

# A Versão 2006 da Equipe POTI de Futebol de Robôs

Auciomar C. T. Cerqueira<sup>1</sup>, Filipe C. A. Lins<sup>1</sup>,  
Adelardo A. D. Medeiros<sup>1</sup>, Pablo J. Alsina<sup>1</sup>

<sup>1</sup>Universidade Federal do Rio Grande do Norte (UFRN)  
UFRN-CT-DCA – Campus Universitário  
59078-900 – Natal – RN

[goi,campos,adelardo,pablo]@dca.ufrn.br

**Abstract.** *This paper describes the electro-mechanical characteristics and the algorithms for perception, game strategy and control adopted in the POTI team of robot soccer.*

**Resumo.** *Este artigo descreve as características eletro-mecânicas e os algoritmos de percepção, estratégia de jogo e controle adotados na equipe POTI de futebol de robôs.*

## 1. Introdução

No futebol de robôs, uma frota de mini-robôs (três robôs no nosso caso) deve disputar uma partida de futebol de forma autônoma, ou seja, sem interferência humana. A idéia geral do jogo é mostrada na figura 1. O sistema de visão (câmera e localização) é responsável por identificar os jogadores e a bola em campo e informar ao módulo da estratégia a localização de cada um deles. A estratégia, considerada a parte inteligente de todo o sistema, é responsável por decidir para qual posição o robô deve ir. De posse das posições presente e futura, o controle (módulo de baixo nível) calcula as tensões que devem ser aplicadas a cada motor para se chegar na posição desejada. As tensões são enviadas via rádio para os jogadores do time. Este ciclo de localizar, decidir posição futura, calcular e enviar tensões, é repetido continuamente até o fim ou interrupção da partida.

No restante deste artigo serão apresentados as características eletro-mecânicas dos robôs (seção 2) e os módulos de visão (seção 3), estratégia (seção 4) e controle (seção 5) adotados pela equipe POTI de futebol de robôs, desenvolvida pelo Departamento de Engenharia de Computação e Automação (DCA) da UFRN.

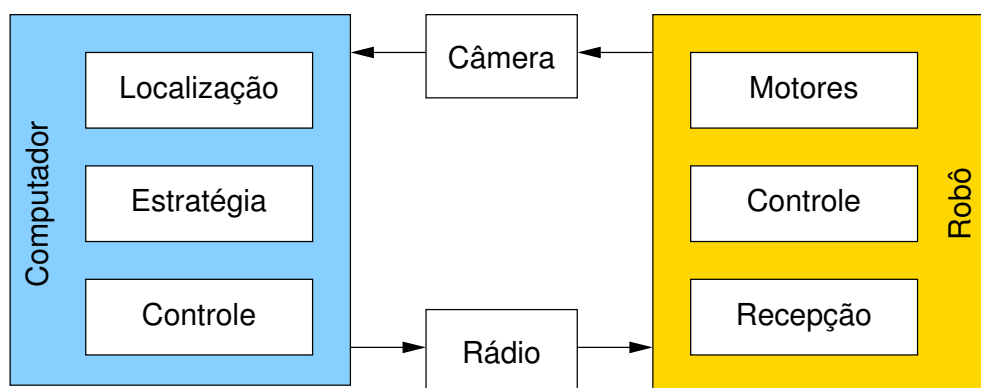


Figura 1. Arquitetura da equipe POTI de futebol de robôs

## 2. Características eletro-mecânicas

No futebol de robôs, classe *small size*, as dimensões de cada robô são padronizadas. Cada robô tem um formato de um cubo com arestas de 7,5cm. Com relação a outras características, como tipo de motor, peso, tamanho das rodas e material utilizado na estrutura, são adotadas as escolhas de cada equipe.

No caso da equipe POTI, utilizavam-se anteriormente [Yamamoto et al. 2004, Cerqueira et al. 2005] estruturas de acrílico com motores DC de 9V diretamente acoplados às rodas. A nova versão do robô apresenta uma estrutura em camadas, tendo na parte superior a placa eletrônica, ao centro as pilhas e na base os motores com as caixas de redução e *encoders* (figura 2). Os motores foram trocados por motores DC de 6V, permitindo uma diminuição do número de pilhas para cada robô e tornando-os mais leves. Foram acrescentadas caixas de redução entre os motores e as rodas para aumentar o torque em cada roda, o que melhora o desempenho do robô em baixas velocidades e conseqüentemente permite um controle mais preciso do posicionamento do robô.

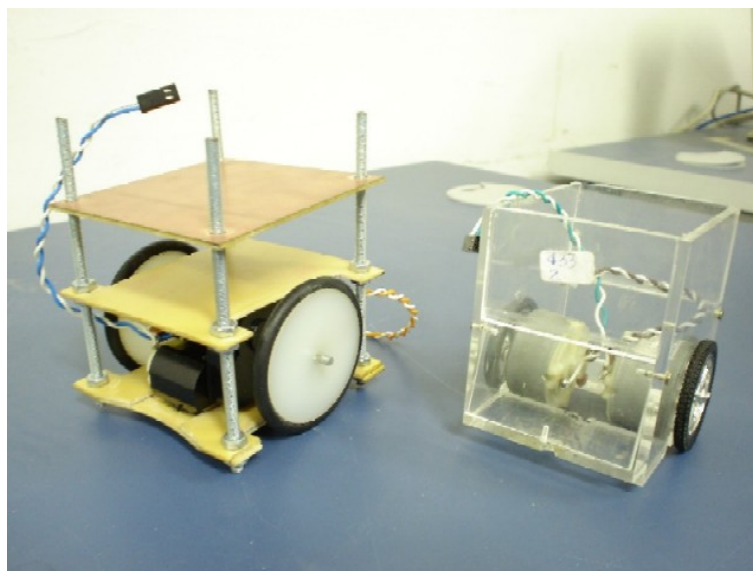


Figura 2. Duas estruturas dos robôs

A eletrônica embarcada do robô é composta basicamente por um receptor de rádio, um microcontrolador PIC 16F876, uma ponte H L293 para acionar os motores nos dois sentidos e dois *encoders*. O receptor FM recebe serialmente as informações vindas do computador, que representam o sentido e a velocidade desejados para cada motor. O microcontrolador calcula (seção 5.1) e envia aos motores os valores das velocidades em PWM (*Pulse Width Modulation*), o que permite, através da ponte H, o acionamento dos motores das rodas nos dois sentidos.

A inclusão dos *encoders* é um outro aprimoramento dos robôs atuais em relação aos anteriores. Foram utilizados *encoders* retirados de *mouses* de computadores. Para melhorar a precisão da leitura, os mesmos foram acoplados ao eixo de cada motor, pois a presença de caixas de redução diminui a precisão dos *encoders* nos eixos das rodas. Os *encoders* tornam possível a existência de um controle local da velocidade dos motores. A utilização do controle embarcado permite corrigir assimetrias dos motores e tornar mais simples e preciso o controle de posição do robô, que é feito no computador.



**Figura 3. Iluminação**



**Figura 4. Cores do jogo**

### **3. Sistema de Visão**

O sistema de visão é fisicamente composto de uma estrutura de iluminação e uma câmera colocada acima do campo de jogo (figura 3), a qual é conectada a uma placa de aquisição de imagens. No caso da equipe POTI, foi utilizada a placa PCI 1411 da National Instruments, a qual fornece uma imagem a cada 33 ms.

O objetivo deste sistema é capturar imagens do jogo e, através de técnicas de processamento digital de imagens, informar ao módulo da estratégia as localizações de cada robô (tanto os do próprio time quanto os do adversário) e da bola.

A distinção entre os jogadores e a bola em uma partida de futebol de robôs é feita com o auxílio de cores e formas. De acordo com as regras, a superfície do campo é preta, as linhas e o círculo central brancos, a bola da cor laranja e cada jogador deve possuir obrigatoriamente um rótulo azul ou amarelo identificando seu time. A equipe, a seu critério, pode decidir acrescentar um rótulo extra a cada robô, com uma cor alternativa, para facilitar o cálculo da orientação dos robôs e/ou diferenciar os jogadores da mesma equipe (Figura 4).

A tarefa do sistema de visão, portanto, pode ser dividida em três fases:

1. Na primeira fase (seção 3.1), o sistema deve localizar as regiões coloridas que existem na imagem e calcular a posição central e a etiqueta de cor para cada uma.
2. Uma vez definidas as posições e as etiquetas de cores das regiões, a segunda fase (seção 3.2) envolve a identificação dos robôs e da bola.
3. Na terceira fase (seção 3.3) é feita a individualização de cada um dos três robôs da equipe.

O tempo de processamento da imagem é importante para o desempenho do time, pois se trata de uma aplicação em tempo real. Na equipe POTI, uma imagem deve ser tratada em menos de 33 ms. Além disso, a técnica adotada deve ser robusta a variações na condição de iluminação e ser capaz de reconhecer robôs de times adversários que usam rótulos de diferentes cores e formatos. As técnicas de processamento digital de imagens utilizadas pela equipe POTI na resolução deste problema são descritas a seguir.

#### **3.1. Detecção e Etiquetagem de Cor de Regiões**

Em anos anteriores [Yamamoto et al. 2004, Cerqueira et al. 2005] a equipe POTI utilizava sistemas de percepção baseados em uma primeira etapa de segmentação da imagem e uma

segunda etapa de análise das regiões encontradas após a segmentação. A segmentação era feita utilizando-se principalmente a luminância do *pixel*: *pixels* escuros pertenciam ao campo e *pixels* claros pertenciam a objetos. Estas abordagens tinham problemas de robustez quanto a variações na iluminação durante o jogo, fazendo com que regiões fossem fundidas ou fracionadas.

Nesta nova versão, foram incorporadas mudanças significativas nas técnicas de processamento de imagens do sistema de visão. São utilizadas faixas de RGB e de LSC (luminância, saturação e crominância) para classificar as cores do jogo. Essas faixas são determinadas na calibração do sistema de visão, anterior à partida. São escolhidos para cada cor limites das faixas RGB e LSC, de modo que um *pixel* não seja enquadrado em duas cores ao mesmo tempo. Não é obrigatório utilizar todos os parâmetros de RGB e LSC para qualificar uma cor. Por exemplo, um *pixel* pode receber a etiqueta de cor azul apenas baseado no fato de a sua componente C estar dentro da faixa correspondente. A escolha de quais componentes são relevantes para etiquetar cada cor também é feita na calibração, a partir da análise de imagens do campo com os objetos.

Existem diversas formas para cálculo das componentes de luminância, saturação e crominância de um *pixel*. Este trabalho propõe uma nova abordagem para se fazer este cálculo, que se apresentou particularmente robusta a variações de iluminação no cálculo da crominância. Nesta proposta, as componentes LSC apresentam seus valores normalizados entre 0 e 1 e são calculadas da seguinte forma:

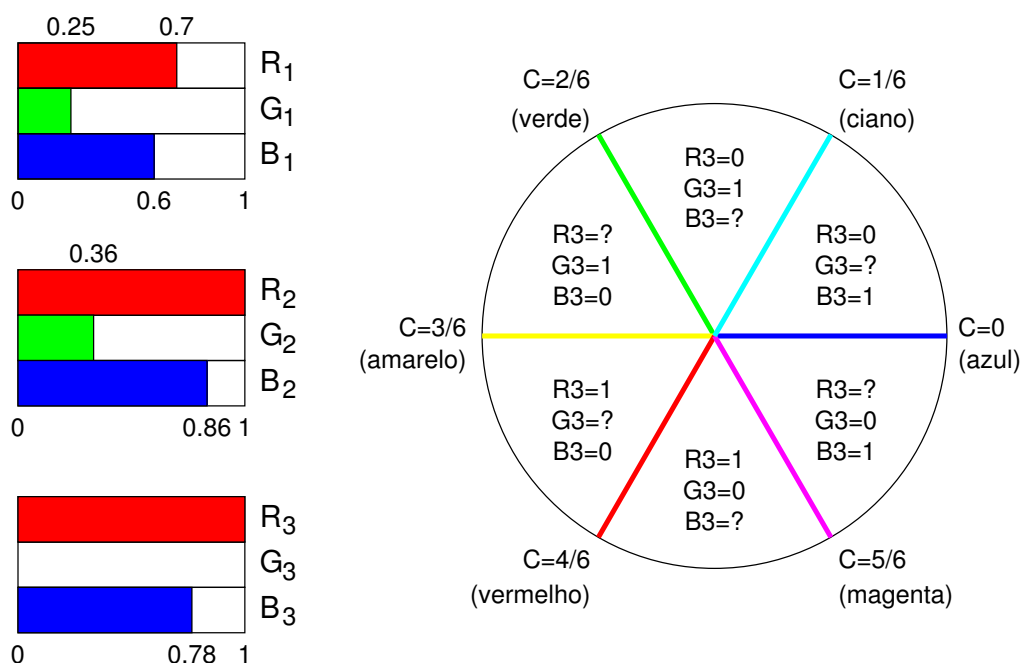
- Para determinar a luminância, as componentes RGB são normalizadas ( $R_1, G_1, B_1$ ); a luminância será o maior destes valores (equação 1).
- Para determinar a saturação, calculam-se as componentes ( $R_2, G_2, B_2$ ) de um *pixel* com a mesma cor mas normalizados pela luminância. Em seguida, a saturação será o maior valor do complemento destas componentes (equação 2, onde  $\bar{R}_2 = 1 - R_2$ ,  $\bar{G}_2 = 1 - G_2$  e  $\bar{B}_2 = 1 - B_2$ ).
- Para determinar a crominância, calculam-se as componentes ( $R_3, G_3, B_3$ ) de um *pixel* com a mesma cor mas com normalizados pela luminância e pela saturação, conforme a equação 3. A maior destas componentes sempre vale 1, a menor sempre vale 0 e a terceira apresenta valor entre 0 e 1. A crominância é calculada mapeando-se estas componentes auxiliares no círculo de cores da figura 5.

$$R_1 = \frac{R}{255} \quad G_1 = \frac{G}{255} \quad B_1 = \frac{B}{255} \quad L = \max(R_1, G_1, B_1) \quad (1)$$

$$R_2 = \frac{R_1}{L} \quad G_2 = \frac{G_1}{L} \quad B_2 = \frac{B_1}{L} \quad S = \max(\bar{R}_2, \bar{G}_2, \bar{B}_2) \quad (2)$$

$$R_3 = 1 - \frac{\bar{R}_2}{S} \quad G_3 = 1 - \frac{\bar{G}_2}{S} \quad B_3 = 1 - \frac{\bar{B}_2}{S} \quad C = \text{Círculo de cores} \quad (3)$$

A figura 5 exemplifica as componentes RGB normalizadas. Neste exemplo, as componentes RGB iniciais apresentam valores 180, 64 e 155, respectivamente. A luminância será igual a 0.7 ( $= R_1 = 180/255$ ) e a saturação apresentará valor 0.64 ( $= 1 - G_2$ ). Como  $R_3 = 1$  e  $G_3 = 0$ , esta cor se situa na faixa entre vermelho ( $C = 4/6 = 0.67$  para  $B_3 = 0$ ) e magenta ( $C = 5/6 = 0.83$  para  $B_3 = 1$ ). Como  $B_3$  tem valor 0.78, a crominância do *pixel* é 0.79 ( $= 4/6 + B_3 * (1/6)$ ). O resultado correspondente a uma cor próxima ao magenta.



**Figura 5. Normalização das componentes RGB**

Cada nova imagem é tratada fazendo-se a análise de seus *pixels* a cada  $N$  linhas e a cada  $N$  colunas. O valor de  $N$  é determinado levando-se em conta a dimensão dos objetos, de forma que se tenha a garantia de que ao menos um *pixel* de cada objeto será incluído na varredura. Os *pixels* escolhidos são analisados pelo algoritmo a seguir:

1. Compara-se o *pixel* atual com as faixas RGB e LSC das cores preto e branco: caso este se enquadre em uma destas faixas, será considerado como pertencente ao campo e a varredura prossegue.
2. Caso contrário, o *pixel* é candidato a pertencer a um objeto. Então ele é comparado com as faixas das cores possíveis. Se não corresponder a nenhuma das cores que se espera encontrar, a varredura prossegue.
3. Definida a etiqueta de cor do *pixel*, usa-se um algoritmo de preenchimento de região (no caso, o algoritmo *Seed Fill* [Gonzalez and Woodes 2000] clássico) para encontrar todos os *pixels* vizinhos com a mesma etiqueta de cor; caso seja encontrado um número mínimo de *pixels*, uma nova região terá sido encontrada. O centro da região será a média das coordenadas  $(x, y)$  de todos os seus *pixels*.

### 3.2. Detecção dos objetos

Uma vez definidas as posições e etiquetas de cores das regiões, pode-se determinar a posição dos robôs e da bola (segunda fase do sistema de visão). A bola é uma região de etiqueta laranja e com as dimensões (ou seja, número de *pixels*) apropriadas. A identificação dos robôs depende da rotulação utilizada pela equipe.

Anteriormente a equipe POTI utilizava uma cor diferente para o rótulo auxiliar de cada robô [Yamamoto et al. 2004, Cerqueira et al. 2005]. Esta estratégia foi modificada, adotando-se um rótulo auxiliar rosa para todos os robôs da equipe. Desta forma, um robô da equipe será reconhecido pela existência de duas regiões bem próximas, sendo uma azul ou amarela e a outra rosa. Os robôs adversários são reconhecidos apenas pela cor do rótulo principal (azul ou amarelo).

### 3.3. Etiquetação dos robôs

Como os rótulos auxiliares dos robôs têm a mesma cor, deve-se utilizar uma regra heurística para diferenciá-los. A individualização dos jogadores do time é feita através da posição no passo anterior: o robô mais perto da posição ocupada pelo robô 0 no passo anterior será o robô 0 no passo atual e assim por diante. Regras especiais lidam com a situação inicial, onde não se tem informação anterior, e com situações onde os robôs estão muito próximos, havendo possibilidade de ambigüidade.

## 4. Estratégia

O módulo da estratégia é considerado o “cérebro” de um time de futebol de robôs. A sua função é de, dadas as localizações de cada objeto em campo, decidir para onde os jogadores do time devem ir, de forma que estes apresentem um comportamento cooperativo inteligente.

A estratégia está organizada em camadas [Yamamoto 2005]. Existem três níveis de decisão: os papéis de cada robô estão no nível mais alto, no segundo nível estão os estados e no nível mais baixo os cálculos das posições desejadas (referências) de cada robô. A figura 6 mostra a hierarquia das camadas da estratégia.

O nível 1 usa as informações de posição dos objetos no campo para delegar a cada robô um dos três papéis possíveis: goleiro, atacante ou zagueiro (seção 4.1). Na segunda camada o processo é dividido e cada robô é tratado individualmente. A partir dos dados do jogo e do papel imposto ao jogador, será decidido qual a melhor ação a se tomar no momento (seção 4.2). Finalmente, no terceiro nível é calculada a posição que cada robô deve buscar para a próxima iteração (seção 4.3).

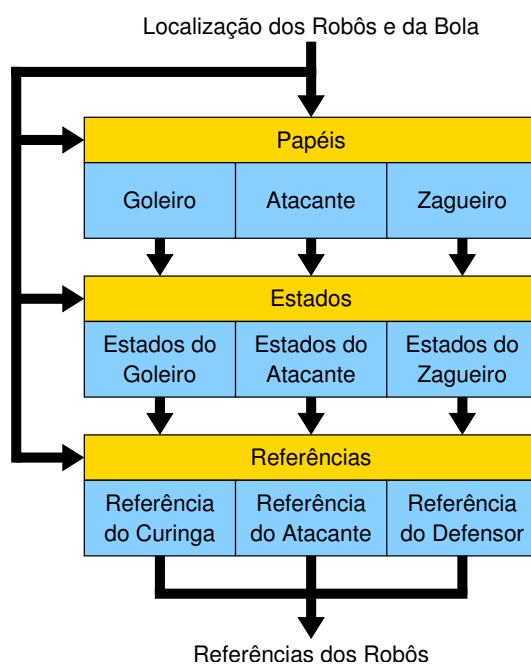


Figura 6. Hierarquia da estratégia

#### 4.1. Alocação Dinâmica de Papéis

Foram definidos três papéis para a equipe POTI de futebol de robôs: goleiro, atacante e zagueiro, tendo cada um dos papéis um certo número de estados associados. No decorrer do jogo a estratégia irá indicar o papel mais adequado para cada robô na situação presente.

O papel de goleiro será sempre atribuído a um mesmo jogador escolhido antes do início da partida, de forma que a alocação dinâmica acontecerá apenas com os papéis de atacante e zagueiro. Para definir os papéis dos jogadores restantes, a primeira preocupação é saber a localização da bola:

- Caso a bola esteja no campo de defesa, o jogador atrás da linha da bola será o atacante. Se ambos estiverem ou atrás ou a frente desta linha, o jogador mais perto da bola será o atacante e o zagueiro será o robô restante.
- Na situação da bola estar no campo de ataque, o jogador mais perto da bola será o atacante e o outro robô será o zagueiro

#### 4.2. Estados dos Jogadores

O goleiro pode estar nos estados DEFESA ou ESPERA. No estado ESPERA, fica no centro do gol. No estado DEFESA, fica posicionado de tal forma a interceptar a trajetória da bola para o gol. O estado ESPERA é adotado quando a bola está no campo de ataque; caso contrário, impõe-se o estado DEFESA.

O atacante é o jogador que mais interage com a bola. Seus possíveis estados são: CHUTAR, ALINHAR e CONTORNAR. Caso a bola esteja no ataque e o atacante à frente da linha da bola, ele irá se posicionar atrás desta linha de modo a CONTORNAR a bola. Se o atacante estiver atrás da bola e sobre a linha formada pela bola e o centro do gol adversário, ele irá assumir o papel de CHUTAR. Caso não esteja sobre a linha, tentará se ALINHAR com esta linha.

O zagueiro apresenta dois estados: ESPERA e ZAGUEIRO. O estado ESPERA é atribuído quando a bola está no campo de ataque. O robô se posiciona no campo de defesa, para esperar uma eventual sobra de bola. O estado ZAGUEIRO posiciona o jogador na linha formada pela posição da bola e centro do gol do time, de modo a interceptar o avanço da bola. Este estado é atribuído quando a bola está no campo de defesa.

#### 4.3. Geração de Referências

A geração de referências envolve a transformação das ações simbólicas decididas para cada robô em informações numéricas de posições ( $x_{ref}$ ,  $y_{ref}$ ). A maioria das referências pode ser calculada a partir de relações trigonométricas simples. Por exemplo, para a ação ALINHAR do atacante, traça-se uma reta do centro do gol adversário até a bola e calcula-se para o robô uma referência sobre esta reta, localizada a uma distância fixa atrás da bola (no caso, 15cm).

Um outro aspecto importante da geração de referências é o desvio de obstáculos. As referências são inicialmente calculadas sem levar em conta a existência dos outros robôs em campo. Em seguida, verifica-se para cada robô se a reta que vai da sua posição atual até a referência calculada colide com algum outro robô. Caso sim, o vetor que vai do robô até a referência será girado de forma a que a trajetória desvie do obstáculo. O ângulo de giro é proporcional à distância do robô ao obstáculo, sendo nulo a partir de uma determinada distância (20cm, no caso). Este raio em torno de cada robô define o que se denominou *zona de influência* do obstáculo [Yamamoto 2005].

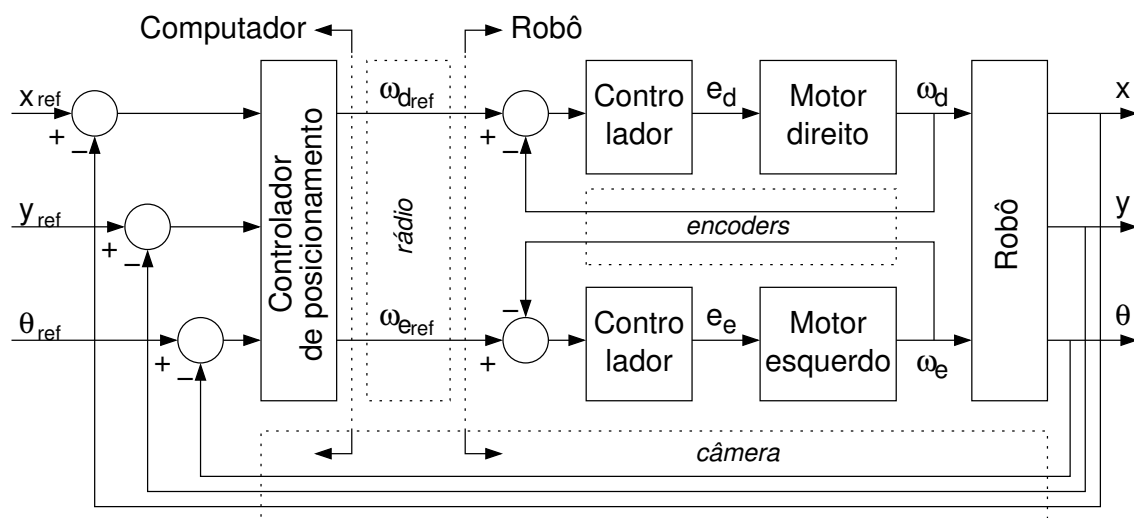


Figura 7. Malhas de controle do robô

## 5. Controle

A equipe POTI utiliza duas malhas de controle, conforme a figura 7: uma interna para o controle de velocidade dos motores e outra externa para controlar a posição do robô.

### 5.1. Controle de Velocidade

O controle de velocidade é realizado pelo microcontrolador embarcado no robô. As variáveis a serem controladas ( $\omega_d$ ,  $\omega_e$ ) são medidas através da contagem do número de pulsos recebidos dos *encoders* durante um período de amostragem. As referências de velocidades ( $\omega_{dref}$ ,  $\omega_{eref}$ ) são calculadas pelo controle de posição e enviadas do computador para o robô via rádio.

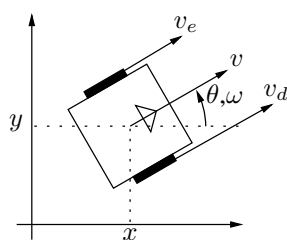
A cada período de amostragem, as últimas referências recebidas são comparadas com as velocidades atuais. As tensões aplicadas nos motores ( $e_d$ ,  $e_e$ ) são então calculadas através de técnicas clássicas de controle de velocidade de motores de corrente contínua. No nosso caso é utilizado um controlador proporcional-integral (PI) [Ogata 2003].

### 5.2. Controle de Posição do Robô

O controle de posição é feito no computador. As variáveis controladas são a posição atual do robô ( $x$ ,  $y$ ,  $\theta$ ), calculadas pelo módulo de percepção a partir da análise das imagens da câmera de vídeo (seção 3). A estratégia define as referências (seção 5), que são a posição para a qual o robô deve se deslocar ( $x_{ref}$ ,  $y_{ref}$ ). A saída deste controlador é as velocidades angulares desejadas ( $\omega_{dref}$ ,  $\omega_{eref}$ ) para os dois motores do robô.

Consideremos o modelo esquemático do robô na figura 8. No espaço cartesiano ( $x$ ,  $y$ ) definiremos a posição do robô como a posição do seu centro e sua orientação  $\theta$  como sendo o ângulo entre o vetor da orientação do robô e o eixo  $x$ . Denominam-se  $v$  e  $\omega$ , respectivamente, as velocidades linear e a angular do robô, medidas no seu centro.

Se as rodas têm raio  $R$ ,  $b$  é a largura do robô (distância entre rodas),  $\omega_d$  e  $\omega_e$  são as velocidades angulares das rodas e  $v_d$  e  $v_e$  são as velocidades lineares de um ponto na



**Figura 8. Definições de velocidades do robô**

circunferência das rodas, sabe-se que:

$$v_d = R\omega_d = v + \frac{b}{2}\omega \quad v_e = R\omega_e = v - \frac{b}{2}\omega \quad (4)$$

Com isso, ao invés do controlador de posição calcular diretamente as velocidades desejadas para os motores ( $\omega_{dref}$ ,  $\omega_{eref}$ ), ele pode calcular inicialmente as velocidades linear e angular desejadas ( $v_{ref}$ ,  $\omega_{ref}$ ) e em seguida fazer a conversão usando a equação 4.

A figura 9 ilustra o problema do posicionamento do robô, onde  $\Delta l$  representa a distância entre o robô e a posição de referência ( $x_{ref}$ ,  $y_{ref}$ ) desejada no espaço cartesiano. O problema do controle de posição do robô será resolvido se  $\Delta l \rightarrow 0$ .

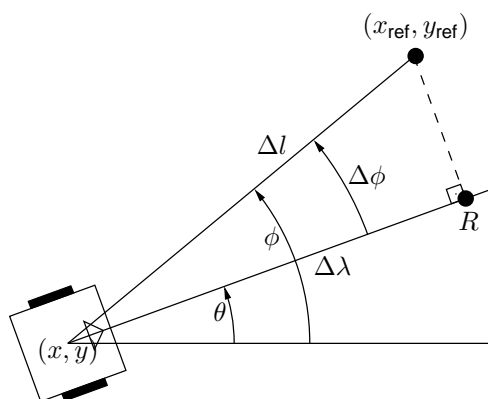
Mostra-se [Vieira et al. 2004] que este problema pode ser resolvido por meio de dois controladores lineares simples:

- Um controlador angular cujo objetivo é controlar a orientação do robô de forma a que ele aponte para a posição de referência desejada ( $x_{ref}$ ,  $y_{ref}$ ). A saída deste controlador é a velocidade angular desejada,  $\omega_{ref}$ . A entrada é o erro angular  $\Delta\phi$ , que é a diferença entre a orientação desejada e a orientação atual e é dado por:

$$\Delta\phi = \phi - \theta = \tan^{-1} \left( \frac{y_{ref} - y}{x_{ref} - x} \right) - \theta \quad (5)$$

- Um controlador linear cujo objetivo é posicionar o robô ao longo da reta que corresponde à sua orientação atual no ponto mais próximo da posição de referência desejada (ponto  $R$  na figura 9). A saída deste controlador é a velocidade linear desejada,  $v_{ref}$ . A entrada é o erro linear  $\Delta\lambda$ , dado por:

$$\Delta\lambda = \Delta l \cdot \cos(\Delta\phi) = \sqrt{(x_{ref} - x)^2 + (y_{ref} - y)^2} \cdot \cos(\Delta\phi) \quad (6)$$



**Figura 9. Controle de posição do robô**

Para o projeto tanto do controlador linear quanto do angular, o sistema a ser controlado é modelado por um integrador simples,  $G(s) = 1/s$ , pois o deslocamento angular  $\theta$  é a integral da velocidade angular  $\omega$ , o mesmo acontecendo com o deslocamento linear  $\lambda$  e a velocidade linear  $v$ .

No caso da equipe POTI, adotou-se um controlador angular proporcional (P) e um controlador linear proporcional-derivativo (PD) [Ogata 2003], dimensionados de forma a garantir um sobressinal de 5% e um tempo de estabilização de 1s para o comportamento angular e de 2s para o linear.

## 6. Conclusões

As modificações e aprimoramentos introduzidos na versão 2006 da equipe POTI buscam contornar problemas e limitações detectados em anos anteriores. Algumas das novas propostas apresentadas neste artigo ainda não foram completamente testadas em condições reais, o que pode fazer com que ocorram alterações durante a fase de testes.

Dentro dos projetos futuros está a tentativa de identificação automática dos limites óticos das faixas RGB e LSC que melhor separam as diversas cores nas imagens (seção 3.1). Técnicas de inteligência artificial (redes neurais, lógica fuzzy) estão sendo testadas e, caso dêem bons resultados, serão incorporadas ao sistema de percepção.

## Referências

- Cerqueira, A. C. T., Lins, F. C. A., Pereira, J. P. P., Medeiros, A. A. D., and Alsina, P. J. (2005). O time POTI de futebol de robôs da UFRN. In *ENRI - Encontro Nacional de Robótica Inteligente*, São Luiz, MA. <http://www.dca.ufrn.br/~adelardo/>.
- Gonzalez, R. C. and Woodes, R. E. (2000). *Processamento de Imagens Digitais*. Edgard Blucher. ISBN: 8521202644.
- Ogata, K. (2003). *Engenharia de Controle Moderno*. Pearson Brasil, 4<sup>a</sup> edition. ISBN: 8587918230.
- Vieira, F. C., Medeiros, A. A. D., Alsina, P. J., and Araújo Jr., A. P. (2004). Position and orientation control of a two-wheeled differentially driven nonholonomic mobile robot. In *ICINCO – International Conference on Informatics in Control, Automation and Robotics*, Setúbal, Portugal. <http://www.dca.ufrn.br/~adelardo/>.
- Yamamoto, M. M. (2005). Planejamento cooperativo de tarefas em um ambiente de futebol de robôs. Dissertação de mestrado, UFRN, Natal, RN.
- Yamamoto, M. M., Araújo Jr., A. P., Alsina, P. J., and Medeiros, A. A. D. d. (2004). A equipe POTI de futebol de robôs. In *ENRI - Encontro Nacional de Robótica Inteligente*, Salvador, BA. <http://www.dca.ufrn.br/~adelardo/>.