

# Tutorial introdutório para as competições de futebol robótico

José A. Carvalho Gonçalves, Pedro H. Portela Pinheiro, José L. S. Magalhães Lima e Paulo J. C. Gomes da Costa

**Title**—Introductory tutorial for robotic soccer competitions

**Abstract**—Soccer was the original motivation for Robocup. Besides being a very popular sport worldwide, soccer brings up a set of challenges for researchers while attracting people to the event, promoting robotics among students, researchers and general public. RoboCup chose to use soccer game as a central topic of research, aiming at innovations to be applied for socially significant problems and industries.

This paper describes the implementation of a system similar to the used by the teams participating in the Robocup small size league (SLL). The system, developed in Object Pascal, allows real time localization and control of an omnidirectional mobile robot. The objective of this paper is to be an introductory tutorial for Robocup soccer games, focusing in the base topics real time localization and control.

**Index Terms**—Artificial vision, Mobile robots, Real time systems.

## I. INTRODUÇÃO

As competições de *robots* são eventos apropriados para a experimentação, investigação e desenvolvimento em muitas áreas relacionadas com ciência e tecnologia [1]. Servem também de motivação à investigação e à aprendizagem pois inspiram os alunos a abordar matérias que poderiam parecer complexas ou desinteressantes fora do enquadramento fornecido pelo projecto.

Este artigo descreve detalhadamente um sistema inspirado no que é utilizado pelas equipas participantes na Small Size League (SLL) do Robocup, o qual serve de bibliografia introdutória sob a forma de tutorial para os aspectos essenciais das competições do Robocup, tais como localização e controlo em tempo real. É claro que certos aspectos relativos ao desempenho foram relaxados para permitir uma maior simplicidade no projecto e assim torná-lo adequado a um trabalho didáctico assim como permitir o uso de material (câmaras, Sistema de aquisição, PC para o processamento) mais comum e barato.

Esta descrição permite assim duas abordagens didácticas potencialmente interessantes: pode ser usado como um tutorial, quase um guião, para um aluno que queira implementar um sistema robótico funcional, por outro lado pode permitir a

José Gonçalves e José Lima são professores assistentes do departamento de Electrotecnia da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Bragança, Portugal. Email: {goncalves, jllima}@ipb.pt

Pedro Pinheiro foi aluno de licenciatura de Engenharia Informática da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Bragança, Portugal, trabalha neste momento na empresa AEIOU. Email: ei9212@alunos.ipb.pt

Paulo Costa é professor Auxiliar do departamento de Engenharia Electrotécnica e de computadores da Faculdade de Engenharia da Universidade do Porto, Portugal. Email: paco@fe.up.pt

DOI (Digital Object Identifier) Pendiente

construção de um sistema que possibilita a abordagem de diferentes áreas associadas à robótica e motiva trabalhos circunscritos a essas áreas, por exemplo, localização baseada em visão, fusão sensorial, projecto de controladores, geração de trajectórias, navegação, etc.

O sistema apresentado é versátil, sendo uma mais valia em várias etapas de aprendizagem no domínio da robótica móvel. É utilizado para validação de controladores e táticas (monitorizando em tempo real jogos segundo as regras do Robocup Junior), permite a validação de sistemas de localização e navegação para ambientes estruturados (monitorizando o desempenho do sistema de localização e navegação de um *robot*) e é também utilizado em actividades de investigação (estudo de controlo e técnicas de fusão sensorial).

### A. Robocup junior

O *Robocup Junior* [2] consiste na realização de jogos de futebol entre equipas de 1 ou 2 *robots*, cada um deles devendo caber dentro de um cilindro com o diâmetro de 18 cm. A prova decorre num campo pintado com níveis de cinzento para os *robots* determinarem a sua orientação e posição. A bola emite radiação infravermelha o que permite aos *robots* determinarem a sua posição. Esta competição está associada a alunos de escolas secundárias, os quais desenvolvem trabalhos extra-curriculares pelo gosto da aprendizagem, motivados pelo desafio e pela competição. O sistema desenvolvido permite colocando um marcador em cima de um *robot* monitorizar o seu desempenho, tal como descrito na secção III. Com esta ferramenta de validação podem-se comparar dados de vários controladores, servindo de apoio à escolha do melhor controlador. Os *robots* desenvolvidos pelos alunos para esta competição utilizam tipicamente a tecnologia *Lego Mindstorms* [3], aproveitando as suas potencialidades relativas à prototipagem rápida [4] [5].

### B. Localização e navegação em ambientes estruturados

Existem várias competições cujo objectivo é a localização e navegação de *robots* móveis em ambientes estruturados, tais como o Robô Bombeiro do Instituto Politécnico da Guarda e o concurso Micro-Rato da Universidade de Aveiro. Como exemplo serão apresentadas as regras do concurso Micro-Rato [6]. O concurso Micro-Rato é uma competição entre pequenos *robots* móveis e autónomos, com dimensões não excedendo os  $300 \times 300 \times 400$  mm. No decurso da competição, estes *robots* devem cumprir dois objectivos em sequência: o primeiro objectivo é ir desde a "área de partida" até à "área de farol"; o segundo objectivo consiste em regressar à área de

partida, ou aproximar-se dessa área o mais possível utilizando, para tal, a informação recolhida durante o cumprimento do primeiro objectivo. Em qualquer dos casos os *robots* devem evitar todos os obstáculos que se lhes deparem, incluindo outros *robots*. O farol emite radiação infravermelha o que permite aos *robots* determinarem a sua posição.

O sistema descrito na secção III pode ser usado para validar os sistemas de localização e navegação desenvolvidos pelos alunos, comunicando este por rede com o *robot*. Isto é feito durante a navegação de um *robot*, gerando-se um ficheiro que regista a trajectória real (Sistema de visão global) e a estimada (sistema de localização desenvolvido pelos alunos). O ficheiro gerado é um auxiliar valioso para a análise e avaliação do desempenho dos algoritmos de navegação e localização.

A comunicação por rede com o *robot* (figura 1), foi realizada utilizando o protocolo UDP, sendo o mais apropriado para sistemas de tempo-real. Esta utilização deve-se, sobretudo, a factores de velocidade no envio dos pacotes. Se se tiver em conta que num sistema de tempo-real a 25 Hz são enviados 25 pacotes de dados pela rede por segundo, não é obrigatoriamente necessário ter a certeza que o pacote é entregue, e caso o seja, se é na ordem correcta. Como o envio/recepção é constante, na eventualidade de um pacote não ser entregue, o pacote seguinte é recebido num período de tempo tão reduzido que torna a falha praticamente imperceptível.

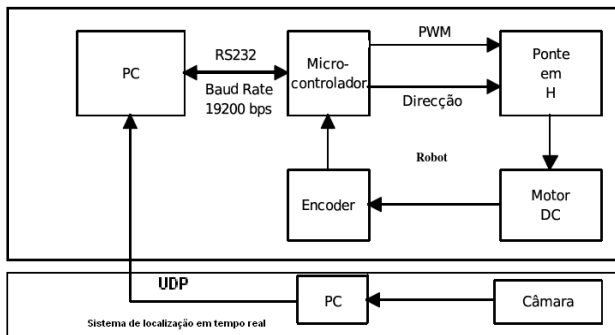


Figura 1. Sistema de validação do sistema de navegação

### C. Investigação em navegação, localização e controlo de robots móveis no domínio do Robocup

O futebol serviu de inspiração para a definição do *Robocup*, pois para além de ser um jogo bastante popular em todo o mundo, o que atrai visitantes a estes eventos, tem um conjunto muito significativo de desafios para os investigadores, pelo facto de ser um jogo cooperativo em ambiente dinâmico [2].

Tratando-se o futebol robótico de um ambiente dinâmico, é necessário que a informação da posição do *robot* e da *bola* esteja disponível com o mínimo de atraso possível [7], sendo que para a aplicação descrita, se está a falar de requisitos de tempo-real a 25 Hz, ou seja, 25 *frames* por segundo.

A utilização de *robots* omnidireccionais nas competições de futebol robótico também traz vantagens tais como diminuição do tempo de reacção do *robot* e a diminuição do número de manobras. O facto de o *robot* se poder deslocar com

velocidades lineares e angular controladas, contribui também para a simplificação na estratégia de jogo [8].

O sistema de localização em tempo real está a ser utilizado para actividades de investigação apresentando-se o exemplo do controlo de um *robot* omnidireccional de três rodas na secção IV. Neste nível de aprendizagem para que possam ser aplicados algoritmos de fusão sensorial é essencial que as medidas relativas (velocidade de cada roda) e as medidas absolutas (visão) estejam disponíveis no mesmo instante. Por esse motivo o controlador deve estar na mesma aplicação que o sistema de localização, evitando-se ao máximo atrasos, por exemplo de transmissão de dados por rede. Um diagrama de blocos de um controlador que tem acesso a estes dados no mesmo instante de tempo está exemplificado na imagem 9 da secção IV.

## II. SISTEMA DE CAPTURA DE IMAGEM

Fisicamente, o sistema representado na figura 2, é constituído pelos seguintes componentes:

- **Câmara:** Sony Video8 XR SteadyShot;
- **Placa de Aquisição de Vídeo:** Pinnacle - chipset Bt878 (driver: Conexant's BtPCI WDM Video Capture);
- **PC:** AMD Athlon 64BIT 3500+, 1024 MB Ram, placa gráfica GECUBE ATI RADEON 512MB DDR e Windows XP Professional SP2;



Figura 2. Sistema de captura de imagem

Assim, este sistema consiste em:

- Colocar uma câmara perpendicular à zona de acção da bola e do *robot* (tapete verde, que simula um relvado). A câmara está presa no centro de uma estrutura metálica, que permite uma altura máxima de 3 metros. Neste caso a câmara está colocada a 2 metros. Quanto maior for a altura da câmara em relação ao relvado, menor será o erro devido à paralaxe, reduzem-se problemas de ocultação da bola e o campo de visão angular da câmara aumenta, porém a qualidade de imagem baixa e a distorção em barril é maior. Estes aspectos serão discutidos em pormenor nas subsecções seguintes.

#### – Paralaxe:

O efeito de paralaxe é minimizado aumentando a altura da câmara. Na figura 3 pode-se ver que para uma altura  $h_2 > h_1$  o erro de paralaxe é reduzido consideravelmente [9].

Este factor pode ser compensado se a altura a que estão colocados os marcadores (círculos coloridos) no *robot* for conhecida. Este mecanismo de compensação foi implementado, permitindo uma maior precisão na determinação da posição quer da bola quer do *robot*. Os

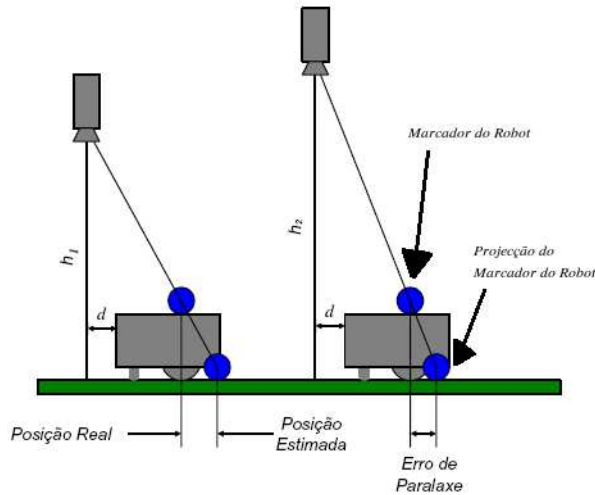


Figura 3. Erro de paralaxe para diferentes alturas da câmara

marcadores são utilizados para extrair informação sobre a localização absoluta do *robot* móvel.

#### – Ocultação da bola:

Este é um problema ainda mais grave que a paralaxe. Para uma câmara colocada a uma distância insuficiente há situações em que o corpo do *robot* oculta, parcial ou totalmente, a bola. Tipicamente, esta situação ocorre quando a bola fica encostada ao *robot*, ou então quando a sombra do *robot* coincide com a bola, provocando também que a bola seja ocultada. Nestas situações, e uma vez que o *robot* não possui chuto, a solução passa pela intervenção humana, ou seja, é necessário posicionar a bola. Na aplicação desenvolvida, para evitar erros, nestas situações a bola é dada como estando fora da zona de acção admissível.

#### – Qualidade da Imagem:

O conceito de qualidade, neste caso, prende-se com o facto de, quanto mais alta estiver colocada a câmara, menos serão os pixels que servirão para identificar os marcadores. Sendo estes de dimensões reduzidas poderá haver situações em que os marcadores não tenham um número suficiente de pixels que o permitam identificar e consequentemente o localizar.

#### – Distorção em barril:

É uma das distorções ópticas das objectivas mais frequente, que consiste no facto de nas extremidades da imagem as linhas rectas ficarem curvas, com a forma de um barril. É causada por uma construção assimétrica, na qual o diafragma está colocado na parte anterior do sistema óptico.

Como no sistema que se está a tratar a área de acção é de  $1 \text{ m}^2$  e a câmara está colocada a uma altura de 2 metros, o efeito da distorção em barril é visível, mas por ser uma área de dimensões reduzidas, este efeito não acrescenta um erro de localização significativo, sendo por isso desprezado. Caso o erro não fosse desprezado, este efeito poderia ser corrigido recorrendo a uma função de

mapeamento pixel-mundo capaz de compensar esse tipo de distorção.

### III. SISTEMA DE LOCALIZAÇÃO

Nesta secção será feita uma descrição do sistema que, a partir da imagem capturada, permite extrair a informação necessária para poder localizar os marcadores.

Assim sendo, o sistema de visão deve possuir as seguintes características:

#### – Precisão:

Uma vez que estamos a lidar com um *robot* com 16 cm de diâmetro (forma circular) e uma *bola* com 4 cm de diâmetro, é necessário que o erro na sua localização não exceda alguns milímetros.

#### – Rapidez:

Tratando-se de um ambiente dinâmico, é necessário que a informação da posição do *robot* e da *bola* esteja disponível com o mínimo de atraso possível. Por forma a perceber a importância desta característica, descreve-se o seguinte cenário: atendendo a velocidades na ordem dos  $2 \text{ ms}^{-1}$ , quer para a *bola* quer para o *robot*, é fácil de perceber que um atraso de 100 ms pode ser considerado muito grave. Este é um intervalo de tempo suficiente para que o *robot* e/ou a *bola* se desloquem cerca de 20 cm. Para que estas situações não aconteçam, os requisitos de tempo-real têm de ser cumpridos, sendo que neste caso, se está a falar de requisitos de tempo-real a 25 Hz, ou seja, 25 *frames* por segundo.

#### – Robustez:

A interpretação da imagem pode, por vezes, dar origem a medidas de posição completamente erradas devido a uma má identificação dos marcadores. Uma das condições mais importantes, senão mesmo a mais importante, é a iluminação. Por vezes, a presença de sombra, por mais ténue que seja, pode alterar significativamente a cor, contribuindo assim, para eventuais erros, quer na calibração da cor, quer na detecção e localização de marcadores.

#### A. Dos Pixels para o Estado do Sistema

Considerando que a partir deste momento se dispõe da representação digital da imagem do relvado, esta será constituída por uma matriz de elementos de imagem  $p_{xy}$ , designados por pixels, contendo cada um uma representação da área de imagem abrangida. Cada pixel corresponde a um vector com três componentes que representam a intensidade das três componentes de cor: vermelho, verde e azul (normalmente designadas de R, G e B). Assim, pode-se definir a seguinte função:

$$\begin{aligned} RGB : I &\longrightarrow \mathfrak{R}^3 \\ p_{xy} &\longrightarrow (r, g, b) \end{aligned} \quad (1)$$

que, para cada pixel, indica a cor associada, sob a forma de um vector com as intensidades das três componentes. Estas componentes são normalmente discretizadas em,

respectivamente,  $n_r$ ,  $n_g$  e  $n_b$  bits. Tipicamente, para  $n_r = n_g = n_b = 8$ , num total de 24 bits, o que já permite uma representação em que o efeito da discretização é visualmente indetectável.

### B. Sistema de Coordenadas

Como o objectivo é localizar o *robot* e a *bola* no relvado é necessário fixar a origem e a orientação dos eixos do sistema de coordenadas a utilizar. Para a origem foi escolhido o centro do relvado e considera-se o eixo dos  $xx$  alinhado longitudinalmente em relação ao relvado.

Cada pixel  $p_{xy}$  pode ser encontrado numa matriz em que  $x$  indica a linha e  $y$  a coluna do pixel. Neste 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, onde  $x_m$  e  $y_m$  definem a dimensão da imagem.

Este sistema de coordenadas reflecte o de uma câmara PAL, explorando-se 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.

Para facilitar a distinção entre as coordenadas ( $x$ ,  $y$ ) correspondentes a uma localização no mundo e as correspondentes ao pixel de coordenadas ( $x$ ,  $y$ ), esta última passará a ser definida pelos conjuntos  $Nx$  e  $Ny$ :

$$\begin{aligned} Nx &= \{0, 1, \dots, x_m - 1\} \\ &\quad \text{e} \\ Ny &= \{0, 1, \dots, y_m - 1\} \end{aligned}$$

### C. Mapeamento Imagem - Mundo

Para extrair as localizações dos objectos observados, a partir da imagem, tem que se construir a seguinte função:

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

A equação 2 mapeia coordenadas 2D da imagem em coordenadas do mundo. Esta função fornece as coordenadas  $x$  e  $y$  assumindo que a componente em  $z$  é zero. Como os objectos em questão (*robot* e *bola*), não estão colocados à mesma altura, haverá erro devido ao fenómeno de paralaxe. Este erro pode ser compensado caso a altura dos objectos seja conhecida. Com este valor pode-se construir uma função que ajusta  $x$  e  $y$  tendo em conta a altura dos objectos e da câmara. Este ajuste só pode ser realizado após uma localização do objecto.

### D. Correção da Paralaxe

Na prática, com a imagem adquirida, apenas se consegue extrair informação relativa aos objectos no plano  $xy$ . Conhecendo  $z$  pode-se corrigir o valor de  $\tilde{x}$ ,  $\tilde{y}$  para obter  $x$ ,  $y$  já devidamente compensados no que diz respeito à paralaxe.

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

$$\begin{aligned} L : \mathbb{R}^3 &\longrightarrow \mathbb{R}^2 \\ (\tilde{x}, \tilde{y}, z) &\longrightarrow (x, y) \end{aligned} \quad (3)$$

com

$$x = L_x(\tilde{x}, \tilde{y}, z) = \tilde{x} - \frac{z}{h} \quad (4)$$

$$y = L_y(\tilde{x}, \tilde{y}, z) = \tilde{y} - \frac{z}{h} \quad (5)$$

onde  $h$  representa a altura a que está colocada a câmara [9].

### E. Calibração da Cor

1) *Escolha dos Marcadores*: Um dos factores mais importantes na calibração da cor, é a escolha dos marcadores que serão utilizados. É através destes que se consegue detectar e localizar quer a *bola* quer o *robot*. No que diz respeito à *bola* a decisão recaiu em usar a cor laranja, uma vez que é a cor mais usada neste tipo de aplicações (caso do futebol robótico, quer na liga SSL quer na MSL [2]).

Para o *robot* a escolha já foi mais cuidadosa. Sabendo à partida que seriam necessários dois marcadores diferentes, um para identificar o centro do *robot* e outro que permitisse saber o ângulo que o *robot* assume em cada instante. Sendo o relvado de cor verde e a *bola* de cor laranja, as cores para o *robot* tinham que ser as mais afastadas possível no cubo RGB. Assim, as cores escolhidas foram o azul para o centro do *robot* e o amarelo para o ângulo, sendo estas as cores oficiais para a diferenciação das equipas na liga SSL [2]. O *robot* utilizado está ilustrado na figura 4.



Figura 4. Protótipo do *robot* omnidireccional

2) *Calibração de Marcadores*: A calibração de marcadores, de uma forma geral, consiste em 'dizer' ao sistema que um determinado marcador fica associado a uma determinada cor. Desta forma, torna-se muito simples encontrar um determinado objecto numa imagem, bastando para isso percorrer o cubo RGB e procurar a cor com que esse objecto foi calibrado.

Neste caso, é necessário efectuar a calibração da *bola* e dos dois marcadores do *robot*, que ficarão associados às seguintes cores:

- **Bola**: mapeada com a cor vermelha;

- **Centro do robot:** mapeada com a cor verde;
- **Ângulo do robot:** mapeada com a cor azul;

A calibração é efectuada com o rato, escolhendo o objecto a calibrar e clicar nesse mesmo objecto na imagem de manipulação. Quantas mais vezes se clicar na mesma zona mais pontos ficam calibrados, aumentando assim a qualidade de calibração. Estes pontos ficam guardados num cubo RGB (array tridimensional de dimensões  $(255 \times 255 \times 255)$ ).

Outro factor importante na calibração de marcadores é a incidência de luz. Assim deve-se calibrar os marcadores em diferentes locais para que, desta forma, possa ficar com mais informação sobre os marcadores, para diferentes intensidades de luz.

### F. Selecção da Área Activa

Outra das vantagens da calibração de posições, é a possibilidade de se poder seleccionar somente a área onde se pretende localizar os marcadores, isto porque, haverá quase sempre zonas da imagem que representam áreas para as quais não há interesse procurar marcadores. Como o relvado não ocupa toda a imagem, existe sempre uma banda em volta da imagem (figura 5) que mostra o chão que o rodeia.

Este processo permite que os algoritmos de pesquisa de marcadores sejam mais rápidos, consequentemente baixando o tempo de processamento, uma vez que, por norma, a área activa vai ser de dimensões menores em relação à imagem adquirida.

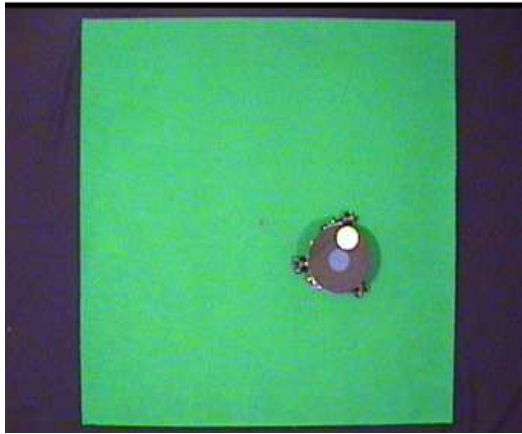


Figura 5. Imagem Captura completa

Assim, para cada linha  $y$ , são indicadas 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:

$$\begin{aligned} V_A : Ny &\longrightarrow Nx \times Nx \\ y &\longrightarrow (x_1, x_2) \end{aligned} \quad (6)$$

A principal vantagem desta representação é que já não obriga a interrogar cada pixel para verificar se ele deve

ser ou não processado. Considerando um cenário em que o processamento está a ser feito segundo um varrimento de linhas, desde o canto superior direito da imagem e pesquisando os pixels da esquerda para a direita:

- Uma linha que não contenha pixels para serem processados pode ser imediatamente descartada. Para isso basta a avaliação da função  $V_A$  para essa linha indicar  $x_2 < x_1$ .
- Numa linha que tenha alguns pixels para tratar, pode-se avançar de imediato para o pixel  $x_1$  descartando todos os pixels anteriores e pode-se parar o processamento quando se atingir  $x_2$ . Assim serão descartados automaticamente os pixels que faltam para acabar a linha.

Outra vantagem, que por si só já justificaria a restrição do processamento da imagem a uma zona activa, é que poderá haver no cenário todo o tipo de objectos, que não haverá interferências mesmo que as suas cores coincidam com as dos marcadores.

### G. Detecção e Localização dos Marcadores

É importante referir que, apesar de existirem referências constantes à *bola*, esta, não deixa de ser um marcador tal como os marcadores do *robot*. Nesta secção explica-se o algoritmo que permite, não só, detectar os marcadores como localizar a posição em que estes se encontram. Convém lembrar que os marcadores têm a mesma forma geométrica (circulares), sendo por isso este algoritmo aplicável a qualquer marcador, mudando apenas o objecto que se pretende localizar.

1) *Detecção dos Marcadores*:- Estando os marcadores mapeados com uma determinada cor na matriz de calibração, basta efectuar uma pesquisa nesta mesma matriz pelo código *RGB* correspondente ao marcador em questão. A partir do momento em que se encontram pixels com este código pode-se concluir que o objecto está a ser detectado.

2) *Localização dos Marcadores*:- Uma vez detectado um pixel calibrado com a cor do objecto que se pretende localizar, é guardada a coordenada em  $x$  e  $y$  da imagem. Este processo é repetido para toda a área activa, sendo que as coordenadas são acumuladas. No fim de percorrer esta área é calculada a média da soma de todas as coordenadas, o que equivale a dizer que se fica com a localização do centro da circunferência, ou seja do marcador.

Assim, sendo  $I$  a imagem que contém os dados da calibração dos objectos e admitindo que pretendemos localizar o marcador que foi mapeado com o código  $r = 255$ ,  $g = 0$  e  $b = 0$ , ou seja, a *bola*, pode-se definir a seguinte função:

$$\begin{aligned} I_{i,j} \cdot R &= 255 \\ \tilde{x} &= \tilde{x} + i \\ \tilde{y} &= \tilde{y} + j \\ NPontos &= NPontos + 1 \end{aligned} \quad (7)$$

da qual resulta o total acumulado para as duas coordenadas, bem como o número total de pontos que foram detectados. Pode-se então admitir que a posição em que o marcador se encontra é dada por:

$$x = \frac{\tilde{x}}{NPontos} \quad (8)$$

$$y = \frac{\tilde{y}}{NPontos} \quad (9)$$

Contudo, e por este processo ser simples, é preciso ter especial atenção em alguns aspectos. Um deles, foi anteriormente descrito e já está resolvido, que é o facto de objectos que estão fora da área activa não poderem entrar no calculo da média.

Agora considerem-se os seguintes cenários:

– **Cenário 1:**

Dada a altura da câmara e as reduzidas dimensões dos marcadores, podem ocorrer situações em que o marcador calibrado não tenha mais que dois ou três pontos calibrados. Será que se pode confiar na posição que é retornada? E se estes pontos nem se encontram juntos, ou muito próximos, será que se pode admitir estes pontos pertencem mesmo ao marcador?

– **Cenário 2:**

Admitindo que é sempre possível acontecerem ocorrências de ruído aleatório (por numerosas causas), podem ocorrer situações em que existem bastantes pixels a caracterizar um marcador e existirem pixels com o mesmo código *RGB* espalhados pela imagem. Deverão estes pontos entrar no cálculo da média?

De seguida serão apresentadas as soluções que foram adoptadas para solucionar estes possíveis aspectos.

• **Solução para Cenário 1:**

A solução para este caso, passa por admitir que a *bola* não está dentro da área de acção activa. Assim, se forem encontrados menos que 5 pixels admite-se que estes pixels não representam um marcador e é indicada uma posição que depois de ser sujeita ao mapeamento imagem - mundo, dará uma posição fora da área activa.

• **Solução para Cenário 2:**

A solução para este caso, passa por calcular uma nova posição, contudo só serão incluídos para o cálculo da média os pixels que estejam numa vizinhança em relação à posição anteriormente calculada. Desta forma todos os pixels que estejam espalhados pela imagem e que não pertençam ao marcador não serão incluídos, corrigindo o erro fornecido pela posição anterior.

Como as dimensões dos marcadores são reduzidas, a vizinhança estabelecida para definir os pixels pertencem ao marcador, tem uma dimensão de  $20 \times 20$ .

Desta forma às funções descritas anteriormente em (7), (8) e (9) serão acrescentadas as seguintes funções :

$$I_{i,j}.R = 255 \wedge i$$

$$\exists [x - 20, x + 20] \wedge j \exists [y - 20, y + 20]$$

$$\tilde{x} = \tilde{x} + i$$

$$\tilde{y} = \tilde{y} + j$$

$$NPontos = NPontos + 1(10)$$

$$x = \frac{\tilde{x}}{NPontos} \quad (11)$$

$$y = \frac{\tilde{y}}{NPontos} \quad (12)$$

**H. Determinação do Ângulo do robot**

Como já foi referido anteriormente na subsecção (III-G), os marcadores de cor azul e amarelo são utilizados para detecção e localização do *robot*. Assim, o marcador azul permite saber a posição em que o *robot* se encontra, e o marcador amarelo permite saber a orientação que o *robot* toma num determinado momento.

A informação útil que se pode retirar dos marcadores (para este caso em concreto), é a posição do centro de  $(x,y)$  de cada um.

Considerando  $P_r(x, y)$  a posição fornecida pelo marcador azul e  $P_a(x, y)$  a posição fornecida pelo marcador amarelo, pode-se definir o vector que une  $P_r$  a  $P_a$  tendo como parâmetros  $a$  e  $b$ , tal como exemplificado na figura 6.

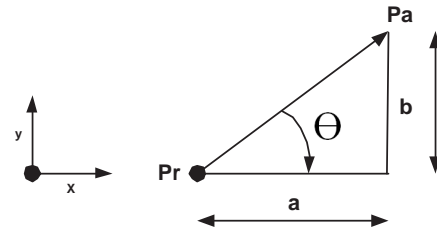


Figura 6. Ângulo do *robot* em relação ao sistema de eixos

Assim, as componentes do vector em  $x$  e  $y$  são dadas por  $a$  e  $b$  respectivamente:

$$a = P_a(x) - P_r(x) \quad (13)$$

$$b = P_a(y) - P_r(y) \quad (14)$$

Desta forma é possível saber o ângulo do *robot*, recorrendo à operação trigonométrica:

$$\theta = \arctan \frac{b}{a} \quad (15)$$

Sendo esta operação realizada pela função  $\arctan2(b,a)$ , a qual recebe 2 parâmetros, resolvendo o problema relacionado com divisão por zero e calculando o ângulo independentemente do quadrante em que se encontre.

**I. Estudo do erro do sistema de localização**

Foi realizado para o sistema de localização uma análise das distribuições de probabilidade do erro [10][11], aproximadas a distribuições normais [12][13], apresentado-se os resultados em unidades SI.

O número de pixels obtidos para o marcador azul ( $Q1$ ), afecta a variância do erro na localização em  $x$  e  $y$ , tal como exemplificado na tabela seguinte:

Tabela I

$Q1$	$x$	$y$
5-10	1,5E-05	1,9E-05
10-20	9,25E-06	7,36E-06
20-30	4,84E-06	4,86E-06
30-40	4,15E-06	3,80E-06
$\geq 40$	1,96E-06	2,21E-06

Por sua vez a variância do erro do ângulo é afectada pelo número de pixels obtidos para ambos os marcadores, azul ( $Q1$ ) e amarelo ( $Q2$ ), tal como exemplificado na tabela seguinte:

Tabela II

$Q1$	5-10	10-20	20-30	30-40	$\geq 40$
5-10	0,14	8E-02	1,2E-02	1E-02	6,2E-03
10-20	1,6E-02	9,9E-03	1,3E-02	6,6E-03	4,6E-03
20-30	1,5E-02	9,9E-03	7,2E-03	4,9E-03	3,9E-03
30-40	1,4E-02	9,5E-03	5,9E-03	4,4E-03	2,9E-03
$\geq 40$	1,4E-02	7,2E-03	5,77E-03	3E-03	3E-03

$Q2$

A variância para menos que 5 pixels não é apresentada, pois para esta situação considera-se que a confiança no sensor é nula.

#### IV. CONTROLADOR DO *robot* MÓVEL

Mecanicamente o *robot* possui três rodas motrizes omnidireccionais desfasadas de  $120^\circ$  (figura 7), permitindo movimentos em todas as direcções com velocidade controlada. O sistema de accionamento é constituído por três motores de corrente contínua com caixa redutora. Para se efectuar o controlo da velocidade de cada uma das rodas é usada modulação de largura de impulsos (PWM), controlando-se independentemente cada uma das rodas. O controlo dos motores é feito em malha fechada utilizando um micro-controlador AVR programado em C e um PC com uma aplicação em *Object Pascal*, comunicando entre si usando a norma RS232.

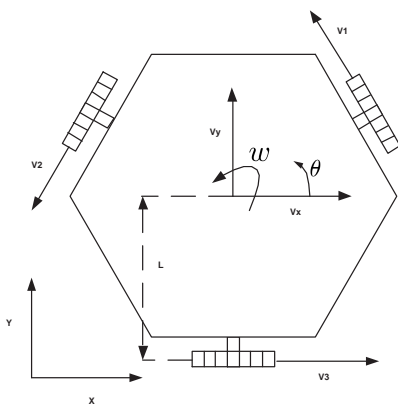


Figura 7. Geometria de um *robot* omnidireccional de três rodas.

A odometria e o controlador são calculados no PC comunicando este com o micro-controlador. O micro-controlador envia para o PC as transições dos *encoders* relativas a cada roda, com estes dados possibilita-se o cálculo do erro da

velocidade para controlo em malha fechada da velocidade dos motores. O controlador tem como parâmetros de saída a velocidade a que devem rodar os motores para que o *robot* alcance um determinado objectivo. O diagrama de blocos do sistema está representado na figura 8.

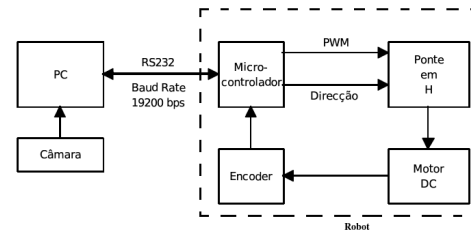


Figura 8. Diagrama de blocos do sistema

O controlador tem como objectivo a deslocação do *robot* para um determinado ponto com velocidade controlada, o seu diagrama de blocos é apresentado na figura 9. Para controlar o *robot* é necessário controlar independentemente cada uma das rodas, as quais lhe vão imprimir movimento. O controlador tem diversas hierarquias, sendo a sua função a mais baixo nível controlar independentemente cada uma das rodas e a mais alto nível determinar a que velocidade deve girar cada roda para que o *robot* se desloque com uma determinada velocidade angular e linear. O controlo a mais alto nível é efectuado baseando-se no sistema de equações (16) e num controlador de posição. A mais baixo nível cada motor é controlado em malha fechada usando-se um controlador PI.

$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} = \begin{pmatrix} -\sin(\theta) & \cos(\theta) & L \\ -\sin(\frac{\pi}{3} - \theta) & -\cos(\frac{\pi}{3} - \theta) & L \\ \sin(\frac{\pi}{3} + \theta) & -\cos(\frac{\pi}{3} + \theta) & L \end{pmatrix} \begin{pmatrix} V_x \\ V_y \\ w \end{pmatrix} \quad (16)$$

Este controlador já tinha sido parcialmente implementado, não sendo utilizada visão artificial para a localização, sendo a estimação do posicionamento efectuada com base no cálculo da odometria [14].

#### A. Desempenho do controlador

Com o objectivo de analisar o desempenho do controlador foi realizada uma corrida, estando descrita pelo fluxograma da figura 10. Como podemos observar o *robot* desloca-se passando por vários pontos efectuando deste modo uma trajectória, fazendo evoluir a máquina de estados sempre que um objectivo seja alcançado.

É possível concluir por observação dos gráficos da figura 11 que se torna vantajosa a utilização de *robots* omnidireccionais no futebol robótico pois se podem fazer trajectórias em todas as direcções com velocidades lineares e angular controladas sem que seja necessário variar o ângulo previamente. Apesar disso é frequente o ângulo tenha que ser corrigido pois surgem variações em relação à orientação desejada para o *robot*. Estas variações devem essencialmente ao tipo de trajectória realizada, por exemplo mudanças bruscas de direcção, arranques e travagens fazem com que as rodas plissem, perdendo-

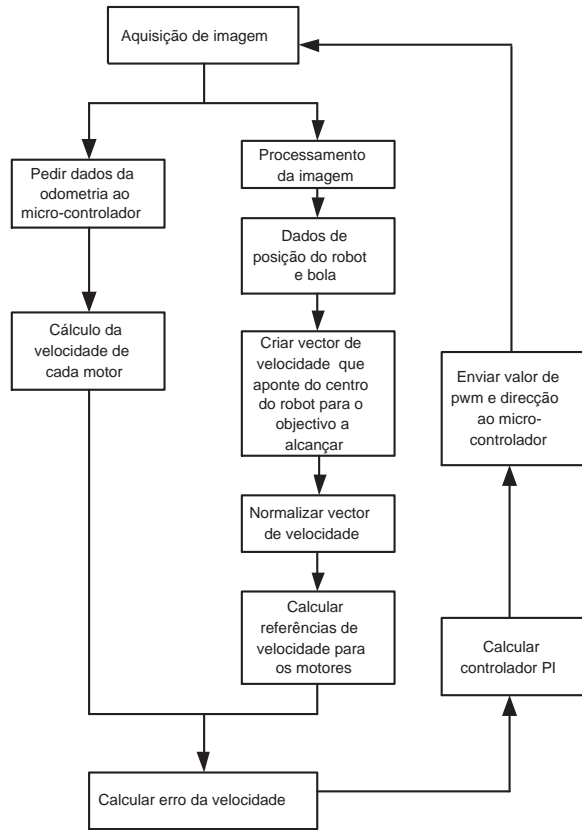


Figura 9. Diagrama de blocos do controlador

se tracção e consequentemente piorando o desempenho do controlador do *robot*.

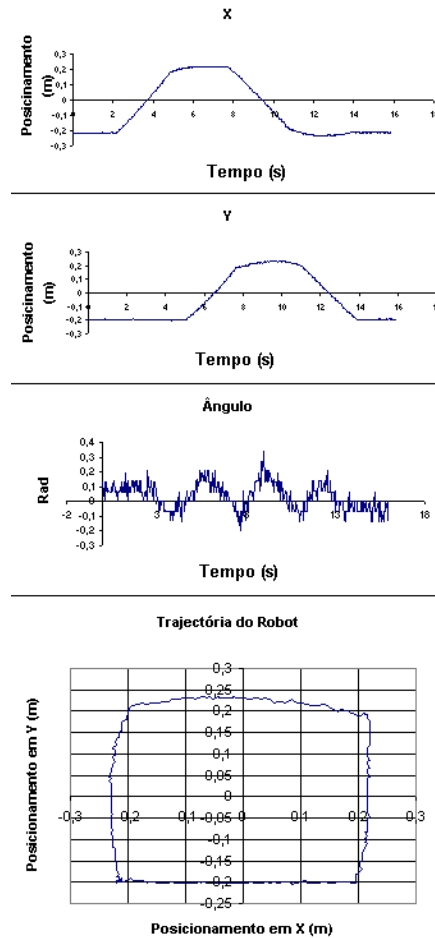


Figura 11. Corrida do *robot*

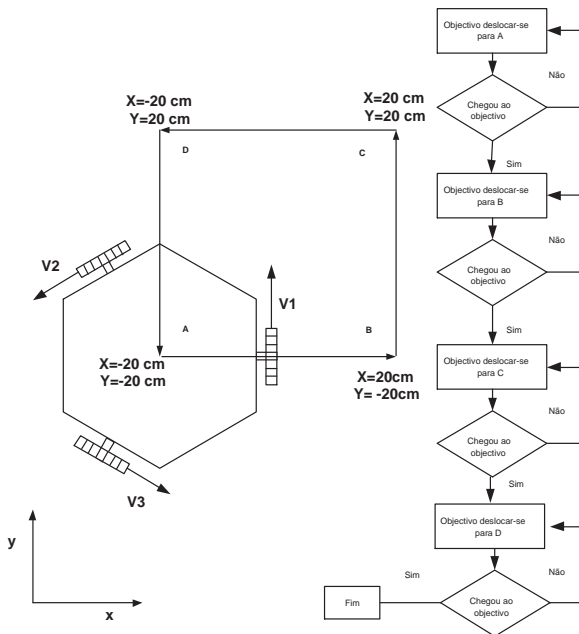


Figura 10. Fluxograma da corrida efectuada

### V. CONCLUSÕES

A ferramenta desenvolvida é versátil para a sua utilização em várias etapas de aprendizagem no domínio da robótica móvel. Pode ser utilizada para validação de controladores e tácticas (monitorizando em tempo real jogos segundo as regras do Robocup Junior), permite a validação de sistemas de localização e navegação para ambientes estruturados (monitorizando o desempenho do sistema de localização e navegação de um *robot*) e é também utilizado em actividades de investigação (estudo de controlo e técnicas de fusão sensorial).

Em robótica móvel os requisitos de tempo real são muito apertados, levando a que os algoritmos de localização sejam o mais possível optimizados, principalmente no que diz respeito ao processamento de imagem. Um algoritmo que acrescente muito atraso no processamento, pode ter consequências graves, principalmente no que diz respeito ao tempo de reacção do *robot*. Para esta aplicação foram atingidos os requisitos de tempo-real pretendidos, ou seja, 25 frames por segundo (máximo permitido pelo sistema PAL).

## REFERÊNCIAS

- [1] L. Almeida, J. Azevedo, C. Cardeira, P. Costa, P. Fonseca, P. Lima, F. Ribeiro and V. Santos, *Mobile robot competitions: Fostering advances in research, development and education in robotics*, Proceedings of the 4th Portuguese conference in Automatic Control, 2000.
- [2] "Robocup" <http://www.robocup.org/>, 2007.
- [3] H. Lund and L. Pagliarini, *Robocup jr. with lego mindstorms*, in Proceedings of the 2000 IEEE International Conference on Robotics and Automation, San Francisco, CA, April 2000, IEEE.
- [4] P. Gawthrop and E. McGookin, *Using Lego in Control Education*, Proceedings of the 7th Ifac Symposium on Advances in control education, 2006.
- [5] J. Gonçalves, J. Lima and P. Costa, *Rapid prototyping of mobile robots extending Lego Mindstorms platform*, Proceedings of the 7th Ifac Symposium on Advances in control education, 2006.
- [6] L. Almeida, J. Azevedo and B. Cunha, *Sete anos de Concurso Micro-Rato na Universidade de Aveiro*, Revista Robótica, Nrº 48, Publindústria, 2002.
- [7] A. Sousa, *Arquitecturas de Sistemas Robóticos e Localização em Tempo Real Através de Visão*, Tese de Doutoramento, Faculdade de Engenharia da Universidade do Porto, 2003.
- [8] F. Ribeiro, I. Moutinho, P. Silva, C. Fraga and N. Pereira, *Controlling Omni-Directional Wheels of a Robocup MSL Autonomous Mobile robot*, Proceedings of the Scientific Meeting of the Robotics Portuguese Open, 2004.
- [9] P. Costa, *Localização em tempo real de múltiplos robots num ambiente dinâmico*, Tese de Doutoramento, Faculdade de Engenharia da Universidade do Porto, 1999.
- [10] S. Thrun, W. Burgard and D. Fox, *Probabilistic Robotics*, MIT Press, 2005.
- [11] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, S. Thrun, *Principles of Robot Motion : Theory, Algorithms, and Implementations*, MIT Press, 2005.
- [12] M. I. Ribeiro, *Gaussian Probability Density Functions: Properties and Error Characterization*, Technical Report, IST, 2004.
- [13] R. Negenborn, *Robot Localization and Kalman Filters - On finding your position in a noisy world*, Master Thesis, Utrecht University, 2003.
- [14] J. Gonçalves, P. Costa and A. Moreira, *Controlo e estimação do posicionamento absoluto de um robot omnidireccional de três rodas*, Revista Robótica, Nrº 60, Publindústria, pp 18-24, 2005.



**José Gonçalves**, Bragança, Portugal, 1976, Licenciado em Engenharia electrotécnica e de computadores ramo de Automação, produção e electrónica industrial pela Faculdade de Engenharia da Universidade do Porto (FEUP) em 2000, obteve o Mestrado em Engenharia electrotécnica e de computadores pela FEUP em 2005 no ramo de Informática e automação, tendo realizado uma tese de dissertação intitulada: "Controlo de robots omnidireccionais". Actualmente é aluno de Doutoramento na FEUP na área de Controlo e Robótica e lecciona no Instituto Politécnico

de Bragança unidades curriculares das áreas de automação, robótica, sistemas embebidos, electrónica e instrumentação. Proferiu palestras e publicou artigos relacionados com experiências didácticas e metodologias de ensino nas áreas de robótica móvel, robótica industrial e automação. É investigador no Instituto de Sistemas e Robótica do Porto (Portugal), sendo os seus interesses na investigação localização, navegação, controlo e prototipagem de robots móveis, tendo publicado artigos nestas áreas.



**Pedro Pinheiro**, Massarelos, Porto, 1982, Obteve o Bacharel em Engenharia Informática no Instituto Politécnico de Bragança (IPB) em 2004, obteve a Licenciatura em Engenharia Informática no IPB em 2006, realizou um projecto de final de curso subordinado ao tema "Sistema de Localização em tempo-real para robótica industrial e móvel, actualmente exerce funções de programador no Portal AEIQU.PT.



**José Lima**, Porto, Portugal, 1978, Licenciado em Engenharia electrotécnica e de computadores ramo de Automação, produção e electrónica industrial pela Faculdade de Engenharia da Universidade do Porto (FEUP) em 2001, obteve o Mestrado em Engenharia electrotécnica e de computadores pela FEUP em 2004 no ramo de Informática e automação, tendo realizado uma tese de dissertação intitulada: "Sistemas de aquisição de imagem via Ethernet e processamento em tempo real". Actualmente é aluno de Doutoramento na FEUP na área de Controlo e

Robótica e lecciona no Instituto Politécnico de Bragança unidades curriculares das áreas de sistemas embebidos, electrónica, instrumentação e processamento de imagem. É investigador no Instituto de Sistemas e Robótica do Porto (Portugal), sendo os seus interesses na investigação visão industrial, navegação robots móveis, sistemas embebidos, controlo e automação, tendo publicado artigos nestas áreas.



**Paulo Costa**, Porto, Portugal, 1968, Licenciado em Engenharia electrotécnica e de computadores ramo de Informática e Sistemas pela Faculdade de Engenharia da Universidade do Porto (FEUP) em 1991, obteve o Mestrado em Engenharia Electrotécnica e de computadores pela FEUP em 1995 no ramo de Sistemas, tendo realizado uma tese de dissertação intitulada: "Identificação, modelização e controlo de um robot móvel autónomo". Obteve o Doutoramento em Engenharia Electrotécnica e de computadores pela FEUP em 2000, tendo realizado uma tese de

dissertação intitulada: "Localização em tempo real de múltiplos robots em um ambiente dinâmico". Lecciona na Faculdade de Engenharia da Universidade do Porto unidades curriculares das áreas de robótica, controlo, sistemas embebidos, automação e visão industrial. É investigador no Instituto de Sistemas e Robótica do Porto (Portugal), sendo os seus interesses na investigação controlo em tempo real, robots autónomos e sistemas de visão, tendo publicado artigos nestas áreas. Participa activamente nas competições do Robocup nas ligas SSL e MSL, sendo Team Leader da equipa 5DPO. No ano de 2006 obteve o segundo lugar no Robocup que decorreu em Bremen (Alemanha), primeiro lugar no RoboLudens que decorreu em Heindoven (Holanda) e primeiro lugar no Robotics Portuguese open que decorreu em Guimarães (Portugal). Pertence ao comité executivo do Robocup da liga SSL. É orientador de várias teses de mestrado e doutoramento na Faculdade de Engenharia da Universidade do Porto nas áreas de visão industrial e robótica móvel. Participou em cooperação com a indústria em projectos de controlo de qualidade de produtos utilizando visão industrial.

