo desconhecimento acerca do seu valor. Num ponto intermédio, procuraremos caracterizar a grandeza indicando a *probabilidade* de cada possível valor.

Consideremos pois o universo U_Y de todos os valores que uma variável Y pode tomar. Se a grandeza que Y representa toma valores reais chamaremos a Y uma *variável aleatória*. Podemos expandir o conceito para o caso vectorial quando a grandeza em questão seja mais correctamente representada por um vector.

Nos casos em que Y é uma função do tempo $Y(t)$ designamo-la um *processo estocástico*. Temos assim que o universo de um processo estocástico será um conjunto de funções. Usualmente t é um parâmetro definido em \mathfrak{R} ou em \mathfrak{R}_0^+ . Nos casos em que a variável que indexa o tempo só toma valores discretos, temos o nosso universo constituído não por funções mas sim por

sucessões. Para facilitar a distinção em relação ao caso anterior denotaremos a função que indexa o tempo por k . Assim, para $Y(k)$ assumiremos que $k \in \{0, 1, \dots\}$ ou $k \in \{\dots, -1, 0, 1, \dots\}$.

A definição do universo não é, naturalmente, suficiente para caracterizar uma variável aleatória. É necessário estabelecer uma *lei de probabilidade* que indique quão provável é que Y tome um dado valor. Podemos considerar a resolução de Y num conjunto de valores como um evento A . A lei de probabilidade indicará então a probabilidade $P(A)$ (ou $P(Y = A)$) desse evento. Dispondo então de uma correspondência de um para um entre eventos e subconjuntos do universo, teremos uma lei de probabilidades caso se verifique:

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 infinita de eventos mutualmente exclusiva

Naturalmente definir explicitamente a lei de probabilidade para todos os eventos só será possível caso o nosso universo seja finito e o seu número de elementos não muito elevado. Normalmente, definiremos a lei de probabilidade recorrendo a uma *função densidade de probabilidade* ou mais abreviadamente fdp. Para o caso de uma variável aleatória escalar Y a função densidade de probabilidade $f_y(x)$ definida de \mathfrak{R} em \mathfrak{R} deverá obedecer a:

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

Esta função permite o cálculo da probabilidade de um dado evento $P(A)$ graças a:

$$P(A) = \int_A f_y(x) dx \quad (4.1)$$

No caso em que os valores que a variável aleatória pode tomar são discretos, isto é o universo U_y é finito, podemos descrever directamente a função $P(A)$ garantindo que verifica:

$$\sum_i P(Y = i) = 1$$

A função densidade de probabilidade de uma variável aleatória vectorial $Y = (y_1, \dots, y_n)$ segue um esquema semelhante. Teremos $f_y(X)$ com $X \in \mathfrak{R}^n$:

$$\begin{aligned} f_y : \mathfrak{R}^n &\longrightarrow \mathfrak{R} \\ (X) &\longrightarrow f_y(X) \end{aligned} \quad (4.2)$$

que deverá satisfazer:

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

Podemos obter $P(A)$ através de:

$$P(A) = \int_A \cdots \int f_y(X) dX \quad (4.3)$$

Infelizmente, representar a lei de probabilidade para um processo aleatório através da função densidade de probabilidade é, em geral, um processo demasiado complexo. Dado que um processo estocástico envolve um número infinito de variáveis aleatórias torna-se necessários, no mínimo, definir as densidades conjuntas para cada grupo de i variáveis aleatórias:

$$Y(t_1), Y(t_2), \dots, Y(t_i) \quad (4.4)$$

para:

$$t_1 < t_2 < \dots < t_i \quad (4.5)$$

Este processo deve ser aplicado para todas as escolhas possíveis de $t_1 < t_2 < \dots < t_i$ e para todos os grupos com $i \in \mathfrak{N}$. A densidade conjunta para o grupo i pode ser designada por densidade de ordem i do processo. Estas funções devem ainda manter a consistência em termos das respectivas densidades marginais, por exemplo deve-se verificar:

$$f_{x(t_1)}(x(t_1)) = \int_{-\infty}^{+\infty} f_{x(t_1), x(t_2)}(x(t_1), x(t_2)) dx(t_2) \quad (4.6)$$

Naturalmente, recorrer a esta técnica para definir a lei de probabilidade de um processo estocástico é obviamente uma tarefa desaconselhável. Este problema será significativamente simplificado caso se proceda a uma conveniente restrição das classes de distribuições que queremos ter à nossa disposição. Para proceder a essa tarefa é necessário a introdução de alguns conceitos.

Um parâmetro frequentemente usado para caracterizar certos atributos de uma variável aleatória é a sua *média*:

$$\mu_y = E(Y) = \int_{-\infty}^{+\infty} x f_y(x) dx \quad (4.7)$$

Outros parâmetros interessantes são a *variância*:

$$\sigma_y^2 = V(Y) = E\left(\left(Y - \mu_y\right)^2\right) = E\left(Y^2\right) - \left(\mu_y\right)^2 \quad (4.8)$$

e a *covariância* que é definida entre duas variáveis aleatórias Y e Z com médias respectivamente μ_y e μ_z :

$$\text{cov}(Y, Z) = E\left((Y - \mu_y)(Z - \mu_z)\right) = E(YZ) - \mu_y\mu_z \quad (4.9)$$

No caso em que $Y = (y_1, \dots, y_n)$ é uma variável aleatória vectorial teremos que a média será um vector:

$$\mu_y = (E(y_1), \dots, E(y_n)) \quad (4.10)$$

e a matriz:

$$\sigma_y^2 = \begin{bmatrix} \text{cov}(y_1, y_1) & \text{cov}(y_1, y_2) & \dots & \text{cov}(y_1, y_n) \\ \text{cov}(y_2, y_1) & \text{cov}(y_2, y_2) & \dots & \text{cov}(y_2, y_n) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(y_n, y_1) & \text{cov}(y_n, y_2) & \dots & \text{cov}(y_n, y_n) \end{bmatrix} \quad (4.11)$$

é designada por *matriz de covariância*. Note-se que as variâncias $V(y_i)$ podem ser encontradas na diagonal da matriz.

Para o caso de um processo estocástico $Y(t)$ estes parâmetros serão definidos em função da variável que indexa o tempo. Pode-se ainda definir outras duas funções que recebem como argumento dois instantes de tempo t_1 e t_2 :

$$R_y(t_1, t_2) = E\left(Y(t_1)Y(t_2)^T\right) \quad (4.12)$$

normalmente designada pela *correlação*, e também

$$C_y(t_1, t_2) = \text{cov}(Y(t_1), Y(t_2)) \quad (4.13)$$

que é uma matriz de covariância, mas entre duas variáveis aleatórias.

Como

$$C_y(t_1, t_2) = R_y(t_1, t_2) - \mu_y(t_1)\mu_y(t_2) \quad (4.14)$$

para processos de média nula teremos:

$$R_y(t_1, t_2) = C_y(t_1, t_2) \quad (4.15)$$

Por este motivo, veremos que em muita da literatura se usa $R_y(t_1, t_2)$ no lugar de $C_y(t_1, t_2)$.

Para processos estacionários, verifica-se que estas funções só dependem da diferença ente t_1 e t_2 . Nesses casos, é costume definir $\tau = t_2 - t_1$ e escrever $R_y(\tau)$ e $C_y(\tau)$.

Os processos estocásticos cujas funções de densidade de probabilidade de ordem k seguem uma distribuição gaussiana, são denominados processos gaussianos. Como a função de densidade de probabilidade de uma distribuição gaussiana fica completamente definida pela sua média e covariância

temos que um processo gaussiano terá todas as suas funções de densidade de probabilidade definidas graças a $\mu_Y(t) = E(Y(t))$ e $C_y(t_1, t_2) = \text{cov}(Y(t_1), Y(t_2))$ desde que estejam definidos para todo o t e qualquer combinação de t_1 e t_2 . Restringindo-nos a processos gaussianos conseguiremos um método viável de descrever a lei de probabilidade do processo.

Felizmente esta restrição não é demasiado inibidora, para além da facilidade em descrever o processo, em geral os processos gaussianos apresentam ainda uma série de características bastante importantes:

- A combinação linear de processos gaussianos dá origem ainda a um processo gaussiano. Graças a esta propriedade o conjunto dos processos gaussianos é fechado para operações lineares. Isso permite que o resultado deste género de operações possa ser ainda representado com exactidão. Caso tal não se verificasse, poderíamos facilmente ter operações lineares cujo resultado somente poderia ser apresentado de uma forma aproximada. Outra consequência desta propriedade é que dispendo de um sistema dinâmico linear e se na sua entrada considerarmos presente um processo gaussiano, obteremos na saída ainda um processo gaussiano.
- Muitos sinais vulgarmente presentes nos sistemas que queremos modelizar são adequadamente representados por processos gaussianos. Todo o nosso esforço seria em vão caso o poder de modelização de um processo gaussiano fosse baixo. Na realidade trata-se do caso oposto pois uma grande parte dos sinais com características estocásticas pode ser representado com um grau de fidelidade excelente recorrendo a sinais com distribuição gaussiana. Uma justificação para esse facto decorre de uma propriedade desta distribuição que é evidenciada no Teorema do Limite Central:

Sendo $Z(n)$, $n \in \mathbb{N}$ uma sequência de variáveis aleatórias independentes com variância $V(Z(n)) = \sigma^2 < \infty$ então a sequência

$$\frac{S(n) - E(S(n))}{\sqrt{V(S(n))}} \quad (4.16)$$

onde $S(n) = \sum_{i=1}^n Z(i)$, converge em distribuição para a variável gaussiana normalizada, quando $n \rightarrow +\infty$.

Nota: a variável gaussiana normalizada é uma variável aleatória com distribuição gaussiana, média nula e variância unitária.

Melhor ainda, mesmo que as variáveis $Z(n)$ não apresentem a mesma variância, desde que ela esteja limitada inferiormente e superiormente, isto é $0 < \varepsilon \leq V(Z(n)) \leq C < \infty$, então o teorema do limite central ainda se aplica.

Assim, temos que o efeito de muitas perturbações, independentemente das distribuições individuais, poderá ser sumariado recorrendo a uma distribuição gaussiana.

- Uma série de outras propriedades facilitam a análise de sistemas onde apenas processos gaussianos estão envolvidos. Por exemplo, a ausência de correlação indica independência, enquanto no caso geral isso não pode ser assumido. A hipótese ergódica é garantida desde que o processo seja estacionário e a covariância absolutamente integrável.

Por todas estas razões, recorreremos quase exclusivamente a processos gaussianos para descrever os sistemas que iremos estudar.

4.2 O Filtro de Kalman

Para caracterizar o estado de um sistema num dado instante de tempo, normalmente dispomos de duas fontes de informação.

1. Uma fonte de informação é o estado anterior do sistema. Se conhecermos o estado do sistema num dado instante, anterior ao actual, e também dispusermos de um modelo da dinâmica do sistema poderemos "prever" o estado actual. Para tal recorreremos ao modelo e calculamos o estado actual considerando que o sistema evoluiu tal como o modelo prevê. Esta técnica pode ser utilizada mesmo que o estado esteja descrito recorrendo a uma variável aleatória. Nesse caso o modelo fornecerá uma nova variável aleatória que descreve uma previsão para o estado actual. Como vimos, caso estejamos a trabalhar com distribuições gaussianas bastará caracterizar a média e a covariância do novo estado. Naturalmente que este estimador vai introduzir algum erro inerente ao facto do modelo não ser uma representação completamente exacta da dinâmica do sistema.
2. A outra grande fonte de informação é o conjunto de observações provenientes dos sensores conectados ao sistema. Estando a medir as grandezas adequadas será possível reconstruir o estado do sistema. Mesmo que para isso seja necessária uma sequência de medidas. Infelizmente, esta informação normalmente não permite obter o estado do sistema com toda a exactidão. Invariavelmente, as medidas estão contaminadas com ruído proveniente de muitas fontes. Basta o facto de a medida ter sido quantizada em n bits para que o arredondamento implícito possa ser considerado como uma fonte de ruído.

Temos assim a situação em que gostaríamos de combinar a informação destas duas fontes de uma forma eficiente, preferencialmente óptima, de modo a obter uma estimativa do estado que apresente o mínimo de incerteza.

Neste caso estamos a caracterizar a incerteza como a covariância da variável aleatória que representa o estado actual.

O Filtro de Kalman implementa o mecanismo que desempenha o papel que desejávamos. Dentro de determinadas condições, ele possibilitará a fusão óptima da informação proveniente da dinâmica e das observações. Isto, considerando a optimalidade como a minimização da variância da estimativa do estado.

Consideremos um sistema com dinâmica linear e cujas saídas são uma combinação linear do estado, isto é:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (4.17)$$

$$y(t) = Cx(t) \quad (4.18)$$

com $x = [x_1 \ x_2 \ \dots \ x_n]^T$, $u = [u_1 \ u_2 \ \dots \ u_p]^T$ e $y = [y_1 \ y_2 \ \dots \ y_q]^T$.

As matrizes A , B e C terão as dimensões adequadas, neste caso A será uma matriz com $m \times m$, B terá dimensões $p \times m$ e C $m \times q$, por uma questão de simplicidade consideraremos que são invariantes no tempo.

Por vezes, por uma questão de facilidade de implementação, é mais conveniente dispor da versão em tempo discreto deste modelo. Um modelo linear em tempo contínuo pode ser convertido para um modelo discreto sem perda de informação, desde que os instantes de actuação e de amostragem estejam bem definidos. Coincidindo os instantes em que o novo controlo se altera com os instantes de amostragem das medidas, temos que para o sistema 4.17 o modelo em tempo discreto correspondente será:

$$x(k) = \Phi x(k-1) + Gu(k-1) \quad (4.19)$$

$$y(k) = Hx(k) \quad (4.20)$$

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

$$\Delta t_k = t_k - t_{k-1} \quad (4.21)$$

$$\Phi(k) = e^{A\Delta t_k} \quad (4.22)$$

$$G = \int_{t_{k-1}}^{t_k} e^{At} B dt \quad (4.23)$$

$$H = C \quad (4.24)$$

Este modelo é ainda um modelo determinístico. Na sua versão estocástica introduzimos as parcelas que modelizam a incerteza associada à dinâmica do sistema $w(k)$ e a incerteza proveniente das medidas $v(k)$:

$$x(k) = \Phi x(k-1) + Gu(k-1) + w(k-1) \quad (4.25)$$

$$y(k) = Hx(k) + v(k) \quad (4.26)$$

Naturalmente, $w(k)$ é um processo estocástico que deverá ser devidamente caracterizado. Normalmente, para assegurar a convergência do filtro de Kalman é necessário que:

$$R_w(k, i) = \begin{cases} Q(k), & \text{se } i = k \\ 0 & \text{se } i \neq k \end{cases} \quad (4.27)$$

$$R_v(k, i) = \begin{cases} R(k), & \text{se } i = k \\ 0 & \text{se } i \neq k \end{cases} \quad (4.28)$$

também:

$$E(w(k) v(i)^T) = 0, \quad \forall k, i \quad (4.29)$$

Estas restrições definem $w(k)$ e $v(k)$ como processos estocásticos compostos por sequências independentes de variáveis aleatórias com variância $Q(k)$ e $R(k)$, respectivamente. Mais ainda 4.29 garante que $w(k)$ e $v(k)$ são processos não correlacionados. Caso queiramos acrescentar à garantia de convergência a optimalidade do filtro bastará adicionar a restrição extra de termos $w(k)$ e $v(k)$ como processos gaussianos. Tendo estabelecido o modelo adequado ao uso de um filtro de Kalman podemos passar a descrever o filtro propriamente dito.

4.2.1 Filtro de Kalman discreto

Associados ao filtro de Kalman há dois grupos de operações, a operação de *previsão* e a operação de *actualização*. Durante a previsão, o estado no instante $k - 1$ é usado para prever, recorrendo às equações do modelo, o estado do sistema no instante k . Naturalmente, durante essa fase a covariância da estimativa do estado terá tendência a aumentar sinalizando a nossa menor confiança nesta extrapolação. Obtemos então uma estimativa provisória do estado, no instante k , e da respectiva covariância. Durante a actualização, as medidas efectuadas no instante k são usadas para aperfeiçoar a nossa estimativa do estado. Normalmente obteremos uma estimativa mais precisa graças à introdução dessa informação extra. Como resultado dessa operação a covariância associada deverá diminuir reportando a nossa maior confiança nos resultados obtidos graças à fusão de toda a informação disponível.

Temos pois, para o instante k , dois momentos distintos: um deles encontra-se logo após a previsão e antes da actualização, o outro momento corresponde àquele em que dispomos do valor final da estimativa do estado, e da sua covariância, logo após a actualização. Para distinguir esses dois momentos usaremos o instante k^- e k . Assim, quando fizermos a previsão representaremos o estado obtido por $x(k^-)$, após a actualização passaremos a escrever $x(k)$.

Dado que a formulação do filtro de Kalman é recursiva, isto é, nós temos a estimativa do estado $x(k)$ dada a partir do estado anterior $x(k - 1)$, torna-

se necessário indicar uma condição inicial, tanto para o estado do sistema $x(0)$ como para a covariância associada $P(0)$.

Equações do Filtro de Kalman discreto

Assim, considerando um sistema dinâmico tal como apresentado em 4.25 e considerando os pressupostos 4.27, 4.28 e 4.29 podemos resumir o filtro de Kalman:

Condições Iniciais:

$$E(x(0)) = X_0, \text{ cov}(x(0)) = P_0 \quad (4.30)$$

Modelo do Sistema:

$$x(k) = \Phi x(k-1) + Gu(k-1) + w(k-1) \quad (4.31)$$

Modelo das Observações:

$$y(k) = Hx(k) + v(k) \quad (4.32)$$

Propagação do Estado:

$$x(k^-) = \Phi x(k-1) + Bu(k-1) \quad (4.33)$$

Propagação da Covariância:

$$P(k^-) = \Phi P(k-1) \Phi^T + Q(k) \quad (4.34)$$

Cálculo do Ganho de Kalman:

$$K(k) = P(k^-) H^T \left(H P(k^-) H^T + R(k) \right)^{-1} \quad (4.35)$$

Actualização da Covariância:

$$P(k) = (I - K(k) H) P(k^-) \quad (4.36)$$

Actualização do Estado:

$$x(k) = x(k^-) + K(k) (y(k) - Hx(k^-)) \quad (4.37)$$

Por vezes pode-se usar, em vez da equação 4.36 a seguinte equação:

$$P(k) = (I - K(k) H) P(k^-) (I - K(k) H)^T + K(k) R K(k)^T \quad (4.38)$$

Esta equação tem a vantagem de ser válida para qualquer ganho $K(k)$ independentemente da forma como foi calculado. Isto é, mesmo que se trate de um ganho $K(k)$ não óptimo. Infelizmente é mais complexa e por isso mais pesada a sua avaliação e é naturalmente mais sensível a problemas numéricos.

4.2.2 Filtro de Kalman Extendido

Há situações em que o sistema não pode ser adequadamente descrito por um sistema linear. Nesses casos, naturalmente, recorreremos a modelos não lineares e surge a dificuldade de aplicar um filtro de Kalman a este sistema. Tentar calcular e descrever o filtro óptimo para um dado sistema não linear é, muito frequentemente, uma tarefa extremamente complexa. Muitas vezes, um filtro subótimo mas mais simples será perfeitamente adequado. Esse papel pode ser desempenhado por uma variante do filtro de Kalman denominada filtro de Kalman Extendido. Este filtro, para cada instante, lineariza a função de transferência em torno do valor do estado estimado e actualiza as estimativas baseadas num filtro de Kalman aplicado a esse sistema linear. Um problema desta técnica é que, para além da suboptimalidade do filtro, há casos em que o filtro pode apresentar problemas de convergência. Porém, numa grande parte dos casos, esse problema não se manifesta e apesar de subótimo o filtro apresenta um desempenho muito satisfatório.

Assumindo para x , y e u as dimensões já indicadas podemos considerar o sistema não linear:

$$\dot{x}(t) = f(x(t), u(t), t) + \bar{w}(t) \quad (4.39)$$

$$y(t) = h(x(t), t) + \bar{v}(t) \quad (4.40)$$

Trata-se de um modelo estocástico e assumiremos restrições para $\bar{w}(t)$ e $\bar{v}(t)$ semelhantes aquelas que indicámos para o caso linear. Vamos usar directamente este modelo contínuo mas considerando que dispomos de observações apenas em instantes discretos. Muitas vezes estamos a procurar filtrar um sistema cujo processo físico está descrito por equações tipicamente dadas em tempo contínuo. Em muitos casos a passagem para um modelo discreto, devido à não linearidade da equação diferencial, não pode ser feita sem perda de informação, ao contrário do que acontecia para sistemas dinâmicos lineares. Assim, a formulação do filtro de Kalman extendido que apresentaremos pode ser caracterizada como contínua-discreta, pois assume um sistema dinâmico contínuo mas as observações e consequentes actualizações das estimativas acontecem em instantes discretos. Neste ambiente misto usaremos t_k para indicar o instante de tempo correspondente ao momento da k -ésima amostra.

Equações do Filtro de Kalman Extendido

Podemos então resumir as equações do filtro de Kalman Extendido. Note-se que temos algumas funções definidas para todos os instantes de tempo enquanto outras consideraremos definidas apenas nos instantes de amostragem.

Definamos,

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (4.41)$$

e também

$$\frac{\partial h}{\partial x} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \cdots & \frac{\partial h_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_q}{\partial x_1} & \frac{\partial h_q}{\partial x_2} & \cdots & \frac{\partial h_q}{\partial x_n} \end{bmatrix} \quad (4.42)$$

Condições Iniciais:

$$E(x(0)) = X_0, \text{ cov}(x(0)) = P_0 \quad (4.43)$$

Modelo do Sistema:

$$\dot{x}(t) = f(x(t), u(t_k), t) + \bar{w}(t) \quad (4.44)$$

Modelo das Observações:

$$y(t_k) = h(x(t_k), t_k) + \bar{v}(t_k) \quad (4.45)$$

Propagação do Estado:

$$\dot{x}(t) = f(x(t), u(t_k), t), \quad t \in]t_{k-1}, t_k] \quad (4.46)$$

Linearização da Dinâmica:

$$A^*(k) = \left. \frac{\partial f}{\partial x} \right|_{\substack{x = x(t_k) \\ u = u(t_k) \\ t = t_k}} \quad (4.47)$$

Discretização da Dinâmica:

$$\Phi^*(k) = e^{A^*(k)(t_k - t_{k-1})} \quad (4.48)$$

Linearização das Observações:

$$H^*(k) = \left. \frac{\partial h}{\partial x} \right|_{x=x(t_{k-1})} \quad (4.49)$$

Propagação da Covariância:

$$P(k^-) = \Phi^*(k) P(k-1) \Phi^*(k)^T + Q(k) \quad (4.50)$$

Cálculo do Ganho de Kalman:

$$K(k) = P(k^-) H^*(k)^T \left(H^*(k) P(k^-) H^*(k)^T + R(k) \right)^{-1} \quad (4.51)$$

Actualização da Covariância:

$$P(k) = (I - K(k) H^*(k)) P(k^-) \quad (4.52)$$

Actualização do Estado:

$$x(k) = x(k^-) + K(k) (y(k) - h(x(k^-), t_k)) \quad (4.53)$$

Como já foi mencionado o filtro de Kalman Extendido não é óptimo e pode sofrer de problemas de divergência. Dadas as características não lineares da dinâmica, mesmo que à partida a distribuição de $x(0)$ fosse gaussiana, não há garantias que ao longo do tempo assim se mantenha. No entanto estamos a propagar a nossa estimativa da covariância assumindo para a dinâmica uma versão linearizada *em torno do estado estimado*. Estes são dois factores que introduzem, inevitavelmente, alguma imprecisão, tanto menor quanto mais válida for esta aproximação.

Capítulo 5

Sistema de Interpretação de Imagem

Neste capítulo descreveremos o sistema que, a partir da imagem capturada, extrai a informação necessária para localizar os robots e a bola.

Que características deve o sistema de visão possuir:

- *Precisão.* Para lidar com robots com cerca de 10 *cm* de lado e uma bola com 4 *cm* de diâmetro, é necessário que o erro na sua localização não exceda alguns milímetros.
- *Rapidez.* Dado o carácter dinâmico do ambiente, é necessário que a informação da posição dos robots e da bola esteja disponível com um atraso mínimo. No nosso problema, a utilidade de uma medida decai rapidamente com o tempo. Atendendo a que podemos ter velocidades da ordem dos 2 ms^{-1} quer para a bola quer para os robots, facilmente se percebe que um atraso de 100 *ms* pode ser considerado muito grave. Este é um intervalo de tempo suficiente para que um robot e/ou a bola se tenham deslocado cerca de 20 *cm*. Se acrescentarmos a esse tempo o necessário para que os sistemas de decisão e de controlo forneçam aos robots os novos comandos veremos que a consequência é uma qualidade de decisão e de controlo muito baixa.
- *Robustez.* O processo de passar de uma grelha de pixeis para medidas do estado dos robots e da bola é um processo delicado. A interpretação da imagem, por vezes, dá origem a medidas de posição completamente disparatadas devido a um erro na identificação dos marcadores. Como poderemos ver, a cor percebida pela câmara é extremamente dependente das condições de iluminação. A presença, mesmo que muito tênue de uma sombra pode alterar significativamente a cor. Este fenómeno não é muito perceptível à vista desarmada. Verifica-se que o cérebro humano é capaz de compensar, de uma forma automática, esse efeito. É por isso que somos capazes de distinguir e identificar com

facilidade as cores em condições de iluminação extremamente díspares. Esse processamento, tão automático que não é por nós notado, não existe para uma câmara/computador.



Figura 5.1: O sistema de captura de imagem

Fisicamente, o sistema consiste em:

- Uma ou mais câmaras de vídeo colocadas directamente sobre o campo. Segundo os regulamentos do RoboCup a altura máxima é de 3 metros. De um modo geral, quanto mais elevadas estiverem as câmaras, em relação ao campo, melhor. Isso deve-se a uma série de factores:

Paralaxe. O efeitos de paralaxe são minimizados. Como podemos ver na figura 5.2 para uma altura $h_2 > h$ o erro de paralaxe diminui visivelmente.

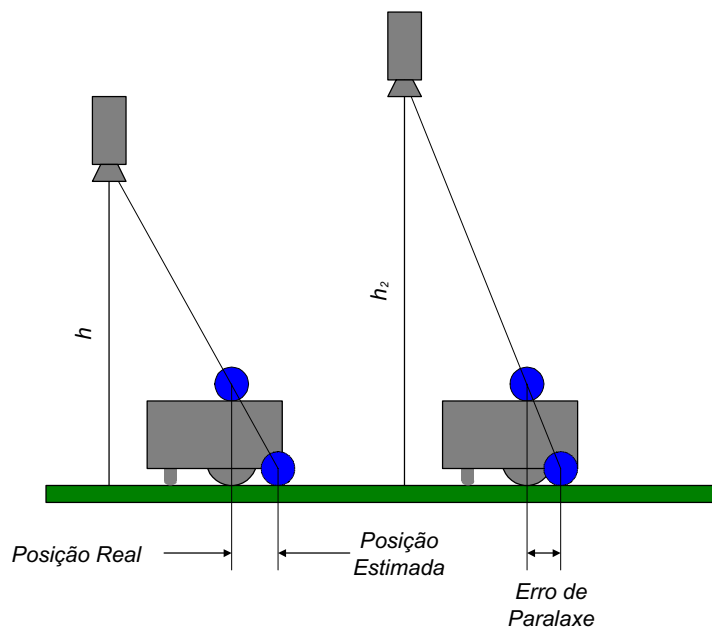


Figura 5.2: O erro de Paralaxe para alturas da câmara diferentes

É claro que este factor pode ser compensado conhecendo a altura a que os marcadores dos robots estão colocados. Esse mecanismo de compensação foi implementado pois possibilita um nível adicional de precisão.

Ocultação da bola. Mais grave, é o problema da ocultação da bola pelo corpo do robot. Para uma câmara posicionada a uma distância insuficiente há situações em que o corpo do robot oculta, parcial ou totalmente, a bola. Tipicamente, esta situação acontece quando estamos perto das extremidades do campo. Lamentavelmente, isso coincide com a maior parte das situações de golo iminente: quando o avançado leva a bola controlada e se aproxima da baliza temos potencialmente uma situação de ocultação. As figuras 5.3 e 5.4 evidenciam a gravidade do problema para uma altura da câmara de 2 m.



Figura 5.3: O Problema da Ocultação da Bola - Na vertical da câmara



Figura 5.4: O Problema da Ocultação da Bola - A 60 cm da vertical da câmara

Campo de visão angular da câmara. Outro problema com distâncias baixas tem a ver com o campo de visão angular da câmara usada. Para capturar todo o campo de jogo, o que implica ser capaz de abranger uma área com cerca de 3 metros de largura, é necessária uma câmara com uma objectiva que apresente um campo de visão com um ângulo superior a 50° . Nem todas as câmaras apresentam essa capacidade. Muitas das que são capazes de abranger essa área apresentam, em maior ou menor grau, um fenómeno de distorção, comumente designado por distorsão em barril. É certo que esse efeito pode ser corrigido recorrendo uma função

de mapeamento pixel-mundo capaz de compensar esse tipo de distorção, mas é sempre uma fonte de incerteza extra.

Naturalmente, a câmara de vídeo deverá fornecer uma imagem a cores pois a maior parte da informação está presente na cor dos marcadores.

- Um sistema de digitalização e aquisição do sinal de vídeo. Este sistema deverá converter o sinal de vídeo para uma representação digitalizada que possa ser usada por um computador.
- Um Computador, tipicamente um PC, capaz de processar a imagem digitalizada e aplicar os algoritmos que a seguir descreveremos, para extrair a informação que procuramos.

Para mais informação acerca do *Hardware* usado, é possível consultar o Apêndice A onde consta uma descrição mais técnica do mesmo.

Como se poderá ver, na generalidade dos exemplos recorreremos a câmaras de baixa qualidade. Essa opção evidencia melhor a robustez ao ruído e às diferentes condições de iluminação dos algoritmos desenvolvidos.

5.1 Dos Pixeis Para o Estado do Sistema

Consideremos a partir de agora que dispomos da representação digital da imagem do campo. Essa representação será constituída por uma matriz de elementos de imagem p_{xy} , normalmente designados por pixeis, contendo cada um uma representação da área de imagem abrangida. Cada pixel, no nosso caso, corresponde a um vector com três componentes que representam a intensidade das três componentes de cor: vermelho, verde e azul. Normalmente estas três componentes são designadas, respectivamente, r , g e b , a partir das suas designações em Inglês. Podemos definir a função:

$$\begin{aligned} rgb : I &\longrightarrow \mathfrak{R}^3 \\ p_{xy} &\longrightarrow (r, g, b) \end{aligned} \tag{5.1}$$

que, para cada pixel, indica a cor associada, sob a forma de um vector com os valores das três componentes de cor fundamentais.

Normalmente, temos essas componentes discretizadas em, respectivamente n_r , n_g e n_b bits. Tipicamente, para $n_r = n_g = n_b = 8$ (total de 24 bits) já se consegue uma representação em que o efeito da discretização é visualmente indetectável. Nesse caso gastaríamos 3 bytes para representar cada pixel. Podemos aceitar o compromisso de poder visualizar algum efeito de discretização e escolher $n_r = n_g = n_b = 5$ (total de 15 bits) ou $n_r = n_b = 5$, $n_g = 6$ (total de 16 bits) para representar um pixel recorrendo apenas a 2 bytes. Dessa maneira podemos reduzir o tamanho da imagem

para $\frac{2}{3}$. Outra vantagem desta opção é que qualquer função que mapeie o vector (r, g, b) num outro vector ou escalar,

$$\begin{aligned} f : \mathfrak{R}^3 &\longrightarrow \mathfrak{R}^n \\ (r, g, b) &\longrightarrow X \end{aligned} \quad (5.2)$$

pode ser representada recorrendo a uma tabela relativamente compacta. Para $n_r = n_g = n_b = 5$ basta uma tabela com 2^{15} entradas. Admitindo que X pode ser representado com n bytes teremos uma tabela de $n2^{15}$ bytes. Para $n = 4$, por exemplo, a tabela terá 128 Kbytes o que é facilmente implementável nos computadores actuais. À medida que o tempo passa, e como a evolução da memória disponível tipicamente segue uma lei de crescimento exponencial, este valor tornar-se-á cada vez menos pesado. Esta vantagem será crucial para permitir uma implementação eficiente dos algoritmos apresentados.

5.2 Sistema de Coordenadas

De um modo geral todas as unidades empregues serão múltiplos ou submúltiplos de unidades do S.I. Como teremos de localizar os robots e a bola no campo é necessário fixar a origem e a orientação dos eixos do sistema de coordenadas a utilizar. Foi escolhido para origem o centro do campo e considera-se o eixo dos xx alinhado longitudinalmente em relação ao campo. O sentido positivo aponta em direcção à baliza adversária. A figura 5.5 ilustra o sistema de coordenadas empregue.

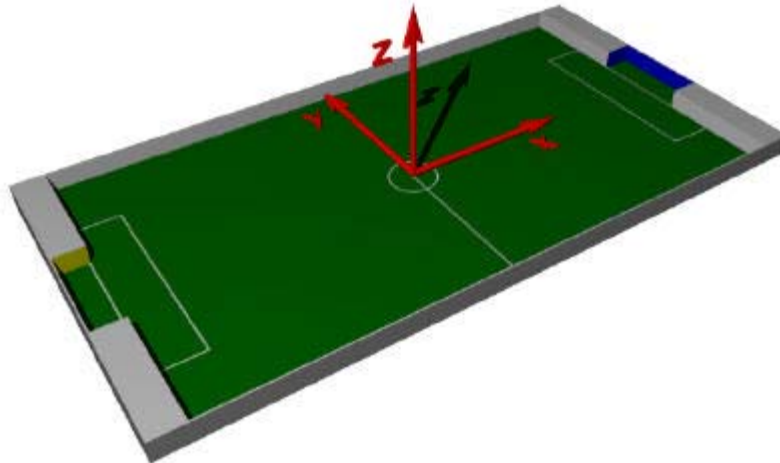


Figura 5.5: O campo com o respectivo sistema de coordenadas

Assumir que o ataque se processa segundo o sentido positivo do eixo dos xx não é limitativo na medida em que, para implementar uma mudança

de lado do campo, basta intervir na função que mapeia as coordenadas da imagem em coordenadas do mundo.

Um caso particular pode ser encontrado nas coordenadas de cada pixel na matriz de elementos da imagem. Cada pixel p_{xy} poderá ser encontrado numa matriz em que x indicará a linha e y a coluna do pixel. No nosso caso o pixel com coordenadas $(0, 0)$ será o pixel do canto superior esquerdo e o pixel com coordenadas $(x_m - 1, y_m - 1)$ encontra-se no canto inferior direito. Este sistema de coordenadas reflete o facto de uma câmara PAL explorar a imagem em linhas que a varrem da esquerda para a direita, numa sequência em que a primeira linha se encontra no topo e as seguintes vão descendo. Nota: no nosso caso para o conjunto câmara PAL e placa digitalizadora temos $x_m = 284$ e $y_m = 288$.

Para facilitar a distinção entre o caso em que estamos a falar de coordenadas (x, y) correspondentes a uma localização no mundo ou correspondentes ao pixel de coordenadas (x, y) definiremos os conjuntos Nx e Ny :

$$Nx = \{0, 1, \dots, x_m - 1\} \quad (5.3)$$

e

$$Ny = \{0, 1, \dots, y_m - 1\} \quad (5.4)$$

Estes conjuntos definem os valores que x e y podem tomar, como coordenadas na imagem.

5.3 Estado do Sistema

Consideremos também o cenário típico de um desafio de futebol robótico, versão liga F-180, do Robocup. Nesta situação temos duas equipas de robots, a que nós controlamos e a equipa adversária., que naturalmente nós não controlamos, com respectivamente n_t e n_a robots em campo. Este número está actualmente configurado para um máximo de cinco robots por equipa. Porém, haverá casos em que n_t e n_a podem tomar valores inferiores pois há equipas que apresentam menos robots para um dado jogo ou, o que também é frequente, há fases do jogo em que uma equipa se vê forçada a retirar do jogo um ou mais robots. Isso pode acontecer devido a problemas técnicos ou porque foi penalizada com a expulsão de robots. Por tudo isto, é necessário que o sistema de visão esteja preparado para lidar com valores de n_t e $n_a \in \{1, 2, 3, 4, 5\}$. Normalmente, a alteração do número de robots acontece durante uma paragem do jogo, durante a qual é possível alterar parâmetros do sistema. Assim é viável aceitar que essa alteração não tenha de ser automaticamente reconhecida pelo sistema.

Para além dos robots temos a bola. A sua localização e velocidade constituem uma parte crucial da informação que devemos conseguir extrair da imagem. Embora a bola seja ligeiramente maior que os marcadores oficiais

e portanto ligeiramente mais fácil de extrair da imagem, pode ser parcialmente encoberta pelos robots. Isso faz com que haja situações em que a sua visibilidade seja inferior à de um marcador. Outro factor de perturbação é o facto de a bola tipicamente se mover a velocidades superiores à dos robots. Isso torna mais complexo o recurso a técnicas baseadas em localidade para encontrar a bola.

Quais são então as componentes do estado do sistema global?

Temos o vector de estado do sistema global:

$$X^T = [X_b^T \ X_{r_1}^T \ \dots \ X_{r_5}^T \ X_{a_1}^T \ \dots \ X_{a_5}^T] \quad (5.5)$$

Em que X_b^T é o vector de estado da bola, $X_{r_i}^T$ é o vector de estado do robot i e $X_{a_i}^T$ é o vector de estado do robot adversário i .

Para o vector de estado da bola teremos:

$$X_b^T = [x_b \ y_b \ v_{bx} \ v_{by}] \quad (5.6)$$

Em que x_b e y_b representam a posição da bola enquanto v_{bx} e v_{by} são as duas componentes da velocidade. Este estado corresponde a um modelo simplificado da bola mas verificámos que a complexidade extra de modelos mais realistas não se traduzia em ganhos apreciáveis na precisão das estimativas.

Cada robot da nossa equipa tem o estado $X_{r_i}^T$ constituído por 5 componentes

$$X_{r_i}^T = [x_{r_i} \ y_{r_i} \ \theta_{r_i} \ v_{r_it} \ v_{r_in} \ \omega_{r_i}] \quad (5.7)$$

Neste modelo v_{r_it} representa a componente da velocidade orientada segundo a direcção indicada por θ_{r_i} . Já v_{r_in} será a componente da velocidade orientada segundo a direcção perpendicular à definida por θ_{r_i} .

Normalmente v_{r_in} deverá ser zero pois teremos deslizamento lateral apenas quando:

- Um robot desliza encostado a um obstáculo.
- O robot está a ser arrastado por outro robot.
- A trajectória que o robot tenta descrever é demasiado apertada e como não dispõe de aderência suficiente, escorrega lateralmente.

Os dois primeiros casos são situações indesejadas ou mesmo penalizadas pelas regras e por isso não deverão constituir situações frequentes durante o jogo.

O terceiro caso pode ser minimizado tanto ao nível da construção do robot como da escolha das trajectórias admissíveis. No entanto é vantajoso dispor da informação acerca do estado de escorregamento lateral. Poderemos

usar essa informação para detectar situações anómalas que podem requerer acções especiais quer ao nível do controlo quer da decisão.

A definição do modelo dos robots adversários é mais difícil devido à variedade de configurações possíveis e à menor informação disponível durante o jogo. Dado que as regras definem apenas a presença de um marcador oficial podemos não dispor de informação acerca da orientação do robot. Mesmo dispondo dessa informação, ela pode ser irrelevante caso o robot seja capaz de movimentação omnidireccional. Por esse motivo, o estado de um $X_{a_i}^T$ robot adversário será dado por:

$$X_{a_i}^T = [x_{a_i} \ y_{a_i} \ v_{a_ix} \ v_{a_iy}] \quad (5.8)$$

5.4 Mapeamento Imagem - Mundo

Como podemos ver na figura 5.6, a câmara introduz alguma distorsão na imagem capturada. Temos claramente visíveis o efeito de distorsão em barril e o mapeamento trapezoidal introduzido pela perspectiva.



Figura 5.6: A imagem capturada pela câmara colocada sobre campo

Se queremos, a partir da imagem, extrair as localizações dos objectos observados temos de construir a função:

$$m : \begin{array}{l} Nx \times Ny \longrightarrow \mathbb{R}^2 \\ (u, v) \longrightarrow (x, y) \end{array} \quad (5.9)$$

que mapeia coordenadas da imagem em coordenadas no mundo. Esta função apenas fornece as coordenadas x , y assumindo que a componente em z é zero. Dado que nem todos os objectos estão a essa altura, haverá um certo erro induzido por fenómenos de paralaxe. Esse erro pode ser compensado caso conheçamos a coordenada z do objecto em questão. Com esse valor podemos construir uma função que ajusta x , y levando em conta a geometria da cena. Porém, esse ajuste só pode ser feito após a correcta identificação do objecto em questão e por isso não pode ser aplicado a este nível. A função m pode ser associada ao mapeamento entre o pixel da imagem e a intersecção com o plano xy , da recta, definida em \mathbb{R}^3 , que passa pelo "centro" da câmara e pela geratriz da pirâmide constituída por todos os raios de luz que iluminam o pixel. Para que m seja capaz de compensar os efeitos de distorsão mencionados não pode ser linear. Por isso, introduzimos termos de ordens mais elevadas [22]. A presença destes termos permite compensar efeitos não muito graves de distorsão de barril, perspectiva e rotação.

$$\begin{aligned} m_x(u, v) &= a_1 + a_2u + a_3v + \\ &\quad + a_4u^2 + a_5v^2 + a_6uv + a_7u|u| + a_8uv^2 \end{aligned} \quad (5.10)$$

$$\begin{aligned} m_y(u, v) &= b_1 + b_2u + b_3v + \\ &\quad + b_4u^2 + b_5v^2 + b_6uv + b_7v|v| + b_8vu^2 \end{aligned} \quad (5.11)$$

Temos assim a_1, a_2, \dots, a_8 e b_1, b_2, \dots, b_8 , num total de 16 parâmetros que devem ser estimados para uma dada posição da câmara. Para os estimar são necessários, pelo menos, 8 pontos de calibração. Naturalmente, recorrendo a mais pontos é possível melhorar a precisão desta calibração. Verificou-se que, desta maneira, era possível obter um mapeamento relativamente preciso.

Uma outra vantagem da estrutura escolhida para a função m é que podemos recorrer a uma função com a mesma estrutura para aproximar a sua inversa. Assim, a função w :

$$\begin{aligned} w : \mathbb{R}^2 &\longrightarrow \mathbb{R}^2 \\ (x, y) &\longrightarrow (u, v) \end{aligned} \quad (5.12)$$

que aproxima m^{-1} será:

$$\begin{aligned} w_u(x, y) &= c_1 + c_2x + c_3y + \\ &\quad + c_4x^2 + c_5y^2 + c_6xy + c_7x|x| + c_8xy^2 \end{aligned} \quad (5.13)$$

$$\begin{aligned} w_v(x, y) &= d_1 + d_2x + d_3y + \\ &\quad + d_4x^2 + d_5y^2 + d_6xy + d_7y|y| + d_8yx^2 \end{aligned} \quad (5.14)$$

Podemos ver na figura 5.7 a imagem obtida por uma das câmaras onde é mais evidente o problema da distorsão em barril. Recorrendo aos pontos de calibração, que estão marcados na imagem, formulamos o problema de estimação como um problema de mínimos quadrados.



Figura 5.7: Os pontos de calibração do mapeamento de coordenadas

Na figura 5.8 podemos ver como uma grelha em coordenadas do mundo é mapeada em coordenadas da imagem. Desta forma o efeito da distorsão em barril é compensado.

5.4.1 A Definição da Escala

Uma questão interessante é a que se põe quando queremos definir a escala da imagem. Isto é, quando queremos indicar as dimensões de um pixel. Não há uma resposta bem definida pois os fenómenos de distorsão mencionados alteram as dimensões de um pixel de uma maneira que é dependente das suas coordenadas. Felizmente, esta variação não é muito grave e não estaremos a cometer uma imprecisão grave se mencionarmos um valor médio para a escala.

5.5 Correção da paralaxe

A imagem obtida, na prática, apenas consegue disponibilizar informação acerca do posicionamento dos diferentes objectos no plano xy . Teoricamente, dado que conhecemos as dimensões dos marcadores e da bola, seria possível



Figura 5.8: A compensação da distorsão em barril

extrair da imagem informação acerca da distância a que estes se encontram. Mas, dada a resolução a que estamos a trabalhar, a precisão possível é demasiado baixa para que se obtenha alguma informação útil. Aliás, a única função para essa informação é a correcção do efeito de paralaxe. Dado que, em princípio, já é conhecida a altura a que os marcadores e a bola devem ser encontrados, podemos aproveitar essa informação logo que o objecto tenha sido identificado. Assim conhecendo z podemos corrigir o valor de \tilde{x} , \tilde{y} para obter x , y já compensados no que toca à paralaxe.

A função L que implementa esta compensação pode ser descrita por:

$$L : \mathfrak{R}^3 \longrightarrow \mathfrak{R}^2$$

$$(\tilde{x}, \tilde{y}, z) \longrightarrow (x, y) \quad (5.15)$$

com

$$x = L_x(\tilde{x}, \tilde{y}, z) = \tilde{x} \left(1 - \frac{z}{h}\right) \quad (5.16)$$

$$y = L_y(\tilde{x}, \tilde{y}, z) = \tilde{y} \left(1 - \frac{z}{h}\right) \quad (5.17)$$

onde h representa a altura a que está colocada a câmara.

5.6 Classificação das Cores

Um dos elementos cruciais do sistema de interpretação da imagem é a função que classifica os pixels baseada na cor que ele apresenta. Consideremos as n_{col} cores associadas aos objectos a extrair da imagem. Podemos classificar cada pixel indicando o grau de pertença, da cor que ele representa, ao conjunto de cada uma dessas n_{col} cores. A função que implementa essa operação pode ser construída a partir das funções que definem os conjuntos fuzzy associados às cores que elegemos seguir.

Consideremos o espaço das cores possíveis para um pixel. Como vimos esse espaço, assemelha-se a um cubo definido entre $(0, 0, 0)$ e $(2^{n_r}, 2^{n_g}, 2^{n_b})$. Como os valores, que cada componente de cor pode tomar, estão discretizado, teremos uma grelha tridimensional com todos os triplos inteiros contidos nesse cubo. Quando falarmos da cor de um pixel pode-se ler: o ponto no espaço rgb definido pelas componentes de cor do pixel.

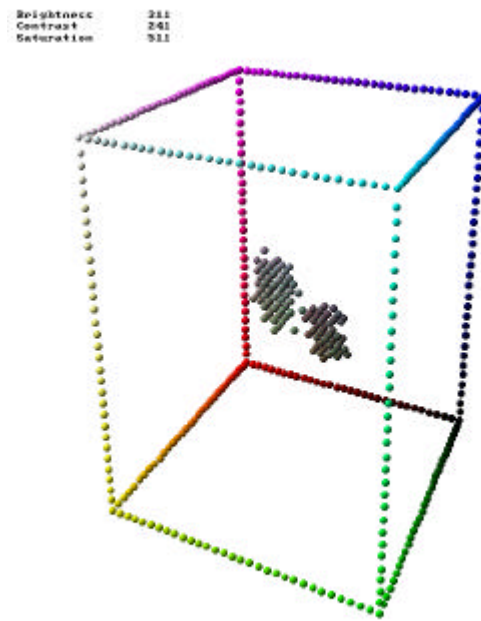


Figura 5.9: O Espaço rgb e algumas regiões desse espaço

Considerando o nosso universo U_{rgb} podemos definir o conjunto F constituído pelas "cores que a bola de golfe laranja apresenta na imagem". Esta região é representada por um conjunto *fuzzy* com mais propriedade do que por um conjunto clássico. Para diferentes localizações, e consequentemente

diferentes condições de iluminação, a bola pode apresentar uma cor ligeiramente diferente. E se há cores que inequivocamente podemos dizer que pertencem ao conjunto F , já haverá outros "laranjas" que só podem ser associados em menor grau a esse conjunto. Evita-se assim o paradoxo da duna de areia[16] e ganha-se uma medida que indica a certeza que temos que um dado pixel provenha de uma bola de golfe laranja.

Podemos pois definir uma série de conjuntos fuzzy F_i associados às cores dos marcadores que queremos localizar.

5.7 Estimação dos Conjuntos Associados aos Marcadores

Um dos problemas mais delicados a resolver é o de conseguir identificar os conjuntos F_i . Verificamos que a forma desses conjuntos era dependente de inúmeros factores:

- O tipo e intensidade de iluminação.
- A câmara usada.
- Os ajustes na câmara do tempo de exposição, abertura do diafragma, nível de zoom e outros parâmetros ópticos.
- Os ajustes da intensidade, contraste e saturação da placa de aquisição.

Qualquer mudança num destes factores leva a ter de redefinir completamente todos os conjuntos F_i . É certo que, passada a fase de instalação, é razoável esperar que estes factores se mostrem relativamente estáveis. De qualquer modo pareceu-nos importante delinear um processo de estimação do mapeamento das cores dos marcadores que fosse:

- Relativamente rápido de efectuar.
- Simples de operar.
- Interactivo.
- Incremental.
- Flexível.

Todas estas características são importantes pois a qualidade da definição dos conjuntos F_i é crucial para o bom funcionamento de todo o sistema. Note-se que não é necessário inferir que por isso este sistema possa ser menos robusto, por exemplo a variações na iluminação. Se elas são esperadas e muitas vezes inevitáveis elas podem ser incorporadas na definição de F_i . Isso

pode ser feito facilmente se o esquema de estimação possuir as características que mencionamos. Como pode tal ser conseguido?

Naturalmente teremos de colectar dados reais para fazer esta estimação. Recorrendo a uma sequência de imagens típicas de um cenário de jogo podemos, a partir de indicações dadas preferencialmente de uma forma interactiva, estabelecer o posicionamento dos marcadores que queremos seguir. A partir dessa informação podemos construir um histograma que acumula o aparecimento de cada cor durante a sequência de imagens indicada.

5.7.1 Construção do Histograma

Para definir um conjunto F_i é necessário construir previamente um histograma que acumule as ocorrências de cada cor em pixels que se sabe que representam o i -ésimo marcador. Esta acumulação deverá ser repetida para diferentes posições do marcador. Assim poderemos abranger uma variedade de casos que representem a maioria das situações que encontraremos durante um jogo. Podemos considerar este histograma H_i como uma função:

$$\begin{aligned} H_i : \mathbb{R}^3 &\longrightarrow \mathbb{R}_0^+ \\ (r, g, b) &\longrightarrow h \end{aligned} \quad (5.18)$$

que acumula, para cada ponto do espaço rgb a quantidade de vezes que foi observado pesada pela relevância de cada uma dessas observações. Tipicamente, durante a definição de H_i consideramos os pontos no centro do marcador com relevância unitária, decrescendo esse valor até zero perto das bordas. Normalmente, nessa zona, a cor observada não é tão adequada pois não está iluminada tão uniformemente, está contaminada por luz refletida do campo e já se verificam fenómenos de *aliasing* com as cores vizinhas. Assim, partindo de um histograma inicial H_i^0 que associa zero a cada ponto do espaço rgb , podemos construir uma sequência H_i^0, H_i^1, \dots de histograma em que passamos de um histograma H_i^n para o histograma H_i^{n+1} após uma das duas operações:

- Observação de um marcador. Nesta operação, um conjunto fuzzy de pixels o_i , que se sabe representar a imagem do marcador em questão, é usado para actualizar o histograma. Teremos:

$$H_i^{n+1}(r, g, b) = H_i^n(r, g, b) + m_{o_i}(r, g, b) \quad (5.19)$$

- Eliminação de cores. Neste caso iremos usar um conjunto fuzzy de pixels n_i , que se sabe *não* representar a imagem do marcador em questão, para actualizar o histograma. Definindo:

$$H_i^{n+1}(r, g, b) = H_i^n(r, g, b) - m_{n_i}(r, g, b) \quad (5.20)$$

Teremos:

$$H_i^{n+1}(r, g, b) = \begin{cases} H_i^{n+}(r, g, b) & \text{se } H_i^{n+}(r, g, b) > 0 \\ 0 & \text{se } H_i^{n+}(r, g, b) \leq 0 \end{cases} \quad (5.21)$$

Esta última operação é muito importante pois permite eliminar pontos do espaço rgb que, apesar de aparecerem na imagem do marcador, também aparecem em outras zonas da imagem. A sua presença levaria a classificar pixels dessas zonas como potenciais pixels do marcador. Isso daria origem à presença de marcadores fantasma que poderiam, em certos casos, ser confundidos com os verdadeiros marcadores. Outra desvantagem destes potenciais pixels fantasma é o acréscimo de pontos a processar após a primeira operação de classificação.

5.7.2 Estimação de um Conjunto Associado à Cor

Dispondo do histograma H_i podemos estimar o conjunto F_i recorrendo à função r . Esta função está definida:

$$\begin{aligned} r : \mathfrak{R}_0^+ &\longrightarrow [0, 1] \\ h &\longrightarrow m \end{aligned} \quad (5.22)$$

com

$$r(h) = \begin{cases} 0 & \text{se } h < T_m \\ \frac{(h-T_m)}{(T_M-T_m)} & \text{se } T_m \leq h \leq T_M \\ 1 & \text{se } h > T_M \end{cases} \quad (5.23)$$

Para conhecer F_i devemos obter m_{F_i} , para tal compomos a função r com H_i :

$$m_{F_i}(r, g, b) = r(H_i(r, g, b)) \quad (5.24)$$

A função r está definida em três segmentos. O efeito do primeiro segmento é o de uma zona morta. Desta maneira podemos desprezar cores que, apesar de mostrarem um valor positivo no histograma, não consideramos significativas. Certas perturbações, durante a fase de calibração, podem levar a que apareçam cores no histograma que não representam o marcador em questão. Há outro problema que esta zona morta pode resolver. Não é vantajoso classificar cores que, apesar de representarem com alguma legitimidade um marcador, também representam outro marcador. Caso a sua presença num dos histogramas seja pouco significativa, a sua influência no conjunto F_i pode ser eliminada graças ao efeito da zona morta. Para isso basta que escolhamos um limiar T_m superior à sua contagem em H_i .

O terceiro segmento pode ser visto como uma saturação. O seu propósito é o de nivelar cores que consideramos representarem perfeitamente o marcador mas que têm valores diferente no histograma devido à sequência de calibração ter favorecido certas posições.

5.7.3 Medida da Qualidade da Definição dos Conjuntos Associados às Cores

Caso as cores que procuramos identificar estejam bem separadas no espaço rgb , verificaremos que:

$$\forall i \neq j, F_i \cap F_j = \phi \quad (5.25)$$

Esta é a situação ideal. Na prática poderá haver alguma sobreposição mas pelo menos esperamos que, dados os conjuntos C_{ij} :

$$C_{ij} = F_i \cap F_j, \quad i, j \in \{1, 2, \dots, n_{col}\} \quad (5.26)$$

se verifique:

$$\forall i \neq j, M(C_{ij}) < s_{\max} \quad (5.27)$$

Com s_{\max} como o limiar de sobreposição que consideramos aceitável.

Podemos ver nas figuras 5.10-5.12 uma representação dos conjuntos F_0, \dots, F_3 correspondentes respectivamente: à bola (zona laranja), marcador oficial amarelo (zona amarela), marcador secundário violeta (zona violeta) e marcador oficial azul (zona azul).

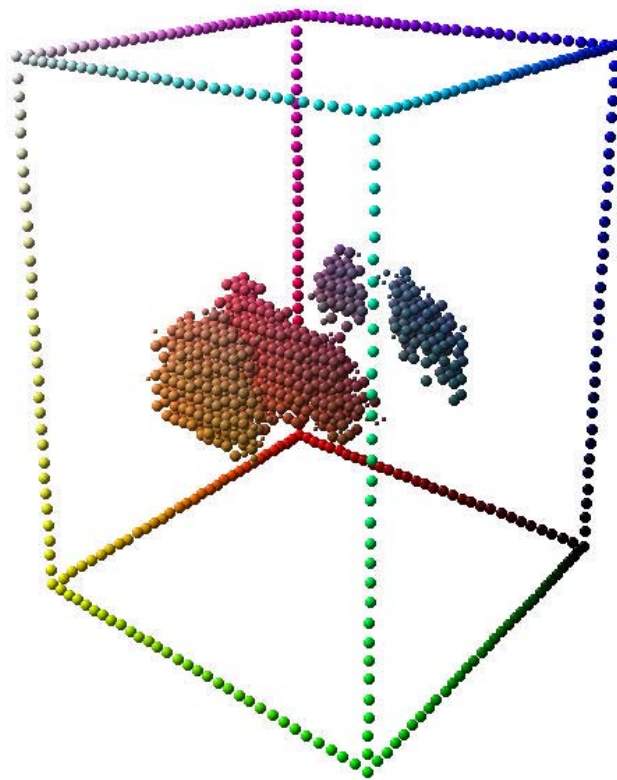


Figura 5.10: Representação dos Conjuntos F_0, \dots, F_3

Por uma questão de melhor visualização apresentamos os mesmos conjuntos agora sob ângulos diferentes.

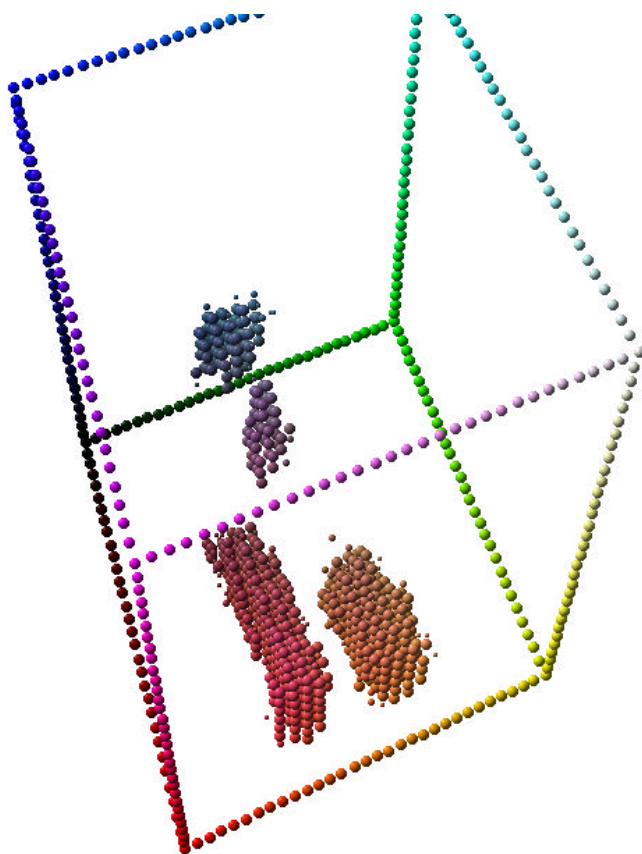


Figura 5.11: Representação dos Conjuntos F_0, \dots, F_3 segundo outro ângulo

Estas figuras ilustram a separação entre os conjuntos que é fundamental para uma correcta classificação dos marcadores. Note-se também as diferentes formas dos conjuntos que não são facilmente definíveis recorrendo exclusivamente a um esquema paramétrico.

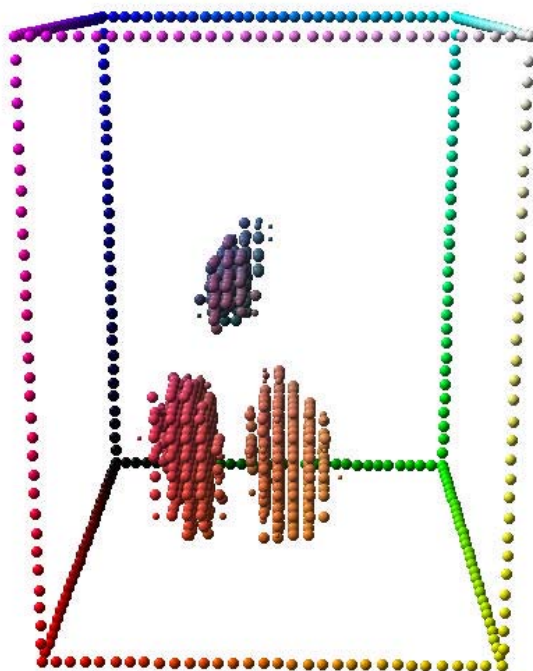


Figura 5.12: Representação dos Conjuntos F_0, \dots, F_3 evidenciando a separação amarelo-laranja

5.8 Selecção da Área Activa

Haverá sempre zonas da imagem que representam áreas para as quais não há interesse em procurar marcadores. O campo não ocupa toda a imagem e tipicamente há uma banda no topo e no fundo que mostra apenas o chão que rodeia o campo. Naturalmente que essa área é desinteressante no que respeita à localização dos robots e da bola. Definamos a área activa como o conjunto clássico A_I que contém a coordenadas dos pixeis onde iremos procurar marcadores. Surge a questão de como representar esse conjunto. Poderíamos, à partida, construir uma função V :

$$\begin{aligned} V : Nx \times Ny &\longrightarrow \{0, 1\} \\ (x, y) &\longrightarrow v \end{aligned} \quad (5.28)$$

que indicaria se uma dado pixel p_{xy} pertence ou não a A_I . Porém, essa solução não poupava uma pesquisa a *todos* os pixeis da imagem nem que fosse apenas para verificar que o dado pixel deveria ser ignorado. Podemos conseguir um resultado mais interessante se assumirmos o seguinte pressuposto:

- A área que queremos activa, isto é A_I , é convexa. Na realidade, para os efeito que queremos nem precisamos de uma restrição tão forte, basta que a área em questão seja convexa para linhas horizontais. Isto é, dados dois pixeis de coordenadas (x_1, y) e (x_2, y) pertencentes a A_I , se verifique:

$$\forall \alpha \in [0, 1], (x, y) \in A_I \text{ com } (x, y) = \alpha(x_1, y) + (1 - \alpha)(x_2, y) \quad (5.29)$$

Desta forma podemos, para cada linha y , indicar duas coordenadas x_1 e x_2 que marcam o início e o fim da área activa para essa linha. Isto equivale a construir uma função V_A :

$$\begin{aligned} V_A : Ny &\longrightarrow Nx \times Nx \\ y &\longrightarrow (x_1, x_2) \end{aligned} \quad (5.30)$$

Como a câmara estará fixa e o campo também, esta função V_A , só terá de ser definida durante a calibração inicial e poderá ser facilmente representada por uma tabela.

A principal vantagem desta representação do conjunto A_I é que já não obriga a interrogar cada pixel para verificar se ele deve ser processado ou não. Consideremos um cenário em que o processamento está a ser feito segundo um varrimento de linhas, desde a primeira no topo e pesquisando os pixeis da esquerda para a direita:

- Uma linha que não contenha pixeis para serem processados pode ser imediatamente esquecida logo que se verifique essa situação. Para isso basta a avaliação da função V_A para essa linha indicar $x_2 < x_1$. Todos os pixeis dessa linha podem ser descartados com uma operação.

- Numa linha em que temos alguns pixels para tratar, podemos avançar imediatamente para o pixel x_1 descartando todos os pixels antes e podemos parar o processamento quando atingirmos x_2 . Aí descartamos automaticamente os pixels que faltam para acabar a linha.

Usando esta representação, pixels da imagem que não queiramos processar, praticamente não ocupam tempo de processamento. O tempo de processamento tomado será então proporcional ao número de pixels em A_I e não ao número total de pixels da imagem.



Figura 5.13: Imagem Capturada completa

Numa situação típica, verificamos que a parte activa da imagem pode abranger apenas 65% do total dos pixels da imagem, com o consequente ganho no tempo de processamento. Outra vantagem, que por si só já justificaria a restrição do processamento da imagem a uma zona activa, é que poderemos ter no cenário todo o tipo de objectos que não haverá interferência mesmo que as suas cores coincidam com as dos marcadores.

É certo que há casos em que o pressuposto de convexidade horizontal não pode ser tomado. Por exemplo, quando queremos usar informação já disponível para restringir a procura dos marcadores. Nesse caso teremos várias elipses ou rectângulos espalhados pelo campo indicando os conjuntos de pixels que devemos considerar. Em todo caso podemos usar estas técnicas cumulativamente, primeiro V_A é usada para extrair os pixels relativos ao campo e depois um refinamento extra pode ser conseguido com uma outra definição da área activa.



Figura 5.14: Parte Activa da Imagem Capturada

5.9 Classificação e Ordenação

Dispondo de F_i conjuntos associados às cores dos marcadores que queremos localizar podemos agora percorrer a imagem classificando cada um dos seus pixels de acordo com o seu grau de pertença a cada uma das n_{col} cores.

O universo onde F_i está definido é constituído pelo espaço rgb , ou seja um cubo limitado entre $(0, 0, 0)$ e $(2^{n_r}, 2^{n_g}, 2^{n_b})$. Teremos estimada, para cada um dos conjuntos F_i , uma função m_{F_i} que, de acordo com a equação 3.1, definirá esses conjuntos. Esta função será do tipo indicado em 5.2 com $X = [0, 1]$.

Consideremos agora o universo U_{img} constituído por todos os pixel p_{xy} da imagem que queremos classificar. Neste universo podemos definir uma série de conjuntos G_i . Um pixel pertencerá ao conjunto G_i em maior ou menor grau conforme o grau de pertença da sua $côr$ ao conjunto F_i . A função que define cada G_i será então dada por:

$$m_{G_i}(p_{xy}) = m_{F_i}(rgb(p_{xy})) \quad (5.31)$$

É possível considerar o grau de pertença de um pixel ao conjunto G_i como indicando quão parecido esse pixel é com um pixel que representa um marcador da $côr$ i .

Podemos observar que esta classificação tem uma acção perfeitamente local, não utilizando qualquer informação proveniente dos pixels vizinhos. Deste facto, em termos do nosso objectivo de identificar os marcadores em cena, decorrem algumas vantagens e desvantagens.

A principal desvantagem é que estamos a perder informação, pois não introduzimos dados que podem ajudar a classificar um pixel:

- Um pixel que apresenta uma cor parecida com a que queremos mas que está isolado na imagem não é um candidato tão bom como um outro equivalente mas rodeado de mais pixels compatíveis.
- Um pixel que parece bom (apresenta uma cor parecida com a que queremos) mas está rodeado de outros pixels que são claramente incompatíveis, isto é bons candidatos mas a outra cor. Neste caso já não deveria ser considerado tão bom.

As vantagens por outro lado são muito importantes:

- Esta operação, efectuada baseada apenas em um pixel pode ser realizada de uma forma muito mais expedita. Este é um factor muito importante pois é a operação que terá de ser aplicada a *todos* os pixels da imagem que queiramos classificar.
- Como não sofre de dependências locais ou do estado dos pixels já processados é facilmente paralelizável e sequenciável. Sequenciável, significa aqui que podemos aplicar esta classificação imediatamente após a disponibilização do pixel. Deste modo é possível completar uma parte significativa do processamento durante o tempo que está comprometido para a aquisição da imagem para a memória do computador.

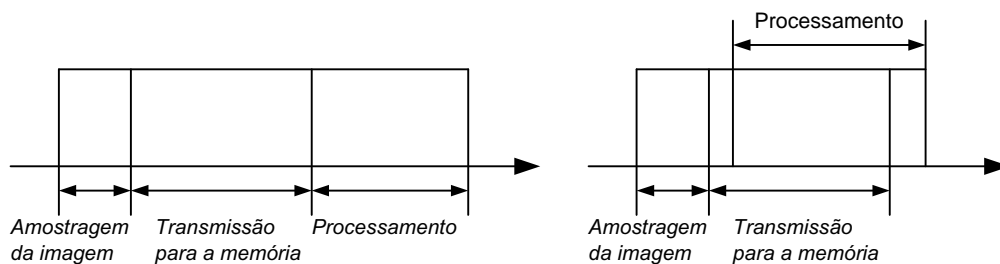


Figura 5.15: Sequências de Processamento

Temos agora para cada pixel p_{xy} um vector $G = (c_1, c_2, \dots, c_{n_{col}})$ que indica o grau de pertença desse pixel a cada um dos conjuntos G_i . Desde que a calibração tenha sido correctamente feita e se verifique a condição 5.27 estaremos num dos seguintes casos:

- O pixel é desinteressante (leia-se: não representa nenhuma das cores que estamos a extrair da imagem) e por isso todas as componente de

G terão valores baixos. Mais precisamente:

$$|G|_{\infty} < c_{\min} \quad (5.32)$$

Em que c_{\min} representa limiar abaixo do qual o grau de pertença é considerado desprezável.

Neste caso trata-se de um pixel associado a uma porção de imagem que não contém nenhum dos marcadores que procuramos e por isso não devemos ocupar-nos mais dele.

- O pixel apresenta um grau de pertença a um dos conjuntos G_j significativa. Essa componente será claramente superior às outras (pelo menos na maior parte dos casos). Neste caso sabemos que se trata de um pixel que é provavelmente originário de uma zona da imagem ocupada por um marcador.

Em conjunto, a aplicação deste procedimento corresponde a construir novos conjunto \tilde{G}_i aplicando um corte- α fuzzy:

$$\tilde{G}_i = fac(G_i) \quad (5.33)$$

De um modo geral podemos considerar que este conjunto de operações realiza uma classificação seguida de uma selecção. Assim, a dimensão dos dados a processar é consideravelmente reduzida. Por este crivo só passarão os pixels que apresentem alguma semelhança no que toca às cores dos marcadores. Esta será a única operação que lida com a totalidade dos pixels candidatos. Todas as outras operações lidarão quase exclusivamente com os pixels pertencentes aos conjuntos \tilde{G}_i . Esta é a chave para possibilitar o processamento em tempo real da imagem e para possibilitar a extracção da posição dos marcadores num intervalo de tempo útil.

5.10 Agregação e Afectação dos Pixels a Marcadores

Até agora, todas as operações efectuadas não levaram em conta qualquer propriedade relacionada com a localidade dos pixels. Para o processamento apresentado isso mostrou-se uma vantagem pois flexibilizou a aplicação dessas operações. Porém há informação proveniente da localidade, absoluta ou relativa, dos pixels que devemos considerar. Neste momento, dada a menor quantidade de pixels a processar torna-se eficiente introduzir essa informação.

Queremos pois passar de um conjunto \tilde{G}_i de pixels que pensamos que podem estar associados a marcadores da cor i , para uma série de conjuntos

$G_i c_j$ onde estão apenas pixels relativos ao marcador j da cor i . Este é um problema clássico de *clustering* mas que, mais uma vez deverá ser resolvido com restrições de tempo real. Verificámos que era possível conseguir resultados satisfatórios com um algoritmo extremamente rápido que pode ser classificado como um algoritmo guloso. Há que levar em conta o facto de podermos definir com uma boa certeza um majorante para o tamanho de um grupo de pixels associado a um dado marcador. Mais uma vez, baseando-nos no facto de a câmara estar fixa em relação ao campo e de as dimensões de um marcadores serem relativamente independentes da sua posição no campo, podemos indicar um valor $R_i m$ tal que:

$$\forall j, \forall p_1, p_2 \in G_i c_j, l^2(p_1, p_2) < R_i m \quad (5.34)$$

Ou seja a distância máxima entre dois pixels pertencentes ao mesmo marcador deverá ser sempre inferior a $R_i m$. Desta maneira podemos decidir facilmente que um pixel não deverá pertencer a um dado marcador, medindo a sua distância ao conjunto de pixels associado a esse marcador. Uma maneira semelhante, embora não totalmente equivalente, de fixar a distância máxima é ir construindo uma estimativa da posição (x_{ij}, y_{ij}) do centro do marcador e usar um valor $CR_i m$ que majore a distância de um pixel candidato a esse centro. A vantagem desta segunda técnica é que evita o cálculo de múltiplas distâncias para o teste de um pixel contra cada conjunto.

5.11 Qualidade de um Marcador

Dispondo de um conjunto $G_i c_j$ onde esperamos encontrar os pixels relativos ao marcador j da cor i , põe-se a questão de saber até que ponto este conjunto representa um marcador e que precisão podemos esperar de uma estimativa da posição do marcador que seja extraída daí. Mais concretamente, do conjunto *fuzzy* de pixels associados a um marcador, queremos saber como extrair uma estimativa da posição do marcador e também uma estimativa da covariância associada à posição. Trata-se de um estimador que, a partir de uma realização que dá ordem a um conjunto *fuzzy*, poderá extrair as coordenadas e a respectiva covariância da posição do centro do marcador. Numa primeira aproximação assumimos que, sendo cada pixel do conjunto $G_i c_j$ uma observação do marcador, o estimador das coordenadas será:

$$\hat{x}_{ij} = \sum_x p_{xy}, p_{xy} \in G_i c_j \quad (5.35)$$

$$\hat{y}_{ij} = \sum_y p_{xy}, p_{xy} \in G_i c_j \quad (5.36)$$

Para estimar a covariância associada:

$$C_{xy,ij} = \begin{bmatrix} c_{xx} & c_{xy} \\ c_{yx} & c_{yy} \end{bmatrix} \quad (5.37)$$

assumimos que:

- A covariância da estimativa da coordenada x era igual à da coordenada y . Atendendo a que a imagem de onde extraímos esta informação apresentava aproximadamente a mesma escala em x e y e dada simetria radial da maioria dos marcadores, esta assunção parece perfeitamente legítima.
- A estimativa da coordenada x era independente da estimativa da coordenada y . Consequentemente $c_{xy} = c_{yx} = 0$.

Ficamos então com:

$$C_{xy,ij} = \begin{bmatrix} c & 0 \\ 0 & c \end{bmatrix} \quad (5.38)$$

Consideramos que, em condições ideais, um conjunto $G_i c_j$ que represente um marcador deverá apresentar um cardinal M^{opt} óptimo. Qualquer desvio em relação a esse valor indicará que se trata de uma observação menos fiável à qual devemos associar uma covariância mais elevada. Dada a matriz de informação correspondente:

$$Y_{xy,ij} = C_{xy,ij}^{-1} \quad (5.39)$$

onde:

$$Y_{xy,ij} = \begin{bmatrix} \frac{1}{c} & 0 \\ 0 & \frac{1}{c} \end{bmatrix} \quad (5.40)$$

usaremos a função:

$$i(x) = \begin{cases} \left(\frac{ek}{2}\right)^2 x^2 e^{-kx} & \text{se } x \geq M^{\min} \\ 0 & \text{se } x < M^{\min} \end{cases} \quad (5.41)$$

com

$$k = \frac{2}{M^{opt}} \quad (5.42)$$

para estimar a matriz de informação:

$$Y_{xy,ij} = \begin{bmatrix} i(M(G_i c_j)) & 0 \\ 0 & i(M(G_i c_j)) \end{bmatrix} \quad (5.43)$$

A constante M^{\min} indica o limiar a baixo do qual escolhemos eliminar por completo a observação. Consideramos que para $M(G_i c_j) < M^{\min}$ se trata de casos em que estamos a observar, não o marcador procurado, mas sim pixels espúrios que podem perturbar significativamente a nossa estimativa.

Conseguimos assim delinear uma estimativa para a posição e respectiva covariância de um dado marcador. Esta informação será muito útil para a suavização das trajectórias que iremos efectuar graças ao recurso a um filtro de Kalman. Nos casos em que $Y_{xy,ij}$ é nula, simplesmente eliminámos a parte da actualização do filtro de Kalman e confiámos apenas na previsão.

5.12 Afecção Marcador-Robot e Marcador-Bola

Para a côm associada à bola não restam muitas dúvidas pois sabemos que só há uma bola em campo. Para conseguir uma estimativa da bola podemos escolher o conjunto $G_{bola}c_j$ que apresente a cardinalidade mais elevada. Isto é que verifique:

$$M(G_{bola}c_j) > M(G_{bola}c_k), \quad \forall j \neq k \quad (5.44)$$

Podemos considerar $M(G_{bola}c_j)$ uma medida da confiança que temos que o conjunto $G_{bola}c_j$ representa uma bola e não se trata de um agrupamento resultante de algum efeito de luz ou de ruído presente na imagem.

Já para o caso dos robots, tanto os da nossa equipa como os da equipa adversária, temos o problema de seleccionar n_{rob} conjuntos $G_{robot}c_j$ que consideramos conter os pixeis que representam os marcadores desses robots. Este problema é diferente se estamos a considerar robots da nossa equipa ou da equipa adversária. Para os nossos robots é muito importante não só conhecer o conjunto das posições dos cinco robots como identificar qual robot ocupa uma dada posição. Esta informação é crítica pois o sistema de controlo deverá fornecer os sinais de controlo a cada robot e qualquer troca será desastrosa.

5.13 Marcadores Adicionais e Correspondente Validação das Observações

O recurso a marcadores adicionais, usando somente as cores preta e branca, permite fornecer, ao sistema de visão, informação extra. Essa informação tem duas funções:

- Por um lado serve de confirmação da nossa identificação do marcador. Se dispusermos, à volta de cada marcador principal, um determinado padrão a preto e branco, podemos usar a presença desse padrão para validar a observação do marcador. Dessa forma temos mais um filtro que eliminará observações espúrias.
- Por outro lado, podemos embeber no padrão à volta do marcador alguma informação. Tipicamente usamos essa possibilidade para identificar univocamente cada robot. Podemos assim conhecer não só a posição mas também qual dos robots se apresenta nessa localização.

Esta técnica por nós introduzida no Mundial de 98 foi rapidamente adoptada por uma variedade de equipas que a aplicaram durante o Mundial de 99.

5.13.1 Algumas Definições Úteis

Definamos \tilde{Y}_{xy} como uma aproximação da luminância do pixel p_{xy} :

$$\tilde{Y}_{xy} = \frac{1}{3} l^1 (rgb(p_{xy}), (0, 0, 0)) \quad (5.45)$$

ou mais simplesmente escrevendo em termos das componentes:

$$\tilde{Y}_{xy} = \frac{1}{3} (r(p_{xy}) + g(p_{xy}) + b(p_{xy})) \quad (5.46)$$

É certo que esta é uma aproximação da luminância. Na realidade a luminância Y é definida[20] por:

$$Y = 0.30r + 0.59g + 0.11b \quad (5.47)$$

onde r , g e b são as intensidades das componentes vermelha, verde e azul.

Porém, para o efeito que procuramos, a aproximação \tilde{Y} é perfeitamente adequada e o seu cálculo significativamente mais expedito. De notar que se para pixels p_{xy} cuja cor associada varia apenas entre tons de cinzento, desde o branco até ao preto, verifica-se que:

$$r = g = b \quad (5.48)$$

e então temos:

$$\tilde{Y}_{xy} = Y_{xy} \quad (5.49)$$

Mesmo para cores onde a equação 5.48 não se verifica exactamente, mas as componentes rgb têm valores semelhantes teremos \tilde{Y} como uma boa aproximação de Y .

Consideremos agora o conjunto $\hat{A}^p(x, y, d)$ que contém todos os pixels cujos centros se encontram a uma distância ao ponto (x, y) inferior a d . Ou seja:

$$\hat{A}^p(x, y, d) = \{p_{xy} : l^p(c(p_{xy}), (x, y)) < d\} \quad (5.50)$$

Este conjunto extrai os pixels que influenciam uma zona da imagem centrada em x, y e de raio d .

Para valores baixos de d , isto é na ordem de poucos pixels, esta definição não reconhece o efeito de pixels cujo centro se encontra a uma distância superior a d mas ainda apresentam uma parte da sua área a uma distância inferior a d . Tendo poucos pixels nessa zona, o que acontece para valores de d baixos, não é correcto desprezar o efeito dos pixels da periferia. Para incluir esse efeito podemos definir um conjunto $A^p(x, y, d)$ *fuzzy* onde incluímos com um grau inferior a 1 esses pixels da periferia. Para definir esse conjunto temos a função $m_{A^p(x, y, d)}$ dada por:

$$m_{A^p(x, y, d)}(p_{xy}) = \begin{cases} 1 & \text{se } l_{xy}^p \leq d - 0.5 \\ 1 - l_{xy}^p & \text{se } d - 0.5 < l_{xy}^p < d + 0.5 \\ 0 & \text{se } l_{xy}^p \geq d + 0.5 \end{cases} \quad (5.51)$$

com

$$l_{xy}^p = l^p(c(p_{xy}), (x, y)) \quad (5.52)$$

Nesta definição estamos a considerar que a área do pixel dentro da zona que definimos pode ser aproximada pela distância do seu centro à fronteira. Mais uma vez trata-se de um ligeiro compromisso que nos permite uma implementação expedita.

Podemos agora considerar uma série de funções que extraem variadas características de um destes conjuntos. Por exemplo podemos ter a função Y_A que extrai a luminância média dos pixels pertencentes ao conjunto A dada por:

$$Y_A = \frac{1}{M(A)} \sum_x \sum_y m_A(p_{xy}) \times Y_{xy} \quad (5.53)$$

5.13.2 Versão 98

Nesta versão recorremos a três marcadores quadrangulares que podem ser brancos ou pretos. Estes três marcadores estão dispostos, segundo a recta equidistante dos centros dos marcadores primário e secundário de cada robot. Um dos marcadores é posicionado, exactamente na intersecção com a linha que une os marcadores primário e secundário. Os outros dois marcadores são posicionados à esquerda e à direita desse marcador central. Ficamos assim com uma sequência de três marcadores em que cada um pode ter a cor branca ou negra. A figura 5.16 ilustra o esquema de posicionamento destes marcadores adicionais ou de identificação.

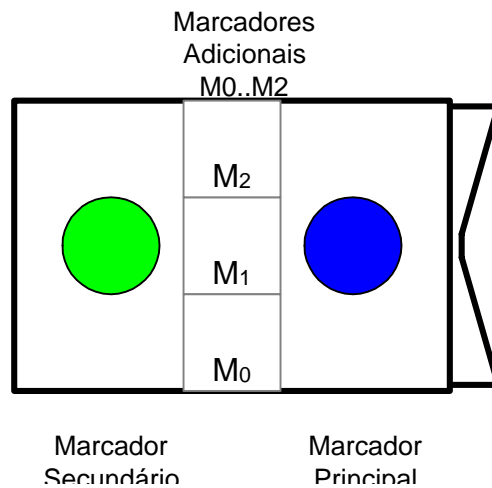


Figura 5.16: Esquema dos marcadores - Versão 98

Podemos associar a cada marcador um bit que pode ser considerado um

1 caso o marcador seja branco e 0 para um marcador negro. Teremos então para os três marcadores M_0 , M_1 e M_2 , respectivamente os bits b_0 , b_1 e b_3 . Com estes três bits podemos escrever 2^3 números diferentes. Considerando que nós numeramos os robots de 1 a 5, podemos usar esta sequência de marcadores para escrever o número do robot em binário. Por exemplo o robot número 4 disporá de marcadores adicionais em que o mais à esquerda será branco, o do meio preto e o da direita também preto. Esta configuração corresponde à sequência branco, preto, preto ou seja 100 binário, que em decimal corresponde ao número 4. Assim ao escrever em binário, recorrendo a marcadores adicionais, um número associado a cada robot podemos identificar a qual robot corresponde um certo par de marcadores principal e secundário. Podemos ver explicitada na figura 5.17 esta configuração para cada um dos robots.

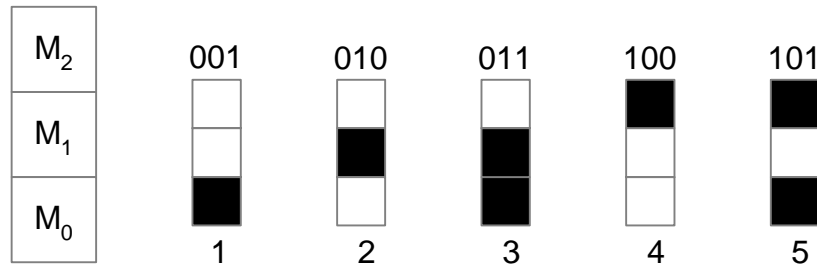


Figura 5.17: As configurações dos marcadores adicionais - Versão 98

Uma outra função destes marcadores adicionais é ajudar a validar um par marcador principal, marcador secundário. Há configurações em que é ambíguo qual é o emparelhamento correcto entre os marcadores principais e secundários.

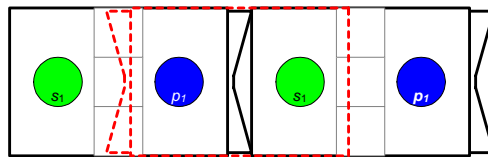


Figura 5.18: Ambiguidade no emparelhamento dos marcadores - Caso A

É certo que no caso A, ilustrado na figura 5.18, o emparelhamento errado poderia ser detectado pela presença de um par (p_1, s_2) demasiado distante para completar a identificação dos robots. Porém o algoritmo teria de ser capaz de desemparelhar marcadores e procurar um novo emparelhamento mais adequado. Pior, a presença de observações espúrias que pudessem ser incorrectamente consideradas marcadores poderia evitar a identificaç

ão deste caso. Outro problema advém do facto desta técnica depender do conhecimento do número de robots presentes.

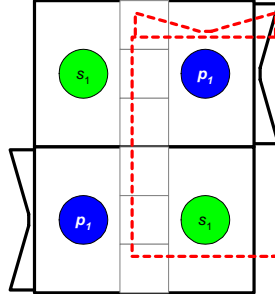


Figura 5.19: Ambiguidade no emparelhamento dos marcadores - Caso B

Finalmente podemos apreciar o caso B, na figura 5.19, onde é impossível, observando apenas a disposição dos marcadores decidir qual a configuração correcta. Sem o recurso aos marcadores adicionais, as técnicas que poderiam ser empregues para distinguir a configuração correcta são pouco satisfatórias:

- *Distinção entre o campo e o topo do robot.* Recorrendo a essa distinção haveria a possibilidade de distinguir a verdadeira atitude de cada robot. Infelizmente é muito difícil distinguir o preto do topo do robot do verde escuro, à sombra, que é a côr do campo. Tanto quanto soubemos nenhuma equipa tinha conseguido realizar esta operação com sucesso.
- *Recurso a informação passada.* Conhecendo a atitude dos robots na imagem anterior seria possível decidir com uma boa certeza a configuração a escolher. Para além do óbvio problema para o instante inicial em que não há informação anterior, temos a inadequação de misturar informação de níveis conceptuais diferentes. Mais ainda, estaríamos a perder a independência temporal que todo o sistema tem mostrado possuir. No entanto, esta é a operação que nos vemos obrigados a efectuar no caso da afectação dos marcadores aos robots adversários.

Felizmente a presença dos marcadores adicionais é suficiente para resolver este problema de uma forma elegante e robusta. Uma das vantagens de escolhermos os números de 1 a 5 para a representação binária é que, para além da correspondência número-número do robot, em todas estas sequências podemos encontrar pelo menos um 0 ou um 1. Assim, para cada robot, teremos sempre pelo menos um marcador branco e um marcador preto.

Dado um par marcador principal, marcador secundário podemos calcular as localizações previsíveis para os marcadores adicionais.

Se definirmos três pontos: c_1 , c_2 e c_3 que correspondem aos centros calculados dos três marcadores adicionais, podemos construir os conjuntos

A_1 , A_2 e A_3 onde colecionamos os pixels que se encontram a uma distância inferior a d_{mc} . Podemos obter então Y_{A_i} como uma medida da luminância média das zonas onde esperamos encontrar os marcadores adicionais. Como esperamos a presença de pelo menos um marcador branco e outro preto teremos:

$$\max(Y_{A_i}) - \min(Y_{A_i}) > s_{bw} \quad (5.54)$$

Ou seja a separação entre a luminância média da zona mais escura e a a luminância média da zona mais clara deverá ser superior a um nível s_{bw} . E também tendo:

$$j = \arg \max_i Y_{A_i} \quad (5.55)$$

$$k = \arg \min_i Y_{A_i} \quad (5.56)$$

devemos verificar que para:

$$z = Y_{A_i} : i \neq j \wedge i \neq k \quad (5.57)$$

$$\max(Y_{A_i}) - z < s_{ec} \quad (5.58)$$

ou

$$z - \min(Y_{A_i}) < s_{ec} \quad (5.59)$$

Ou seja a zona que não tem a luminância máxima nem mínima deverá apresentar uma luminância ainda assim próxima da máxima observada, caso se trate de outro marcador branco, ou então próxima de mínima caso se trate de um marcador preto.

Estas condições servem de filtro para validar a presença de uma sequência de marcadores adicionais no ponto sugerido pelo par marcador principal, marcador auxiliar. Deste modo, uma situação ambígua pode ser resolvida na maioria das vezes.

Na figura 5.20 podemos ver a equipa de 98, durante um treino em Paris, onde é evidente a presença dos marcadores adicionais.

5.13.3 Versão 99

Optamos sempre por colocar o marcador principal, isto é a bola oficial, sobre o eixo de rotação do robot. Esta opção permite extrair directamente desse marcador a posição do centro de coordenadas de cada robot sem o recurso a marcadores adicionais. Durante o projecto dos robots usados em 99 colocamos o centro de rotação mais próximo do centro do robot do que tínhamos em 98. Essa opção inviabilizou a configuração de 98 para os marcadores adicionais. Dado o tamanho mínimo para um marcador, verificamos que não sobrava espaço suficiente entre o marcador principal e o secundário para uma



Figura 5.20: A equipa de 98 durante um treino em Paris

fila de marcadores adicionais, tal como tínhamos feito para os robots de 98. Para contornar esse problema, e dado que agora dispunhamos de um espaço extra entre o marcador principal e a borda do robot, implementámos uma configuração diferente para os marcadores adicionais. Desta vez recorremos a quatro marcadores que dispusémos à volta do marcador principal. Esta configuração está ilustrada na figura 5.21.

Nesta configuração usamos quatro marcadores adicionais, o que possibilita 2^4 combinações possíveis. Naturalmente usamos apenas cinco que foram escolhidas de modo a garantir a presença de, pelo menos, um marcador branco e um preto. Dado que o espaço de escolha é mais vasto, seleccionamos ainda as cinco combinações de modo a que, de uma configuração para qualquer outra válida houvesse a mudança de mais do que um marcador. Desta forma, um erro que avalie um marcador incorrectamente dará origem a um código inválido e não à confusão do robot representado. Uma outra restrição que impusemos à escolha das sequências foi a de usar somente combinações onde estão presentes dois marcadores brancos e dois marcadores pretos.

Outra vantagem desta configuração é que não corremos o risco de procurar os marcadores numa zona externa ao robot. Na versão de 98 havia a possibilidade, ainda que remota, de após a escolha inadequada de um par marcador principal, marcador secundário, fazermos a procura da sequência de marcadores adicionais sobre uma linha de marcação do campo. Aí, seria possível ler inadvertidamente um código válido e aceitar o par erra-

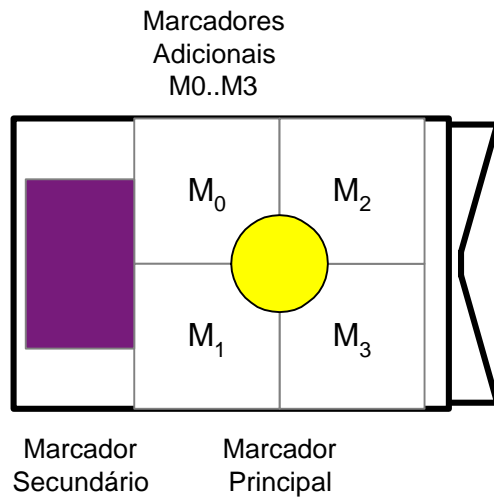


Figura 5.21: Esquema dos marcadores - Versão 99

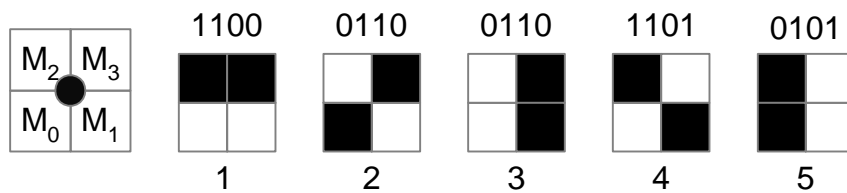


Figura 5.22: As configurações dos marcadores adicionais - Versão 99

damente. É certo que esta ocorrência era rara e rapidamente seria corrigida após uma leve movimentação dos robots, mas ainda assim levaria a um momentâneo descontrolo dos robots envolvidos. Na actual configuração, essa possibilidade está completamente eliminada pois desde que o marcador principal esteja correctamente identificado a pesquisa dos marcadores secundários exercer-se-á sempre no topo de um robot. Na figura 5.22 temos as configurações escolhidas para os cinco robots.

Capítulo 6

Filtragem da Medida do Estado da Bola

Neste capítulo apresentamos o sistema de filtragem das medidas da localização da bola. Para tal desenvolvemos um modelo que se mostrou válido para descrever a maior parte do movimento da bola. Graças à linearidade deste modelo, a aplicação de um filtro de Kalman é uma tarefa que não apresenta problemas de maior. Esta abordagem permite ainda obter uma previsão do novo estado da bola, o que permite facilitar o cálculo do controlo e disponibiliza também uma estimativa das componentes da velocidade. Estas duas funcionalidades são preciosas para o desempenho dos sistemas de decisão e controlo.

6.1 Dinâmica da bola

Um possível modelo para a dinâmica da bola pode ser dado pelas equações:

$$\dot{x}_b = v_{bx} \quad (6.1)$$

$$\dot{y}_b = v_{by} \quad (6.2)$$

$$\dot{v}_{bx} = -\lambda_b v_{bx} \quad (6.3)$$

$$\dot{v}_{by} = -\lambda_b v_{by} \quad (6.4)$$

Este é um modelo muito simplificado. Assume que a bola é como um disco rígido que desliza numa superfície com atrito viscoso. O coeficiente de atrito é λ_b e é independente da direcção do movimento da bola. Uma simplificação extra pode ser conseguida considerando que não existe atrito e logo temos o coeficiente de atrito $\lambda_b = 0$. Essa simplificação é muito aceitável pois os valores típicos de λ_b , para uma bola de golfe sobre uma mesa de Ténis de Mesa, são relativamente baixos.

Segundo este modelo a bola nunca muda de direcção. Estamos a considerar que, quando a bola choca com um robot ou com uma parede, há uma

perturbação que invalida momentaneamente o modelo. Deixar essa incerteza por tratar parece, à primeira vista, uma opção pouco interessante. Poderíamos refinar o modelo introduzindo o conceito de forças actuadas sobre a bola para modelizar o comportamento da mesma durante os choques:

$$\dot{x}_b = v_{bx} \quad (6.5)$$

$$\dot{y}_b = v_{by} \quad (6.6)$$

$$\dot{v}_{bx} = -\lambda_b v_{bx} + u_{bx} \quad (6.7)$$

$$\dot{v}_{by} = -\lambda_b v_{by} + u_{by} \quad (6.8)$$

Infelizmente, determinar u_{bx} e u_{by} não é uma tarefa simples. Para casos bem isolados, tais como choque com uma parede ou contra a superfície lisa de um robot, não é um problema excessivamente complexo. Porém, caso o robot apresente uma forma mais complexa ou aconteça a situação, relativamente frequente, de termos vários robots em contacto com a bola, torna-se praticamente impossível estabelecer, com um mínimo de precisão, a sequência e intensidade dos choques e conseqüentemente determinar as forças que actuam sobre a bola.

Pareceu-nos mais realista introduzir um modelo que:

- Consiga aproveitar a informação de que o modelo simplificado não é momentaneamente válido (porque houve um choque).
- Não obrigue ao cálculo das forças activas sobre a bola.

Dado que o modelo que queremos usar servirá essencialmente para a evolução do estado nas equações do Filtro de Kalman, faz todo o sentido construir uma versão estocástica do modelo:

$$\dot{x}_b = v_{bx} + w_{bpx} \quad (6.9)$$

$$\dot{y}_b = v_{by} + w_{bpy} \quad (6.10)$$

$$\dot{v}_{bx} = -\lambda_b v_{bx} + w_{bv_x} \quad (6.11)$$

$$\dot{v}_{by} = -\lambda_b v_{by} + w_{bv_y} \quad (6.12)$$

Em que $w_{bp[x,y]}$ e $w_{bv[x,y]}$ são processos estocásticos que servirão para caracterizar a incerteza associada à evolução da dinâmica. por uma questão de simplicidade, e na falta de melhor informação, consideraremos que a sua distribuição é gaussiana e de média nula. A $w_{bp[x,y]}$ podemos associar uma covariância muito baixa. Que a posição é o integral da velocidade é algo assegurado fisicamente e nem sequer há a incerteza associada a parâmetros extra. A sua existência serve apenas para lidar com as pequenas imprecisões associadas a uma série de factores:

- O cálculo discreto, que acontece sempre que a informação é processada digitalmente, não tem uma precisão infinita.

- A implementação, normalmente em tempo discreto, pode não ser perfeitamente equivalente à formulação em tempo contínuo.

A assunção da existência dessa incerteza apresenta ainda outra vantagem, no contexto do Filtro de Kalman, pois permite "absorver" com mais facilidade erros na componentes da velocidade.

Já $w_{bp[x,y]}$ deverá apresentar uma covariância que muda ao longo do tempo. Mais concretamente:

- Durante os instantes em que a bola se desloca livremente, isto é, longe de robots e de paredes a variância de $w_{bp[x,y]}$ pode tomar um valor baixo. isso equivale a dizer que estamos praticamente no caso do modelo apresentado em 6.1
- Quando está iminente e durante os instantes de um choque podemos indicar que o modelo não é perfeitamente válido assumindo um valor muito superior para a variância de $w_{bp[x,y]}$. Durante esses instantes estamos a informar o filtro de Kalman de que deverá ser dada maior confiança às observações e não à dinâmica do modelo.

Concretamente teremos então:

$$d_r^{\min} = \min_i (l^p ((xb, yb), (x_{r_i}, y_{r_i}))) \quad (6.13)$$

$$d_a^{\min} = \min_i (l^p ((xb, yb), (x_{a_i}, y_{a_i}))) \quad (6.14)$$

$$d^{\min} = \min (d_r^{\min}, d_a^{\min}) \quad (6.15)$$

que será o valor mínimo da distância entre qualquer robot e a bola. Definiremos então:

$$var(w_{bp}) = \alpha \quad (6.16)$$

$$var(w_{bv}) = \psi \frac{d^{\min} + \beta}{d^{\min} + \gamma} \quad (6.17)$$

Deste modo, podemos indicar uma covariância que varia de acordo com a distância de um robot à bola.

Idealmente, o campo onde se processa o jogo deverá ser perfeitamente plano e horizontal. Na prática haverá sempre algum desvio em relação a essa situação. Esses desvios podem ser incorporado no modelo. Normalmente, estes efeitos não são significativos. Assim, o esforço e complexidade extra não se traduzirão por um desempenho do modelo substancialmente melhor. É porém conveniente dispor da possibilidade de introduzir estes efeitos no

nosso sistema para podermos lidar com uma possível situação em que estes fenómenos não sejam completamente desprezáveis. Nessa medida podemos introduzir o seguinte modelo:

$$\dot{x}_b = v_{bx} + w_{bpx} \quad (6.18)$$

$$\dot{y}_b = v_{by} + w_{bpy} \quad (6.19)$$

$$\dot{v}_{bx} = -\lambda_b v_{bx} - \nabla_x h(x, y) + w_{bv_x} \quad (6.20)$$

$$\dot{v}_{by} = -\lambda_b v_{by} - \nabla_y h(x, y) + w_{bv_y} \quad (6.21)$$

Aqui, $h(x, y)$ indica o desvio da altura do campo em relação à horizontal, para as coordenadas x, y . O facto de explicitarmos o gradiente dessa função evidencia o facto de estarmos a trabalhar com um campo de potencial e garante, no modelo, que $\nabla \times h(x, y) = 0$.

Até agora temos considerado a bola como um disco sem atrito lateral. Poderíamos refinar o modelo considerando que esse atrito não é nulo. Aí, teríamos de expandir o estado para incluir a velocidade de rotação do disco (bola) sobre o seu eixo vertical:

$$\dot{x}_b = v_{bx} + w_{bpx} \quad (6.22)$$

$$\dot{y}_b = v_{by} + w_{bpy} \quad (6.23)$$

$$\dot{v}_{bx} = -\lambda_b v_{bx} - \nabla_x h(x, y) + k_b w_b v_{by} + w_{bv_x} \quad (6.24)$$

$$\dot{v}_{by} = -\lambda_b v_{by} - \nabla_y h(x, y) - k_b w_b v_{bx} + w_{bv_y} \quad (6.25)$$

$$\dot{w}_b = f_b(w_b, v_{bx}, v_{by}) \quad (6.26)$$

Onde os termos $k_b w_b v_{by}$ e $k_b w_b v_{bx}$ representam o contributo do efeito de rotação sobre a aceleração normal à trajectória. Em termos futebolísticos, "o efeito" associado à trajectória da bola. Já $f_b(w_b, v_{bx}, v_{by})$ ilustrará a dinâmica do próprio movimento de rotação. Numa primeira aproximação poderíamos fazer

$$f_b(w_b, v_{bx}, v_{by}) = -\gamma_b w_b \quad (6.27)$$

considerando apenas atrito viscoso.

Mesmo assim, as constantes γ_b e k_b não são muito fáceis de estimar pois:

- Não é possível medir directamente w_b . Pelo menos, com o recurso exclusivo à imagem da bola.
- Os termos $k_b w_b v_{by}$ e $k_b w_b v_{bx}$ têm uma contribuição pouco importante sobre a dinâmica da bola e portanto é difícil obter uma estimativa precisa dessa contribuição

Devido a estes problemas e à pouca influência que estes factores têm sobre a dinâmica da bola, considerámos que este modelo introduzia uma complexidade desnecessária que não levaria a uma melhoria significativa na qualidade dos resultados. Abstivemo-nos ainda de o refinar ainda mais, por exemplo considerando a bola não como um disco mas sim como uma esfera com o conseqüente aumento de variáveis de estado. Esse acréscimo era necessário pois o eixo de rotação passaria a poder ter qualquer orientação no espaço.

6.2 Identificação dos Parâmetros do Modelo

Para usar o modelo representado pelas equações em 6.18 será necessário explicitar a forma que $h(x, y)$ apresentará e estimar os parâmetros associados. Apresentamos duas possíveis formulações, muito simples:

$$h(x, y) = \alpha x + \beta y \quad (6.28)$$

e

$$h(x, y) = \alpha x + \beta y + \gamma x^2 + \lambda y^2 + \mu xy \quad (6.29)$$

No primeiro caso assumimos que o campo é perfeitamente plano e apenas está ligeiramente inclinado numa dada direcção. Temos apenas dois parâmetros para estimar e o sistema mantém-se linear.

No segundo caso já é possível modelizar uma forma curva para o campo embora essa forma não seja mais complexa dos que um parabolóide. Esse parabolóide apresenta o seu eixo paralelo ao eixo z e pode estar rodado em relação aos eixos x e y . O seu vértice pode ser deslocado da origem para outro ponto. Felizmente este modelo é ainda linear já que estaremos a trabalhar com o $\nabla h(x, y)$ e nesse caso teremos:

$$\nabla_x h(x, y) = \alpha + 2\gamma x + \mu y \quad (6.30)$$

$$\nabla_y h(x, y) = \beta + 2\lambda y + \mu x \quad (6.31)$$

Podemos definir um vector θ que coleciona os parâmetros do modelo, isto é:

$$\theta = (\alpha, \beta, \gamma, \lambda, \mu) \quad (6.32)$$

Infelizmente, mesmo para esta definição de $h(x, y)$, não conseguimos representar o caso em que o campo se apresenta como uma sucessão de montes e vales extremamente suaves. Para tal teríamos de escolher para $h(x, y)$ uma função mais complexa ou então escolher uma partição do campo e tabelar o seu valor para cada uma das áreas dessa partição. Naturalmente qualquer um dos casos dá origem a uma quantidade considerável de parâmetros a estimar. O principal problema dessa estimação é o facto de trabalhar com

”sinais” muito pouco intensos e portanto com uma relação sinal/ruído muito desfavorável.

Apesar de tudo procurámos estimar os parâmetros do modelo pois podem ser usados para avaliar o funcionamento do mecanismo de estimação: não havendo uma inclinação significativa um valor muito diferente de zero nesses parâmetros sinalizará um funcionamento menos bom do processo de estimação causado por algum problema no algoritmo ou por um problema na qualidade dos dados empregues.

Para além dos parâmetros presentes em θ o outro parâmetro a estimar é λ_b . Eventualmente poderíamos também tentar caracterizar a covariância de $w_{bp[x,y]}$ e $w_{bv[x,y]}$ mas como vimos, haverá uma série de condições, principalmente quando os choques, em que queremos considerar valores diferentes para essas grandezas.

6.3 Filtro de Kalman

Como dispomos de um modelo linear podemos recorrer a um filtro de Kalman perfeitamente clássico, tal como está descrito nas equações 4.30-4.37. Neste caso o modelo da bola em tempo contínuo descrito na equação 6.18, com a função que representa a inclinação da equação 6.29, poderá ser escrito, evidenciando a dependência temporal das variáveis de estado:

$$\dot{x}_b(t) = v_{bx}(t) + w_{bpx}(t) \quad (6.33)$$

$$\dot{y}_b(t) = v_{by}(t) + w_{bpy}(t) \quad (6.34)$$

$$\dot{v}_{bx}(t) = -\lambda_b v_{bx}(t) - (\alpha + 2\gamma x(t) + \mu y(t)) + w_{bv_x}(t) \quad (6.35)$$

$$\dot{v}_{by}(t) = -\lambda_b v_{by}(t) - (\beta + 2\lambda y(t) + \mu x(t)) + w_{bv_y}(t) \quad (6.36)$$

em notação matricial fica:

$$\dot{x}(t) = Ax(t) + Bu(t) + w(t) \quad (6.37)$$

com

$$x(t) = \begin{bmatrix} x_b(t) \\ y_b(t) \\ v_{bx}(t) \\ v_{by}(t) \end{bmatrix}, \quad w(t) = \begin{bmatrix} w_{bpx}(t) \\ w_{bpy}(t) \\ w_{bv_x}(t) \\ w_{bv_y}(t) \end{bmatrix}, \quad u(t) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (6.38)$$

e

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -2\gamma & -\mu & -\lambda_b & 0 \\ -2\lambda & -\mu & 0 & -\lambda_b \end{bmatrix} \quad (6.39)$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\alpha & 0 \\ 0 & -\beta \end{bmatrix} \quad (6.40)$$

Consideramos $u(t)$ constante pois a sua função é permitir a presença dos termos α e β na dinâmica do sistema.

É necessário ainda indicar o modelo das observações. Neste caso, como as variáveis observadas são x e y temos:

$$y(t) = Hx(t) + v(t) \quad (6.41)$$

onde

$$y(t) = \begin{bmatrix} x_b(t) \\ y_b(t) \end{bmatrix}, \quad v(t) = \begin{bmatrix} v_{bx}(t) \\ v_{by}(t) \end{bmatrix} \quad (6.42)$$

e

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (6.43)$$

Como podemos constatar, o filtro de Kalman possibilitará, não só a filtragem da posição como funcionará como um estimador de estado que nos fornece a velocidade segundo x e y . Este segundo papel é extremamente interessante pois permite incorporar essa informação no mecanismo de decisão. Por exemplo, o robot que serve de guarda redes pode tentar posicionar-se no ponto onde poderá intersectar a bola incorporando assim um elemento preditivo no sistema global de controlo.

Outra função do filtro de Kalman é fornecer, no instante k , uma estimativa do estado no instante $k + 1$ para possibilitar o cálculo dos sinais de controlo, a enviar aos robots por rádio, que será aplicado no instante t_{k+1} .

Capítulo 7

Filtragem da Medida do Estado dos Robots

Neste capítulo abordamos o problema de filtrar as observações das posições dos robots de modo a construir uma estimativa do estado dos mesmos. Este problema é mais complexo do que o do capítulo anterior pois o estado do robot inclui mais variáveis e temos casos em que dispomos da observação de dois marcadores para um robot. Também é necessário recorrer a modelos e filtros diferentes caso se trate de um robot da nossa equipa ou da equipa adversária. Isto porque para os robots da nossa equipa temos necessidade de uma caracterização mais completa. Felizmente essa dificuldade é contrabalançada pela capacidade de que dispomos de recorrer a marcadores extra com uma geometria por nós conhecida.

7.1 Dinâmica do Robot

Para podermos empregar um filtro de Kalman é necessário um modelo, preferencialmente linear, para a dinâmica do sistema que queremos estudar. No nosso caso, dado que podemos separar o estado da equipa nos vários estados de cada robot, teremos de construir e posteriormente estimar um modelo para o movimento de cada robot. Como veremos, não será possível obter um modelo linear adequado. Porém, o modelo não linear que nos pareceu mais adequado pode ser linearizado com facilidade.

Recorrendo a um Filtro de Kalman Extendido podemos usar o modelo não linear. Naturalmente, esta abordagem não fornece as mesmas garantias teóricas. Como é frequentemente assinalado [10] um filtro de Kalman Extendido não fornece as mesmas garantias de convergência e de optimalidade. De qualquer modo, como se trata de uma não linearidade algo suave, o problema não se manifesta impeditivo de um desempenho dentro dos valores esperados de um Filtro de Kalman linear. Os resultados obtidos são perfeitamente satisfatórios e evidenciam as vantagens da filtragem empregue.

Há que fazer aqui a distinção entre os robots pertencentes à equipa que é por nós controlada, dos robots da equipa adversária:

□ Os robots da nossa equipa:

1. Têm uma configuração conhecida.
2. Conhecemos os modos de movimento.
3. Temos, garantidamente, informação sobre a sua atitude.
4. Conhecemos sinal de controlo que lhes está a ser aplicado.

□ Os robots da equipa adversária:

1. Não conhecemos, à partida, as suas restrições no modo de movimentação. Por exemplo é perfeitamente possível que se mostrem capazes de uma movimentação omnidireccional, tornando a informação acerca da sua atitude desnecessária. Outros parâmetros, extremamente importantes, que também são desconhecidos à partida é a sua aceleração e velocidade máxima.
2. O esquema de marcadores implementado nos robots pode não dar informação sobre a sua atitude. Isso pode acontecer, quer porque a configuração dos marcadores não se adequa à nossa capacidade de extracção de informação, quer porque a maneira como essa informação foi codificada não está prevista pelo nosso *software*.
3. Talvez a diferença mais marcante derive do facto de não dispormos de informação sobre os sinais de controlo aplicados nos robots. A falta dessa informação é, per si, suficiente para justificar o recurso a um modelo diferente para caracterizar a sua dinâmica.

Consequentemente, usaremos um modelo diferente para cada um destes grupos de robots.

7.1.1 Robots Controlados

Consideremos o i -ésimo robot, podemos descrever a sua dinâmica recorrendo ao seguinte modelo:

$$\dot{x}_r(t) = v_{rx}(t) \quad (7.1)$$

$$\dot{y}_r(t) = v_{ry}(t) \quad (7.2)$$

$$\dot{\theta}_r(t) = \omega_r(t) \quad (7.3)$$

$$\dot{v}_{rt}(t) = \frac{1}{m} (-\lambda_t v_{rt}(t) + K_v (u_{rd}(t) + u_{re}(t))) \quad (7.4)$$

$$\dot{v}_{rn}(t) = \frac{1}{m} (-\lambda_n v_{rn}(t)) \quad (7.5)$$

$$\dot{\omega}_r(t) = \frac{1}{I} (-\lambda_\omega \omega_r(t) + K_\omega (u_{rd}(t) - u_{re}(t))) \quad (7.6)$$

com

$$v_{rx} = v_{rt} \cos \theta_r - v_{rn} \sin \theta_r \quad (7.7)$$

$$v_{ry} = v_{rt} \sin \theta_r + v_{rn} \cos \theta_r \quad (7.8)$$

Por uma questão de simplicidade na notação não acrescentamos o índice i nas variáveis.

Este modelo assume o robot como um corpo rígido, assente no plano xy . Consideramos que está sujeito a forças de atrito $F_t(t) = \lambda_t v_{rt}(t)$ e $F_n(t) = \lambda_n v_{rn}(t)$, segundo as componentes tangencial e normal. Temos também $T_a(t) = \lambda_\omega \omega_r(t)$ que é um binário de atrito. As forças externas actuam no ponto de contacto das duas rodas com o chão e dão origem a uma componente tangencial $F_t = K_v(u_{rd}(t) + u_{re}(t))$ e um binário $T_\omega = K_\omega(u_{rd}(t) - u_{re}(t))$. Estas forças serão disponibilizadas pelos motores presentes nos robots. A figura 7.1 ilustra o sistema de coordenadas em questão.

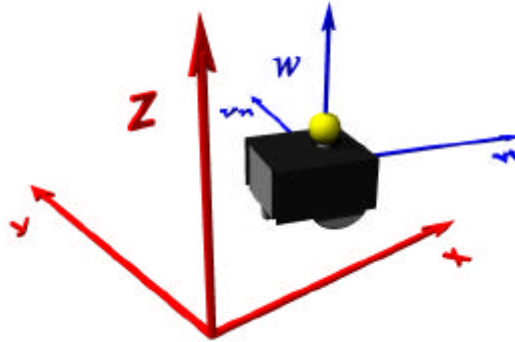


Figura 7.1: Um Robot e respectivas coordenadas

Por outro lado, não podemos esquecer a presença de um ciclo de controlo local. Tipicamente, esse controlador receberá uma referência sob a forma de uma trajectória e actuará sobre os motores para que o robot siga essa trajectória. Ou seja, a partir de $v_{ref}(t)$ e $\omega_{ref}(t)$ disponibilizará $u_{re}(t)$ e $u_{rd}(t)$.

Podemos assumir que o controlador local conseguirá, a maior parte das vezes, realizar com sucesso a tarefa de impor a velocidade linear e angular desejadas. É certo que quando se verificarem variações bruscas nos sinais de entrada o controlador não conseguirá fazer o robot seguir imeditamente essa referência. Adoptamos assim um modelo em que incluímos o efeito do

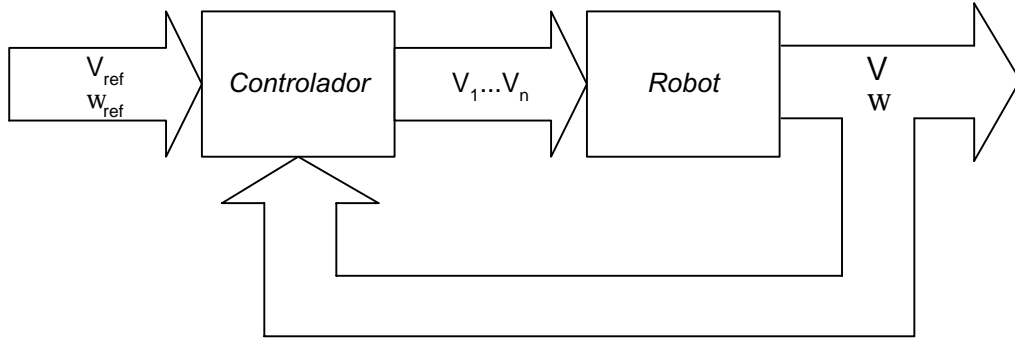


Figura 7.2: O Controlador Local ao Robot

controlador local e modelizamos a sua resposta por um sistema de primeira ordem:

$$\dot{x}_r(t) = v_{rt}(t) \cos \theta_r - v_{rn}(t) \sin \theta_r \quad (7.9)$$

$$\dot{y}_r(t) = v_{rt}(t) \sin \theta_r + v_{rn}(t) \cos \theta_r \quad (7.10)$$

$$\dot{\theta}_r(t) = \omega_r(t) \quad (7.11)$$

$$\dot{v}_{rt}(t) = \alpha (v_{ref} - v_{rt}(t)) \quad (7.12)$$

$$\dot{v}_{rn}(t) = -\beta v_{rn}(t) \quad (7.13)$$

$$\dot{\omega}_r(t) = \gamma (\omega_{ref} - \omega_r(t)) \quad (7.14)$$

Estas são as equações que modelizam a dinâmica, falta ainda explicitar as equações que governam as observações:

$$y(t) = h(x(t)) \quad (7.15)$$

como

$$y(t) = \begin{bmatrix} x_r \\ y_r \\ \theta_r \end{bmatrix} \quad (7.16)$$

fica:

$$h_1(x(t)) = x_r(t) \quad (7.17)$$

$$h_2(x(t)) = y_r(t) \quad (7.18)$$

$$h_3(x(t)) = \theta_r(t) \quad (7.19)$$

Este modelo apresenta apenas três parâmetros para estimar α , β e γ , embora apresente as dificuldades inerentes ao facto de se tratar de um modelo não linear.

É necessário calcular as linearizações da função associada à dinâmica:

$$F = \frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 0 & -v_{rl} \sin \theta_r - v_{rt} \cos \theta_r & \cos \theta_r & -\sin \theta_r & 0 \\ 0 & 0 & v_{rl} \cos \theta_r - v_{rt} \sin \theta_r & \sin \theta_r & \cos \theta_r & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\alpha & 0 & 0 \\ 0 & 0 & 0 & 0 & -\beta & 0 \\ 0 & 0 & 0 & 0 & 0 & -\gamma \end{bmatrix} \quad (7.20)$$

Como $h(x)$ é linear fica simplesmente:

$$\frac{\partial h}{\partial x} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (7.21)$$

Esta aproximação, baseia-se numa expansão em série de Taylor[1] das funções f e h . Normalmente, essa expansão tem um intervalo de convergência que pode ser mais ou menos apertado. Verificamos que a qualidade da aproximação diminuía significativamente para valores de θ crescentes. Assim, vimo-nos obrigados a recorrer a uma aproximação onde consideramos θ constante entre cada instante de amostragem. O que dá origem a:

$$F = \begin{bmatrix} 0 & 0 & 0 & \cos \theta_r & -\sin \theta_r & 0 \\ 0 & 0 & 0 & \sin \theta_r & \cos \theta_r & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\alpha & 0 & 0 \\ 0 & 0 & 0 & 0 & -\beta & 0 \\ 0 & 0 & 0 & 0 & 0 & -\gamma \end{bmatrix} \quad (7.22)$$

Esta equação difere da equação 7.20 apenas nos elementos da terceira coluna, primeira e segunda linha.

Com esta linearização podemos aplicar as equações 4.43 - 4.53 para aplicarmos um Filtro de Kalman Estendido para cada um dos robots da nossa equipa.

7.1.2 Robots Adversários

Como foi realçado, para os robots da equipa adversária não é possível nem desejável construir um modelo tão detalhado. O modelo que escolhemos é relativamente semelhante, em estrutura, ao da bola. Para o i -ésimo robot adversário teremos:

$$\dot{x}_{a_i}(t) = v_{a_{ix}}(t) \quad (7.23)$$

$$\dot{y}_{a_i}(t) = v_{a_{iy}}(t) \quad (7.24)$$

$$\dot{v}_{a_{ix}}(t) = -\lambda_a v_{a_{ix}}(t) \quad (7.25)$$

$$\dot{v}_{a_{iy}}(t) = -\lambda_a v_{a_{iy}}(t) \quad (7.26)$$

Ao contrário do que acontece no caso da bola, teremos λ_a a tomar um valor significativo, indicando a tendência normal de um robot para parar rapidamente. Podemos escrever o correspondente modelo estocástico:

$$\dot{x}_{a_i}(t) = v_{a_ix}(t) + w_{a_ipx}(t) \quad (7.27)$$

$$\dot{y}_{a_i}(t) = v_{a_iy}(t) + w_{a_ipty}(t) \quad (7.28)$$

$$\dot{v}_{a_ix}(t) = -\lambda_a v_{a_ix}(t) + w_{a_ivy}(t) \quad (7.29)$$

$$\dot{v}_{a_iy}(t) = -\lambda_a v_{a_iy}(t) + w_{a_ivy}(t) \quad (7.30)$$

Para caracterizar este modelo precisaremos de indicar as covariâncias de w_{a_ipx} , w_{a_ipty} , w_{a_ivy} e w_{a_ivy} . Por uma questão de simplicidade e para evitar fenómenos de ajuste dos parâmetros a amostar e não ao processo, consideramos:

$$w_{a_ipx} = w_{a_ipty} \quad \forall i \in \{1, 2, 3, 4, 5\} \quad (7.31)$$

e

$$w_{a_ivy} = w_{a_ivy} \quad \forall i \in \{1, 2, 3, 4, 5\} \quad (7.32)$$

O processo de calibração foi conseguido através da estimação destes valores a partir de um filme onde a equipa adversária evolui no campo. Naturalmente que para diferentes equipas é desejável recalibrar estes parâmetros.

Como é evidente este filtro é extremamente semelhante ao aplicado à bola. A dificuldade extra surge aqui devido ao facto de poder haver até cinco robots adversários presentes. Enquanto que, como vimos, para extrair uma observação da bola bastava escolher o conjunto de pixels mais promissor, aqui, existe a dificuldade extra da associar correctamente os conjuntos aos respectivos robots. Isto acontece, porque estes robots não dispõem do mecanismo de marcadores adicionais (pelo menos não é garantido).

Capítulo 8

Implementação e Resultados

Neste capítulo mostramos as opções tomadas para possibilitar a implementação eficiente dos algoritmos apresentados. Será particularmente realçado os aspectos mais delicados que permitiram o desempenho obtido. Em certa medida este assunto já foi considerado anteriormente, pois a definição de todo o sistema teve de ter em conta as restrições em termos de poder computacional e a inerente característica de tempo real do problema. Iremos, no entanto, considerar mais atentamente essas questões.

Apresentamos também os resultados obtidos para diferentes configurações.

8.1 Optimizar?

Provavelmente algumas das optimizações que aqui mostramos parecerão, dentro de algum tempo, datadas. A evolução no poder de computação e na memória disponível por vezes faz certas optimizações parecerem inúteis. Por vezes até ineficientes. Por exemplo, actualmente há processadores em que a operação com vírgula flutuante pode ser mais eficiente do que com certos formatos inteiros. Certas optimizações de velocidade que tornam o código significativamente maior podem revelar-se menos eficientes devido ao facto de o código já não caber completamente na *cache* do processador. O atraso extra introduzido enquanto o processador espera por código que deve ser lido da memória por não caber na *cache*, pode ser mais grave do que a execução do código não "optimizado" segundo essa técnica. Porém, é preciso não esquecer que existe um espectro de dispositivos com processadores muito extenso. Desde os processadores mais potentes a aproximarem-se de 1 *GHz* até aos microcontroladores que trabalham a uma frequência de poucos *MHz*, e por vezes menos ainda, há sempre lugar para optimizações. Provavelmente, no futuro, dispor de computadores significativamente mais rápidos, poderemos usar o poder de computação extra para aumentar a resolução da imagem ou realizar algum processamento adicional para refinar

a informação extraída. Por outro lado, poderá ser viável incorporar estes algoritmos num microcontrolador e facilitar o processamento distribuído por múltiplas câmaras.

8.1.1 Profundidade de cor

O termo profundidade de imagem é usado para designar o número de bits associados a cada componente de cor. Como já mencionamos anteriormente a opção mais eficiente passa por escolher $n_r = n_g = n_b = 5$. Embora visualmente não dê origem a uma reprodução de cor perfeita possibilita uma representação suficientemente precisa no que toca à necessidade de classificar as diferentes cores. A grande vantagem vem do facto que qualquer função que mapeia uma cor num vector poder ser representada por uma tabela com 2^{15} entradas. Como iremos ver, usaremos esse facto para conseguir uma implementação eficiente dos nossos algoritmos.

Inicialmente, trabalhamos com uma profundidade que usava os 16 bits codificando a componente verde em 6 bits em vez de 5. Apesar de possibilitar uma reprodução de cor ligeiramente melhor (ver imagem) as desvantagens não justificam esta opção:

- Uma função que mapeie o espaço de cores é representada por uma tabela com o dobro do tamanho, isto é 2^{16} entradas.
- O processamento da componente verde tem de ser diferente o que leva a um código menos regular e mais deselegante.
- A zona do espaço *rgb* abrangida por um dado valor, para cada uma das três componentes, deixa de ser um cubo para passar a ser um paralelepípedo pois terá o dobro do tamanho segundo a componente verde.

Assim, e como o dispositivo da aquisição permite receber a imagem directamente a 15 bits, escolhemos esse formato para a representação das imagens capturadas.

8.1.2 Implementação do algoritmo de Classificação

A descrição do algoritmo de Classificação feita no capítulo 5 tomou uma dimensão mais conceptual. Naturalmente, existem muitas maneiras de implementar os algoritmos tal como eles foram descritos. Sendo crucial conseguir completar o processamento no tempo disponível entre a chegada de duas imagens¹ é natural que tenha havido um cuidado especial na sua implementação.

¹A versão do software que foi usada em Paris, em 98, processava 25 imagens por segundo. Conseguíamos assim dispor de 40 *ms* entre duas imagens consecutivas. Já em Estocolmo, em 99, graças a um PC mais rápido foi possível fazer o processamento a 50 imagens por segundo. Ou seja em menos de 20 *ms*.

Um dos compromissos necessários foi o de discretizar os graus de pertença de um elemento do conjunto F_i em 256 níveis. Desta forma conseguimos representar esse grau de pertença recorrendo apenas a um byte. Tendo o espaço rgb discretizado com 5 bits por componente conseguimos representar a função m_{F_i} recorrendo a uma tabela com 2^{15} entradas. Cada entrada gasta apenas um byte e temos assim m_{F_i} representada por uma tabela que ocupa apenas 32768 bytes. Naturalmente avaliar $m_{F_i}(r, g, b)$ será uma operação imediata caso o vector (r, g, b) esteja representado com as três componentes empacotadas nos 15 bits menos significativos de uma *word*².

Como foi descrito, é necessário:

1. Classificar cada pixel pertencente à zona activa da imagem indicando o seu grau de pertença a cada uma dos conjuntos G_i .
2. Seguidamente, cada conjunto G_i será expurgado dos pixeis com um grau de pertença inferior a um dado limiar.
3. A operação de agregação de pixeis em centros é feita escolhendo pixeis com graus de pertença decrescentes.

Para que esta última operação possa ser realizada eficientemente é importante dispor de uma representação do conjunto G_i que possibilite o acesso aos pixeis segundo uma ordenação indexada pelo seu grau de pertença.

²O termo *word* é aqui usado para denotar uma palavra de 16 bits.

8.2 Resultados

Os resultados que apresentamos procuram ilustrar o nível de desempenho que o sistema proposto consegue alcançar. As condições segundo as quais estes resultados foram obtidos estão longe das ideais. Tipicamente, durante um desafio oficial do roboCup será de esperar condições mais favoráveis. Dado que os últimos desenvolvimentos do nosso trabalho apontam para o recurso a duas câmaras, cobrindo cada uma um pouco mais de metade do campo, foi essa situação que serviu de base à recolha dos dados que apresentamos.

A calibração dos conjuntos F_i usada pode ser observada nas figuras 8.1 a 8.3. Como é evidente a separação entre as cores é satisfatória embora o amarelo e o laranja estejam bastante próximos.

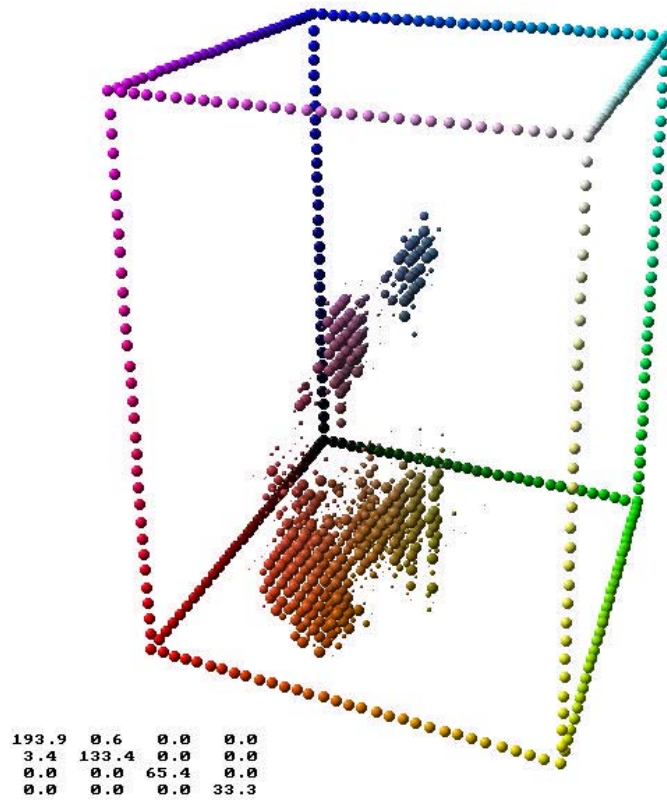


Figura 8.1: Os conjuntos $F_0...F_3$ usados nos exemplos seguintes

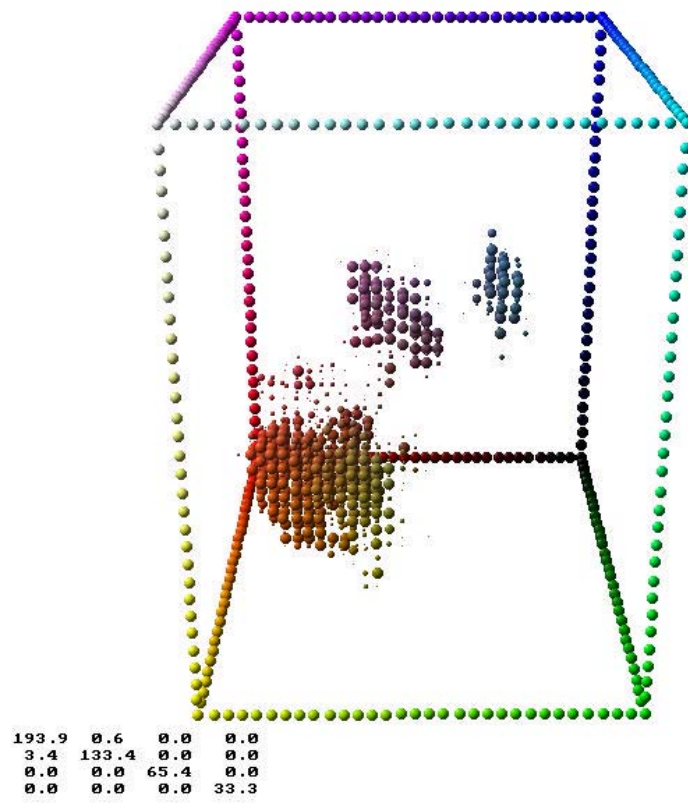
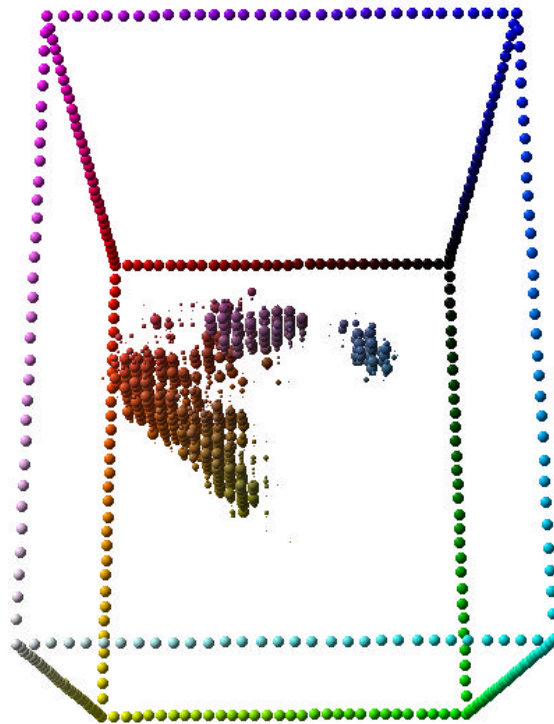


Figura 8.2: Os conjuntos segundo outro ângulo



193.9	0.6	0.0	0.0
3.4	133.4	0.0	0.0
0.0	0.0	65.4	0.0
0.0	0.0	0.0	33.3

Figura 8.3: Os conjuntos segundo uma vista de topo

8.2.1 Caso 1 - Uma Bola Parada

Neste caso apresentamos uma situação em que o nosso sistema adquiriu a posição e velocidade de uma bola parada. Com base nessa informação é possível caracterizar a covariância destas estimativas. O filtro de Kalman tem, neste caso, um contributo moderado pois estamos numa situação em que temos uma boa visibilidade da bola.

Isso pode ser atestado pela análise da média e da covariâncias associadas à posição observada e à estimada:

	Observado	Estimado
μ_x	0.7538	0.7537
μ_y	-0.3944	-0.3944
$cov(x, y)$	$\begin{bmatrix} 0.1229 & 0.0040 \\ 0.0040 & 0.1211 \end{bmatrix} \times 10^{-5}$	$\begin{bmatrix} 0.0964 & 0.0004 \\ 0.0004 & 0.1009 \end{bmatrix} \times 10^{-5}$
σ_x	0.0011	0.00098
σ_y	0.0011	0.0010

Tabela 8.1: Caso 1 - Estatísticas da Posição xy

Como podemos ver na tabela 8.2 este é um caso em que as condições de observação estão francamente favoráveis.

$Q = M(G_{bola}c_0)$	Valor
μ_Q	24.9
$cov(Q)$	9.98
$\max(Q)$	31.56
$\min(Q)$	12.75

Tabela 8.2: Caso 1 - Qualidade das observações

Na figura 8.4 temos as posições observadas da bola e as correspondentes estimativas. Podemos aí observar as estimativas ligeiramente mais próximas do valor médio. Na tabela 8.1 esse efeito está quantificado e podemos constatar que a covariância da posição estimada é ligeiramente inferior à da posição observada.

Na figura 8.5 temos a trajectória observada da bola e a correspondente trajectória estimada. O desvio padrão é de cerca de $1.1mm$ e a maior parte das observações encontram-se num círculo com cerca de $6mm$. Temos assim uma ideia da precisão relativa esperada.

Nas figuras 8.6 e 8.7 podemos observar a caracterização estatística das observações da posição da bola. Note-se a distribuição tipicamente gaussiana.

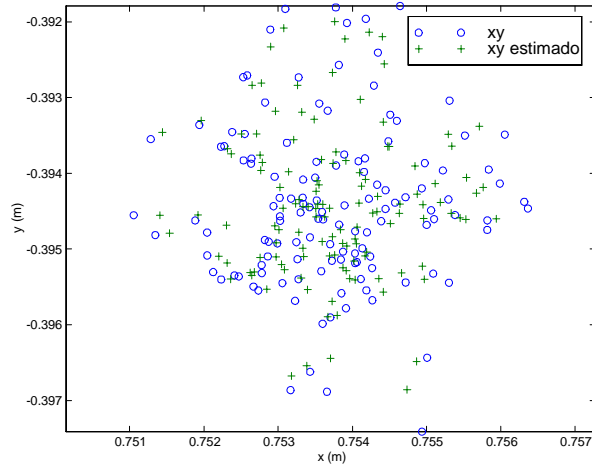


Figura 8.4: Caso 1 - Posição observada e estimada da bola

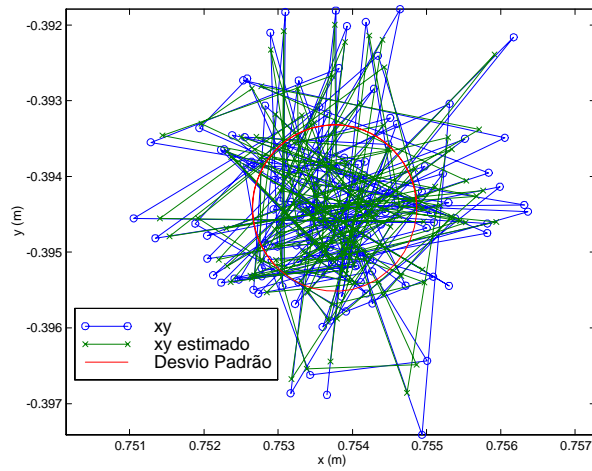


Figura 8.5: Caso 1 - Trajetória observada e estimada da bola

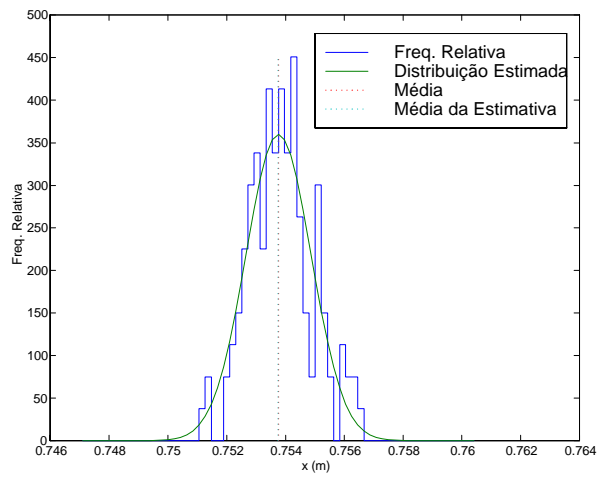


Figura 8.6: Caso 1 - Histograma da distribuição de x

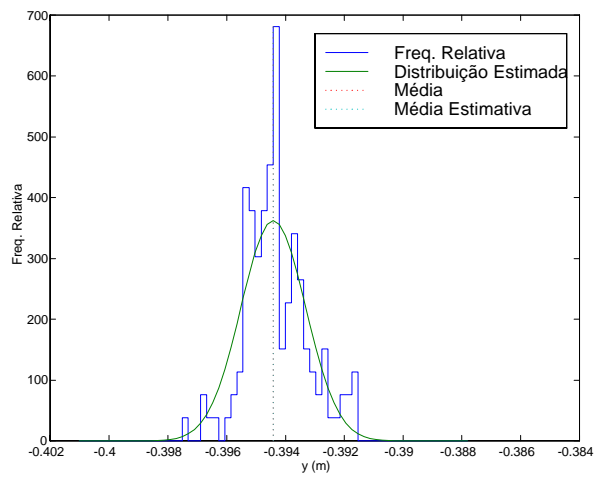


Figura 8.7: Caso 1 - Histograma da distribuição de y

Na tabela 8.3 encontramos a caracterização da velocidade estimada. Sabemos que o seu verdadeiro valor deveria ser zero logo, estas medidas dão-nos uma ideia imediata da precisão que podemos esperar da estimativa da velocidade, pelo menos numa situação estável.

	Velocidade Estimada
$\mu_{\hat{v}_x}$	2×10^{-4}
$\mu_{\hat{v}_y}$	-8.2×10^{-5}
$cov(\hat{v}_x, \hat{v}_y)$	$\begin{bmatrix} 0.7720 & 0.0131 \\ 0.0131 & 0.7495 \end{bmatrix} \times 10^{-4}$
$\sigma_{\hat{v}_x}$	0.0088
$\sigma_{\hat{v}_y}$	0.0087

Tabela 8.3: Caso 1 - Estatísticas da Velocidade

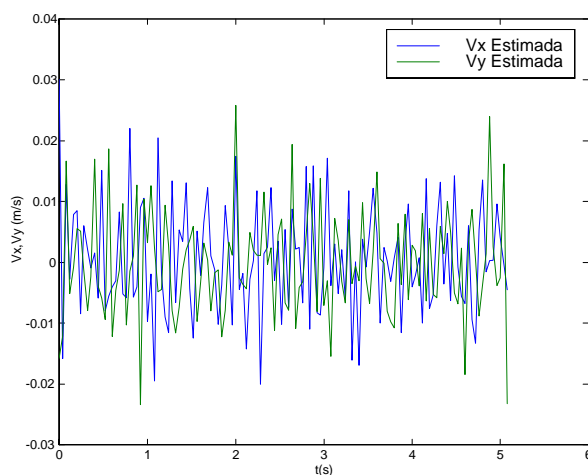


Figura 8.8: Caso 1 - Velocidades estimadas da bola

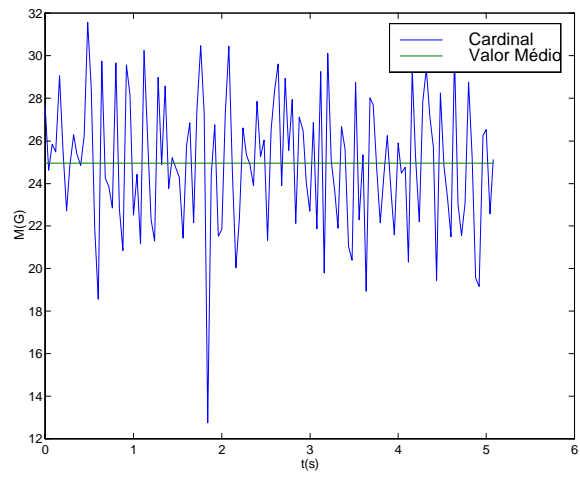


Figura 8.9: Caso 1 - Cardinal de $M (G_{bola}c_0)$

8.2.2 Caso 2 - A Bola em Movimento

Agora, apresentamos uma situação em que o nosso sistema adquiriu a posição e velocidade de uma bola em movimento. Esta aquisição foi realizada sob condições extremamente adversas. A bola inicialmente está parada durante cerca de 2 segundos, a seguir ela é empurrada e começa a atravessar o campo. Numa certa zona do campo montámos um túnel por onde a bola entra. Durante a parte em que a bola descreve a trajectória dentro do túnel o sistema de visão naturalmente não consegue localizar e é o filtro de Kalman que vai constringindo uma estimativa da sua posição. Após o reaparecimento da bola a estimativa e a observação fundem-se de novo para continuar a seguir a bola. Enquanto estamos numa situação em que temos uma boa visibilidade da bola o filtro de Kalman tem um contributo moderado, durante os outros instantes a sua operação é essencial.

Como podemos ver na tabela 8.4, neste caso as condições de observação estão semelhantes às do caso anterior em termos médios. Porém, há agora situações em que a visibilidade da bola desce consideravelmente até se perder por completo.

$Q = M(G_{bola}c_0)$	Instantes Iniciais	Trajectória Completa
μ_Q	20.52	17.95
$cov(Q)$	5.09	53.46
$\max(Q)$	26.92	27.98
$\min(Q)$	16.07	0

Tabela 8.4: Caso 2 - Qualidade das observações

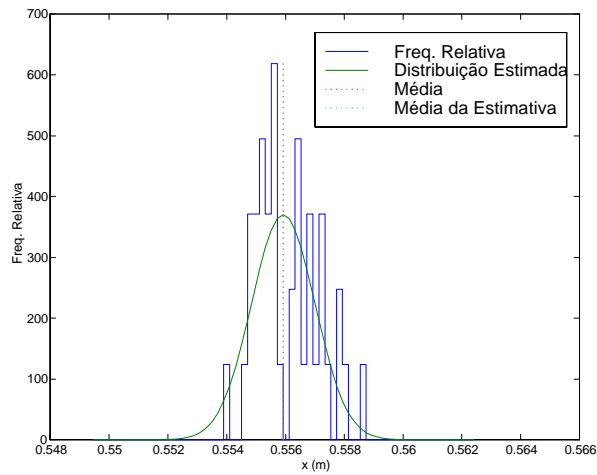


Figura 8.10: Caso 2 - Histograma da distribuição de x (instantes iniciais)

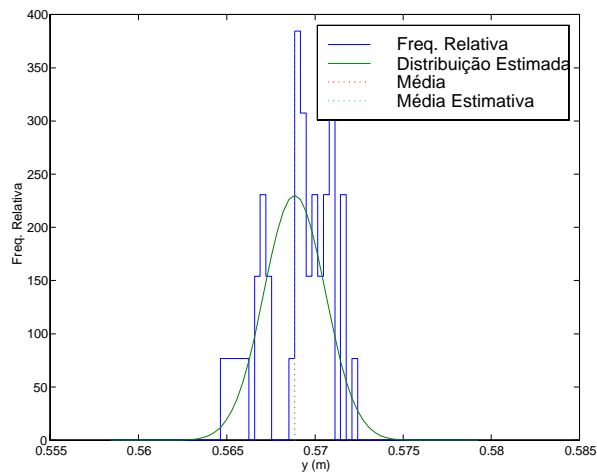


Figura 8.11: Caso 2 - Histograma da distribuição de y (instantes iniciais)

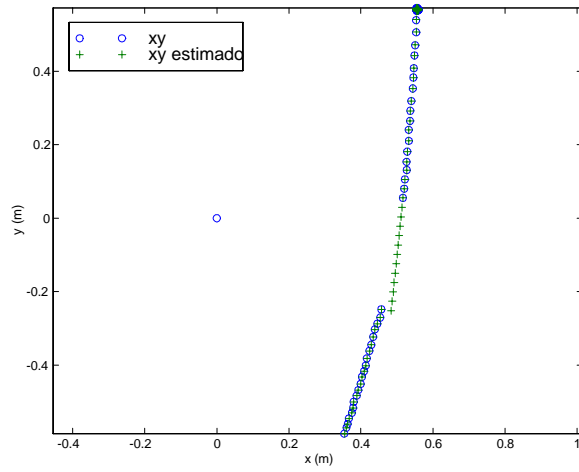


Figura 8.12: Caso 2 - Posição observada e estimada da bola

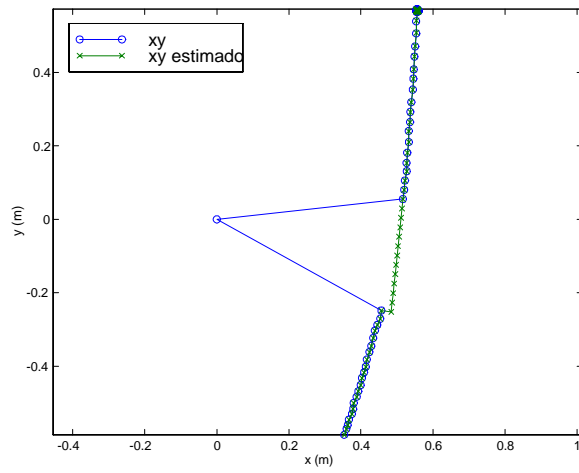


Figura 8.13: Caso 2 - Trajectória observada e estimada da bola

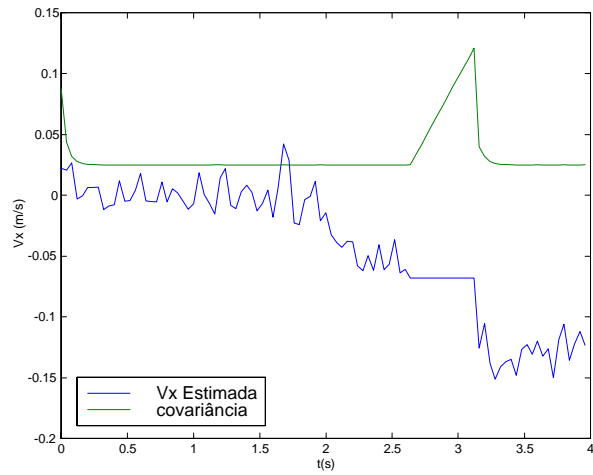


Figura 8.14: Caso 2 - Velocidade estimada segundo x e estimativa da respectiva covariância

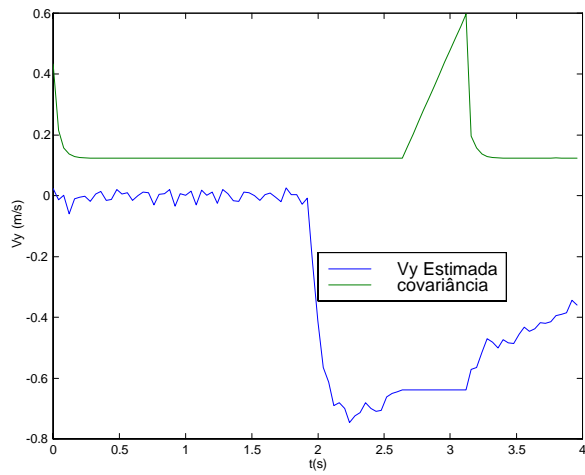


Figura 8.15: Caso 2 - Velocidade estimada segundo y e estimativa da respectiva covariância

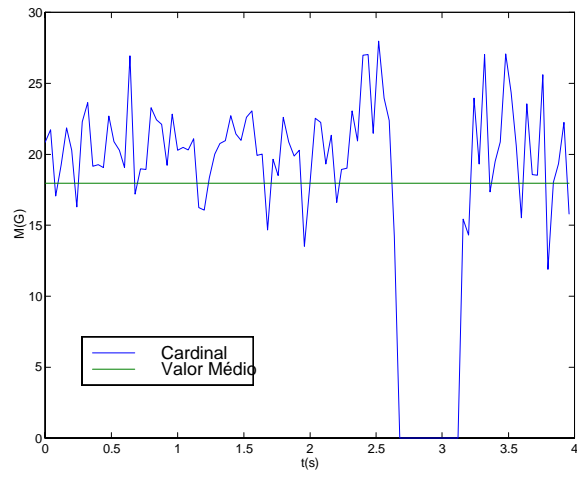


Figura 8.16: Caso 2 - Cardinal de $M (G_{bola}c_0)$

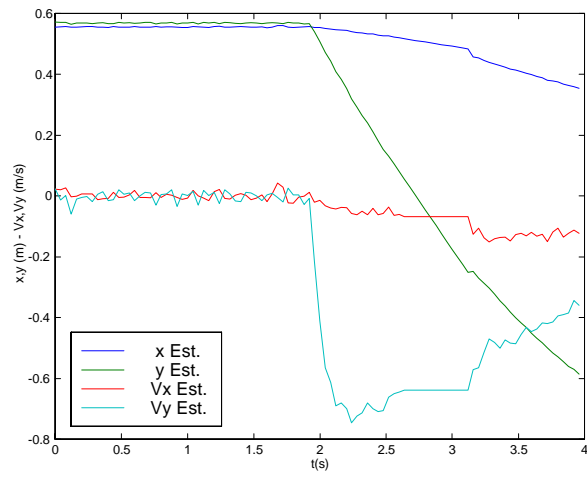


Figura 8.17: Caso 2 - Velocidades e posições estimadas

8.2.3 Caso 3 - O Robot 1 Estacionário

Agora, apresentamos uma situação em que o nosso sistema adquiriu a posição e velocidade de um robot que se encontrava parado.

Na tabela 8.5 podemos consultar a média e a covariância associadas à posição observada e à estimada:

	Observado	Estimado
μ_x	0.9195	0.9195
μ_y	-0.3891	-0.3891
μ_θ	0.3419	0.3352
$cov(x, y)$	$\begin{bmatrix} 0.0854 & -0.0386 \\ -0.0386 & 0.1232 \end{bmatrix} \times 10^{-5}$	$\begin{bmatrix} 0.7030 & -0.3028 \\ -0.3028 & 0.9986 \end{bmatrix} \times 10^{-6}$
σ_x	0.00092	0.00083
σ_y	0.00111	0.00099
σ_θ	0.07122	0.05126

Tabela 8.5: Caso 3 - Estatísticas da Posição xy

Como podemos ver na tabela 8.6, neste caso as condições de observação são excelentes. Pode-se até considerar o marcador secundário algo sobredimensionado.

$Q = M(G_{robot}c_1)$	Marcador Principal	Marcador Secundário
μ_Q	17.85	73.47
$cov(Q)$	7.09	178.76
$\max(Q)$	23.21	95.53
$\min(Q)$	11.30	37.34

Tabela 8.6: Caso 3 - Qualidade das observações

Na figura temos as posições observadas do Robot 1 e as correspondentes estimativas. Novamente, podemos observar as estimativas ligeiramente mais próximas do valor médio do que as medidas.

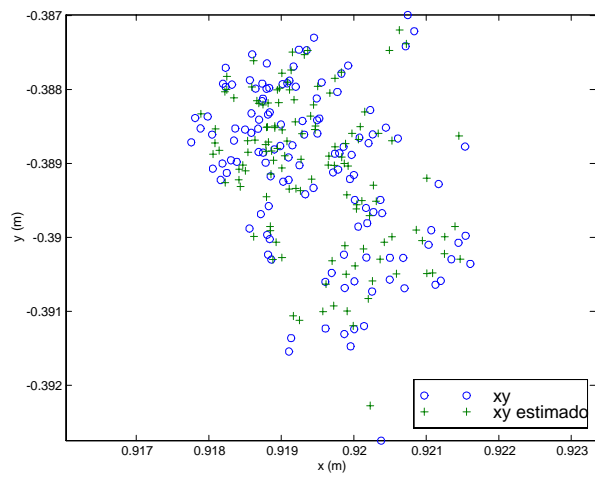


Figura 8.18: Caso 3 - Posição observada e estimada do Robot 1

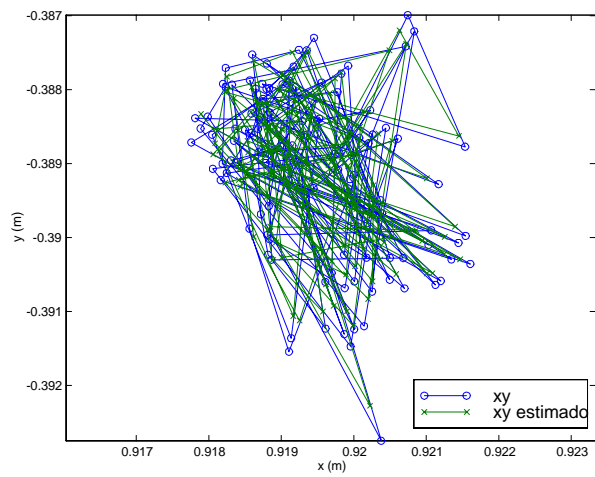


Figura 8.19: Caso 3 - Trajetória observada e estimada do Robot 1

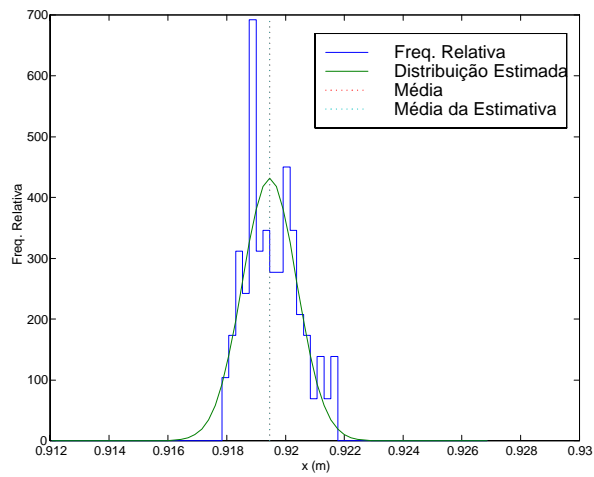


Figura 8.20: Caso 3 - Histograma da distribuição de x

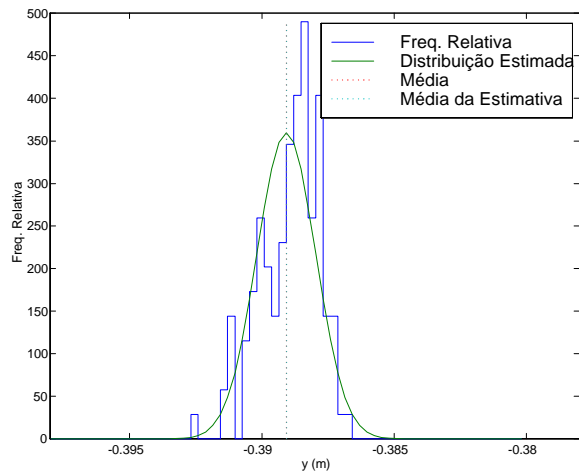


Figura 8.21: Caso 3 - Histograma da distribuição de y

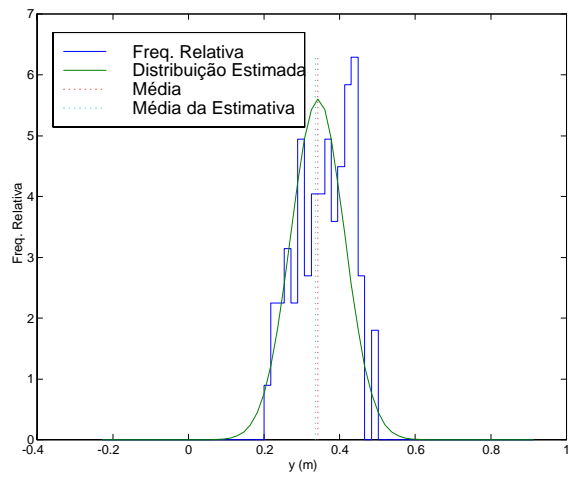


Figura 8.22: Caso 3 - Histograma da distribuição da atitude

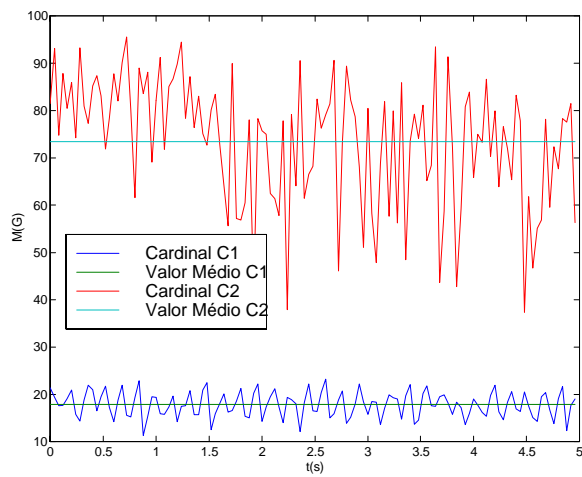


Figura 8.23: Caso 3 - $M(G_{mpC1})$

8.2.4 Caso 4 - O Robot 2 em Movimento

Agora, apresentamos uma situação em que o nosso sistema adquiriu a posição e velocidade de um robot que realizou uma série de manobras. Partindo de uma zona perto da linha do meio campo o Robot 2 descreveu uma trajectória rectilínea cortando o campo na diagonal. À entrada da área parou e iniciou um movimento de rotação segundo o seu eixo.

Como podemos ver na tabela 8.7, neste caso as condições de observação ainda são excelentes. Nota-se porém, que em certas regiões do campo o marcador secundário não apresenta uma visibilidade tão alta como na posição final.

$Q = M(G_{mx}c_2)$	Marcador Principal	Marcador Secundário
μ_Q	17.03	49.50
$cov(Q)$	15.23	236.63
$\max(Q)$	24.66	73.62
$\min(Q)$	0	0

Tabela 8.7: Caso 4 - Qualidade das observações

Na figura 8.24 temos as posições observadas do Robot 2 e as correspondentes estimativas. Neste caso verificámos que houve duas situações em que o sistema de visão não conseguiu identificar o robot na imagem. Quando isso acontece, a posição do robot não está definida e no gráfico aparece localizado no ponto (0,0). Aí, o filtro de Kalman fornece uma estimativa que poderá ser usada para mascarar este problema momentâneo.

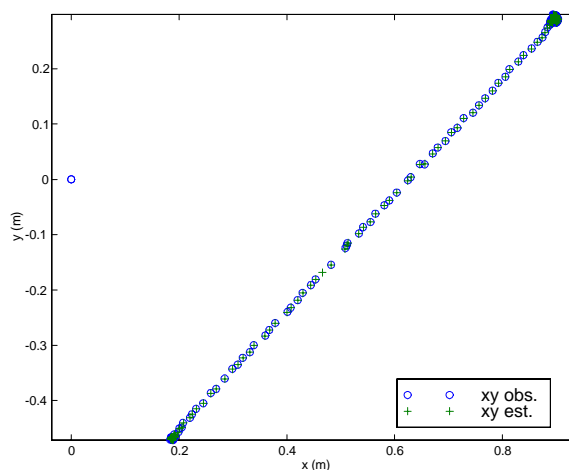


Figura 8.24: Caso 4 - Posição observada e estimada do Robot 2

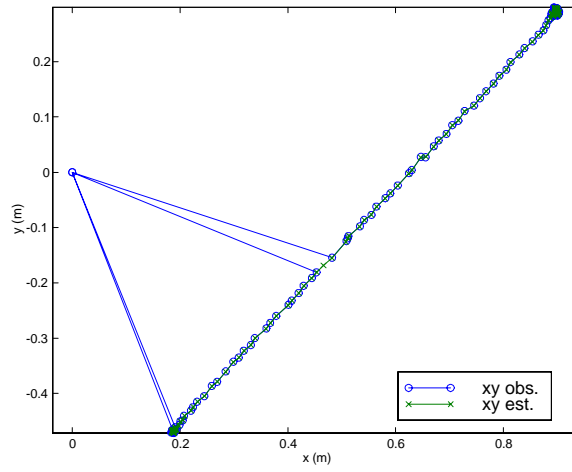


Figura 8.25: Caso 4 - Trajectória observada e estimada do Robot 2

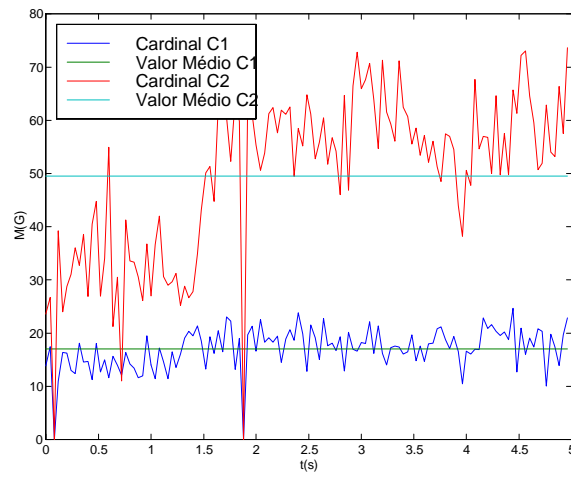


Figura 8.26: Caso 4 - $M(G_{mx}C_2)$

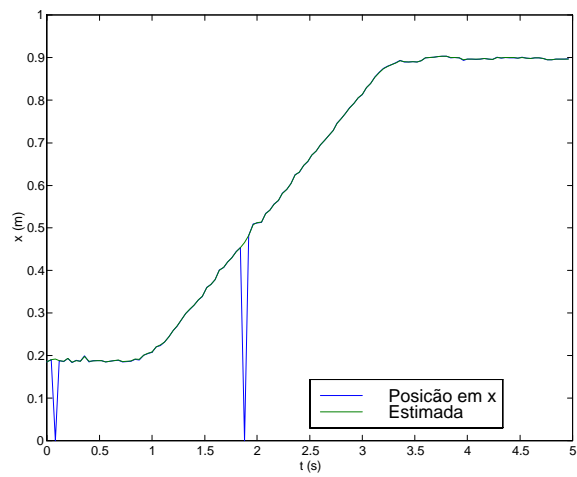


Figura 8.27: Caso 4 - Posição observada e estimada segundo x

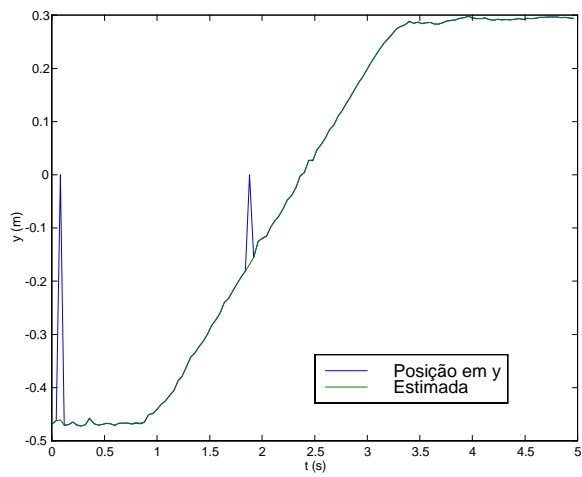


Figura 8.28: Caso 4 - Posição observada e estimada segundo y

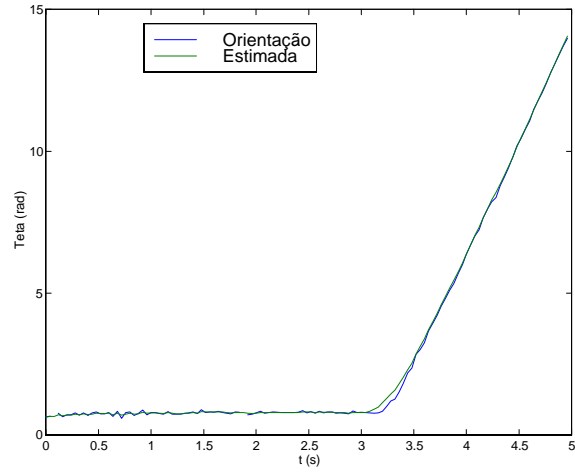


Figura 8.29: Caso 4 - Atitude medida e estimada do Robot 2

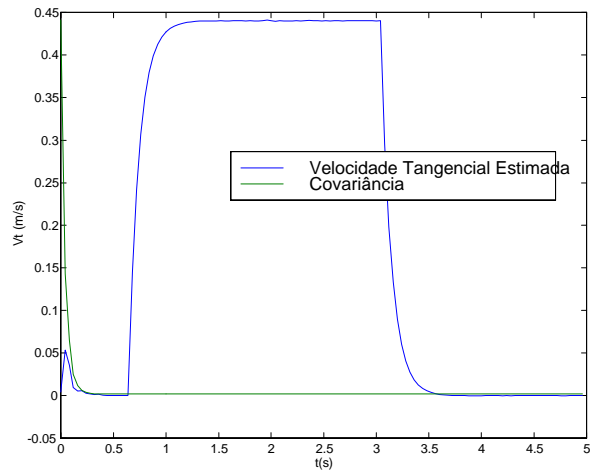


Figura 8.30: Caso 4 - Estimativa da velocidade tangencial e respectiva covariância

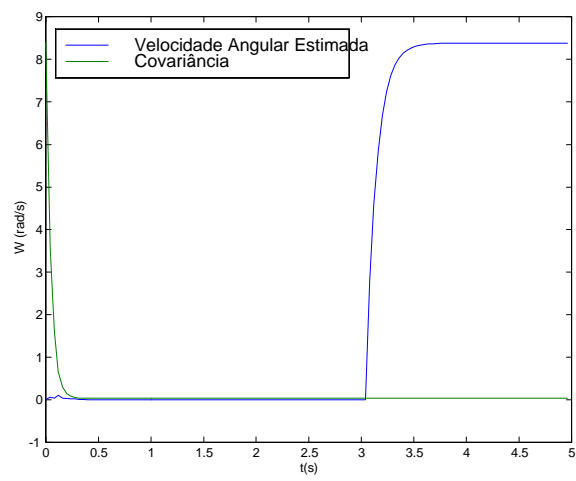


Figura 8.31: Caso 4 - Estimativa da velocidade tangencial e respectiva covariância

Capítulo 9

Conclusões

9.1 Trabalho Desenvolvido

Neste trabalho abordámos o problema da localização de múltiplos robots e de uma série de parâmetros do ambiente. Para tal recorremos a um mecanismo sensorial baseado em visão. Conseguimos aperfeiçoar a nossa estimativa do estado recorrendo a um esquema de filtragem baseada em Filtros de Kalman. Nos casos em que a dinâmica era não linear recorremos ao filtro de Kalman Extendido.

Para dispor de uma plataforma de trabalho padronizada, atacámos o problema proposto pela *RoboCup Federation*. Neste problema é necessário controlar uma equipa de cinco robots que procuram marcar o máximo de golos a outra equipa de robots. Assim temos de localizar não só os nossos cinco robots como os cinco robots do adversário e também a bola. Este processo deve ser efectuado a uma frequência considerável e com o mínimo de atraso, o que introduz considerações de tempo real importantes.

Sintetizámos a nossa solução baseada em conceitos da teoria de conjuntos *fuzzy*. Consequentemente, apresentámos sucintamente os conceitos mais importantes.

O sistema proposto permite, graças a uma classificação *fuzzy* dos pixels da imagem, reduzir consideravelmente a dimensão dos dados a processar. Essa é a chave de um processamento em tempo real baseado apenas em Hardware de uso geral.

Dado que este sistema permite, através do estimador proposto, também extrair uma medida da incerteza associada à estimativa da posição dos marcadores, é possível implementar um filtro de Kalman onde as suas capacidades de previsão e fusão de informação são aproveitada de uma maneira eficaz.

O esquema de calibração proposto, permite ainda um ajuste incremental que facilita o refinamento progressivo da qualidade da classificação. Torna-se possível, interactivamente, verificar e ajustar a calibração que está a ser

efectuada. Inclusivamente, o esquema proposto de construção e erosão do histograma, que serve de base à estimação dos conjuntos que representam o mapeamento marcador-cor no espaço rgb , permite um ajuste muito fino e preciso desses conjuntos. Apresentámos também uma série de métricas que permitem a avaliação da qualidade da calibração, em termos da correcta distinção dos diferentes marcadores.

A introdução de um esquema de filtragem baseado num filtro de Kalman permitiu alcançar um nível extra de precisão pois possibilitou o recurso a conceitos de memória e plausibilidade baseados no conhecimento da dinâmica dos robots.

9.2 Trabalho Futuro

9.2.1 Múltiplas Câmaras

Há claramente uma série de possíveis direcções a tomar para melhorar e expandir o trabalho desenvolvido. Talvez uma das mais importantes, em termos do desempenho do sistema, seja a possibilidade de recorrer a mais do que uma câmara para aquisição da posição da bola. A motivação para este desenvolvimento surge devido às restrições impostas na altura da câmara relativamente ao campo. Estas restrições podem advir de imposições espaciais inerentes ao local onde o campo está montado ou devido a imposições regulamentares. Um problema do primeiro tipo enfrentamos, por exemplo, no nosso laboratório onde a altura do tecto limita a altura máxima a que a câmara pode ser colocada. Por outro lado as actuais regras do RoboCup impõem uma altura máxima de $3m$ para a câmara. Essa regra já foi aplicada em 99 no campeonato mundial realizado em Estocolmo. Anteriormente, no campeonato mundial de Paris em 98 a altura da câmara em relação ao campo foi de cerca de $4.5 m$. Uma altura inferior leva a vários problemas:

- Limita a disponibilidade de câmaras pois as mais vulgares não dispõem de um campo de visão angular suficiente.
- Normalmente leva ao recurso a sistemas ópticos que introduzem uma distorção significativa.
- Piora o problema da ocultação da bola pelo corpo dos robots.

Este último é o problema mais grave, pois com robots que apresentem uma altura próxima da máxima (normalmente, $15 cm$ mas pode chegar a $22.5 cm$ para aqueles que apresentem visão local) verifica-se que haverá situações onde a bola pode ficar completamente ocultada. A maior parte das vezes, a dinâmica do jogo e a ajuda dada pelo filtro de Kalman empregue, conseguem que este problema não afecte significativamente o desempenho.

Apesar disso, há casos em que se nota um pior desempenho motivado exactamente por este factor.

Para aliviar este problema podemos recorrer a mais do que uma câmara. Colocando, por exemplo, em vez de uma câmara sobre a vertical do centro do campo, duas câmaras sobre os centros dos meios campos, podemos amenizar os efeitos de ocultação. Graças a esta técnica a zona de sombra para um robot com as faces planas e 15 *cm* de altura, estando a bola à entrada da área, passa de 46 *mm* para cerca de 18 *mm*.

Pode tornar-se então necessário descentralizar o processamento da imagem caso não seja possível efectuar esse processamento somente num PC. Felizmente, para centralizar a informação bastará que os computadores encarregues de processar as imagens comuniquem as posições observadas dos robots, acompanhadas da respectiva confiança em cada observação. Assim o sistema de processamento de imagem, tal como foi descrito, poderá ser usado quase sem alteração. O único ponto mais sensível será a necessidade de lidar com o facto do número de robots presente em cada imagem variar significativamente ao longo do jogo. Um outro ponto a ter em atenção é a zona de sobreposição onde podemos ter um robot visto pelas duas câmaras. Essa situação terá de ser tratada com o devido cuidado para não dar origem a descontinuidades na posição quando um robot deixa de ser seguido por uma câmara e passa a ser seguido por outra.

O problema mais delicado será introduzido pelo facto de passarmos a ter informação que chega a intervalos de tempo que já não estão fixos. A menos que as duas câmaras estejam sincronizadas entre si, algo que não é possível conseguir com uma câmara comum, as observações dos robots e da bola chegarão em instantes que dependem da câmara que está a seguir esse objecto. Isso implica um esquema de controlo mais complexo pois deverá levar em conta esse factor.

Uma vantagem extra do recurso a mais de uma câmara é o equivalente aumento na resolução da imagem que estamos a tratar. Se mantivermos a resolução mas passarmos a observar apenas metade do campo, isso equivale a melhorar a resolução em ambas as direcções por um factor de $\sqrt{2}$. Imediatamente a precisão será melhorada de um factor semelhante. Como passamos a ter mais pixeis observados por cada marcador podemos esperar uma melhoria extra na precisão.

Uma extensão natural desta ideia é possibilitar o recurso a várias câmaras localizadas de modo a maximizar a precisão das observações.

9.2.2 Generalização do Algoritmo de Visão

O bom desempenho deste sistema para resolver o problema de localização de robot da liga F-180 faz-nos confiar na possibilidade de o expandir com sucesso para problemas semelhantes que aparecem, por exemplo, na liga F-2000. As principais diferenças de liga F-2000 do RoboCup em relação à liga

F-180 são:

- As dimensões do campo. Temos um campo com cerca de 9 *m* de comprimento por 5 *m* de largura. Naturalmente todas as outras dimensões físicas do campo escalam de acordo.
- Os robots podem ter cerca de 50 *cm* de diâmetro e até 80 *cm* de altura.
- Cada equipa é constituída no máximo por 4 robots.
- Está proibido recurso a sistemas de visão global, isto é, as câmaras deverão apenas estar montadas nos próprios robots. É, no entanto, permitido o recurso a um sistema de computação central, que pode estar fora do campo.

As experiências já efectuadas foram bastante promissoras. Verificámos que muitas vezes não se conseguia um efeito de redução do número de pixeis a processar tão significativo, pois os marcadores a seguir podem ficar muito maiores na imagem dado que as distâncias em questão são muito variáveis. Porém, o nosso esquema de calibração e classificação dos pixeis parece suficientemente geral para poder ser usado com sucesso neste problema.

9.2.3 Processamento Extra sobre a Imagem da Bola

Ao contrário dos outros marcadores, a bola pode ser parcialmente ocultada. Quando isso acontece, a fórmula usada para extrair o centro da bola a partir do centro dos pixeis associados introduzirá algum erro. Os pixeis ocultados, naturalmente não conseguem contribuir para o cálculo do centro da bola e teremos a nossa estimativa desviada para a zona visível. Para atacar este problema seria necessário um processamento extra directamente sobre a imagem. Várias técnicas poderiam ser experimentadas:

- Optimizando a correlação da forma da bola com uma figura correspondente a um círculo cortado por um plano poderíamos escolher a configuração que melhor se ajusta à imagem. Depois de estimado o corte seria possível estabelecer a localização do centro da bola.
- A técnica anterior poderia ser aplicada recorrendo também à geometria do Robot que oculta a bola.

9.2.4 Estudo da Robustez e Optimização do Estimador da Posição de um Marcador

O estimador usado é uma aproximação muito simples, que mostrou um desempenho perfeitamente aceitável. No entanto chegámos a verificar que o grau de pertença de um pixel estava correlacionado com a distância ao

centro do marcador. Caracterizando essa relação seria possível construir uma função que estimando a distância do pixel ao centro do respectivo marcador serviria para definir a covariância dessa observação do centro. A partir desse pressuposto seria viável proceder à síntese de um estimador para as coordenadas do centro e respectiva covariância.

9.2.5 Introdução de um Mecanismo de Adaptatividade no Mapeamento das Cores

Actualmente, usamos uma série de amostras iniciais de onde construímos o mapeamento que classifica as cores. Durante o jogo e treinos há um quantidade considerável de situações que poderiam ser aproveitadas para um crescente refinamento deste mapeamento. Ao introduzir essa capacidade possibilitaríamos a introdução de uma componente adaptativa no sistema de visão. É claro que este mecanismo terá de ser cuidadosamente formulado de modo a não introduzir situações de instabilidade.

Bibliografia

- [1] Robert Grover Brown, Patrick Y. C. Hwang, "Introduction to Random Signals and Applied Kalman Filtering", John Wiley & Sons, third edition, 1997.
- [2] Eduardo Caridade, "Veículo Autónomo Didático", Tese de Mestrado, FEUP, 1995.
- [3] J. L. Martins de Carvalho, "Dynamic Systems and Automatic Control", Prentice Hall, 1993.
- [4] Paulo Costa, "Identificação, Modelização e Controlo de um Veículo Móvel Autónomo", Tese de Mestrado, FEUP, 1995.
- [5] Paulo Costa, António Moreira, Armando Sousa, Paulo Marques, Pedro Costa, Aníbal Matos, "5dpo Team Description", Robocup 1999 - Robot World Cup Soccer Games and Conference, Stockolm, Sweeden, pp 85-89, 1999.
- [6] Paulo Costa, António Moreira, Armando Sousa, Paulo Marques, Pedro Costa, Aníbal Matos, "5dpo-2000 Team Description", Robocup 1999 - Robot World Cup Soccer Games and Conference, Stockolm, Sweeden, pp. 217-220, 1999.
- [7] Paulo Costa, António Moreira, Armando Sousa, Paulo Marques, Pedro Costa, "5dpo Team Description", Robocup 1998 - Robot World Cup Soccer Games and Conference, Paris, France, pp 511-517.
- [8] C. Fantuzzi, "Bases of Fuzzy Control", Proceedings of the International Summer School, pp. 1-35, 1998.
- [9] Dinis Fernandes, "Arquiteturas de Seguimento Visual e Captura por um Manipulador Robótico de Objectos em Movimento", Tese de Mestrado, Instituto Superior Técnico, Lisboa, 1997.
- [10] Arthur Gelb, "Applied Optimal Estimation", The M.I.T. Press, 1989.
- [11] K. R. Godfrey, "Correlation Methods", Automatica, Vol 16, pp. 527-534, Pergamon Press Ltd, 1980.

- [12] Kwun Han, Manuela Veloso, "Reactive Visual Control of Multiple Non-Holonomic Robotic Agents", School of Computer Science, Carnegie Mellon University, Proceedings of 1998 IEEE International Conference on Robotics and Automation (ICRA '98), May 16-20, Leuven, Belgium, 1998.
- [13] Kwun Han, Manuela Veloso, "Physical Model Based Multi-objects Tracking and Prediction in RoboSoccer", Working Notes of the AAAI 1997 Fall Symposium on Model-directed Autonomous Systems, November 8-10th, Cambridge, Massachusetts, 1997.
- [14] Eugene, Hecht, "Óptica", Fundação Calouste Gulbenkian, 1991.
- [15] George Klir, Bo Yuan, "Fuzzy Sets and Fuzzy Logic", Prentice Hall, 1995.
- [16] Bart KOSKO, "Neural Networks and Fuzzy Systems, a Dynamical Systems Approach to Machine Intelligence", Prentice Hall, 1992.
- [17] A. N. Kolmogorov, S. V. Fomin, "Elementos da Teoria das Funções e de Análise Funcional", Editora Mir, 1982.
- [18] Carlos Machado, Fernando Campos, Manuel Martins, Sérgio Sampaio, Fernando Ribeiro, "Mechanics and Electronics of the Footballer Autonomous Mobile Robot", University of Minho, Dep. of Industrial Electronics, Guimarães, 1998.
- [19] Carlos Machado, Fernando Campos, Fernando Ribeiro, "Footballer Autonomous Mobile Robot - Control and Vision System", University of Minho, Dep. of Industrial Electronics, Guimarães, 1998.
- [20] Rudolf Mäusl, "Television Technology", Rohde & Schwarz, Munique, 1979.
- [21] Armando de Moraes, "Dicionário de Inglês-Português", Porto Editora, 3ª Edição, 1998.
- [22] William K. Pratt, "Digital Image Processing", Second Edition, Wiley-Interscience, 1991.
- [23] Pedro Silva, "Navegação Acústica em Ambientes Estruturados", Tese de Mestrado, FEUP, 1995.
- [24] V. Strejc, "Least Squares Parameter Estimation", Automatica, Vol 16, pp. 535-550, Pergamon Press Ltd, 1980.
- [25] Robocup Federation, "RoboCup F-180 League Rules", 1999.

- [26] E. Uchibe, M. Asada, S. Noda, Y. Takahashi and K. Hosoda, "Vision-Based Reinforcement Learning for RoboCup : Towards Real Robot Competition", Proc. of IROS 96 Workshop on RoboCup, 1996.
- [27] Manuela Veloso, Peter Stone, Kwun Han, Sorin Achim, "The CMUnited-97 Small Robot Team" in Proceedings of RoboCup-97: The First Robot World Cup Soccer Games and Conferences, H. Kitano (ed.), Springer Verlag, Berlin, 1998.
- [28] Manuela Veloso, Peter Stone, Kwun Han, "Prediction, Behaviors, and Collaboration in a Team of Robotic Soccer Agents", in Proceedings of International Conference on Multi Agent Systems (ICMAS'98), Cite des Sciences - La Villette, Paris, France, July 4 - 7, 1998.
- [29] Manuela Veloso, Peter Stone, Kwun Han, "The CMUnited-97 Robotic Soccer Team: Perception and Multiagent Control", in Proceedings of Second International Conference on Autonomous Agents, Second International Conference on Autonomous Agents (Agents '98), Minneapolis/St. Paul, May 9-13, 1998.
- [30] L. A. Zadeh, "Fuzzy Sets", Inform. Control, 1965.

Apêndice A

Descrição do Hardware

□ Câmaras:

Camcorder Blaupunkt S-VHS modelo CCR-650S

Câmara de vigilância Topica TP-60001A

□ Placas de Aquisição baseada no integrado BT848/849 da Brooktree.

Video Highway Extreme

AverMedia Video Capture

Genius Video Wonder II Pro

□ Computadores:

Versão 98 - PC com Processador Intel Pentium II 266MHz e 64 Mega-Bytes de RAM

Versão 99 - PC com Processador Intel Pentium II 450MHz e 128 Mega-Bytes de RAM

Apêndice B

Desenvolvimento do Software

B.1 Ferramentas de Desenvolvimento

- Sistema Operativo: DOS

Ocasionalmente, durante o desenvolvimento recorreremos ao Windows 95-98, mas a versão final do sistema é usada em DOS.

- Compilador de C/C++: Djgpp

Este compilador é uma adaptação para DOS do compilador da GNU, GCC. Está disponível, livre, em "<http://www.delorie.com/djgpp/>".

- Ambiente de desenvolvimento: Rhide

Este ambiente de desenvolvimento, emula o funcionamento do último ambiente da Borland para DOS e por isso é extremamente familiar. Outra vantagem é a sua disponibilidade, livre, tanto para DOS como para Linux. Pode ser encontrado em "<http://www.tu-chemnitz.de/~rho/rhide.html>".

B.2 Sistema Desenvolvido

O sistema desenvolvido, foi implementado exclusivamente em linguagem C e C++. A opção de basear o sistema em MS-DOS e não noutra sistema operativo mais complexo tal como Windows ou Linux deveu-se à falta de garantias de tempo real. Qualquer um destes outros sistemas operativos pode, e frequentemente faz isso, retirar o controlo ao programa durante o que para nós são longos milisegundos. Outras soluções, tais como sistemas operativos projectados expressamente para tarefas de tempo real, para além do preço tipicamente elevado revelam frequentemente problemas ao nível da disponibilidade de *drivers* para os periféricos de que necessitávamos.

Para fornecer um mínimo de funcionalidade ao nível de interface com o utilizador recorreremos a uma biblioteca disponível, livre e com o código fonte disponível, chamada *Allegro*.

Está disponível em "<http://www.talula.demon.co.uk/allegro/>".