



Equipe FEI-06 de Futebol de Robôs

Murilo Fernandes Martins, Ronaldo Kazuyoshi Satomi, José Angelo Gurzoni Junior, Valquiria Fenelon Pereira, Flavio Tonidandel e Reinaldo A.C. Bianchi

Centro Universitário da FEI – UniFEI
Av. Humberto de A. Castelo Branco, 3972
09850-901 - São Bernardo do Campo - SP - Brazil

{murilofm, flaviot, rbianchi} @ fei.edu.br; valfenelon@uol.com.br;
junior.gurzoni@hipernet.com.br; rksatomi@yahoo.com.br

Abstract. *This article presents the techniques of computational vision, robot control and game strategy used by FEI-06 robot soccer team. The computational vision is based on the Hough Transformation for circles, the robot control is a Proportional and derivative control with Kalman filter, and the strategy is based on game zones.*

Resumo. *Este artigo descreve as técnicas de visão computacional, controle de robôs e estratégia de jogo usada pela equipe de futebol de robôs FEI-06. O sistema de visão computacional é baseado em transformada de Hough para círculos, o controle é um PD (proporcional Derivativo) com filtro de Kalman e a estratégia é baseada em zonas de jogo.*

1. Introdução

Desde seu surgimento, o Futebol de Robôs tem sido uma plataforma motivante de pesquisa em inteligência artificial e desenvolvimento de robótica móvel autônoma. Isto porque, o Futebol de Robôs envolve diversas áreas de pesquisa, como Visão Computacional, eletrônica, mecânica, controle de sistemas e outras técnicas de Inteligência Artificial.

Um sistema de visão computacional para o futebol de robôs deve ser rápido e robusto, e é desejável que seja capaz de tolerar ruídos e variações de intensidade luminosa. Já o sistema de controle deve permitir que o robô siga trajetórias com o mínimo de erro possível. A estratégia deve permitir que os robôs atuem como um sistema integrado multi-agentes com o objetivo de fazer gol na equipe adversária. Todas essas considerações motivam cada vez mais as pesquisas na área de futebol de robôs.

Este trabalho descreve o sistema de visão, de controle e a estratégia da equipe de futebol de robôs da FEI (FEI06). O artigo descreve o uso de duas técnicas de segmentação de imagens que detectam os objetos – robôs móveis - por duas maneiras: detecta uma única forma (círculos) com várias cores e detecta várias formas de uma única cor. Além disso, descreve o sistema de controle Proporcional-derivativo (PD) implementado junto com o Filtro de Kalman (FK) e por fim, mostra como a estratégia de jogo está sendo desenvolvida.

2. Sistema de Visão Computacional implementado

No futebol de robôs, a categoria Mirosot da FIRA (2005) os robôs são detectados por meio de cores padronizadas que cada equipe que disputa uma partida tem no topo de cada robô. Segundo as regras, uma equipe deve ter a cor amarela e a outra equipe deve ter a cor azul no topo dos objetos para que seja reconhecida facilmente pelo adversário.

Entretanto, a regra não padroniza a forma geométrica que o robô deve possuir em cima com a cor especificada, desde que a área interna seja um sólido de $12,25 \text{ cm}^2$.

Identificar os objetos pela forma só é possível se a forma for pré-definida ou conhecida a priori. Identificar os objetos por cor pode ter influência da luminosidade e deixar o sistema de visão menos robusto na detecção dos objetos.

Assim, os robôs do time adversário, que podem ter qualquer forma mas uma única cor específica, serão detectados por cor. Já os robôs do time que estão usando o sistema de visão deste artigo, detecta os robôs por objetos circulares e independentes da cor, de forma mais robusta com relação a luminosidade.

Para detectar os robôs de seu próprio time, o sistema de visão proposto neste artigo utiliza-se da forma geométrica circular. Para isto, utiliza-se da transformada de Hough (Hough, 1959) e de outras técnicas de visão computacional como detecção de bordas (Canny 1986) e imagem em escala de cinza.

O modelo de etiqueta utilizado pela equipe FEI-06 possui dois círculos dispostos a 45° em relação a frente do robô (Figura 1), onde os círculos possuem o mesmo diâmetro da bola utilizada.

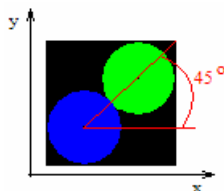


Figura 1. Modelo de etiqueta para diferenciação dos robôs

O sistema de visão computacional implementado possui basicamente sete etapas, que são elas: aquisição da imagem, subtração de fundo, conversão de cores para escala de cinza, aplicação do filtro de bordas, geração do espaço de Hough, determinação de pontos com alta probabilidade de serem centros de círculos na imagem e reconhecimento dos objetos (robôs e bola).

A aquisição de imagem é feita por uma placa de aquisição de imagens com capacidade de aquisição de 30 quadros por segundo, em imagens de 320×240 ou 640×480 pontos.

Para detectar os círculos, apenas as bordas da imagem são relevantes para a Transformada de Hough (TH). Para otimizar o tempo desse algoritmo utilizou-se um método simples de subtração de fundo que calcula a diferença entre a imagem capturada e uma imagem do fundo, sem os objetos móveis..

Após a subtração do fundo, a imagem é transformada de colorida para escala de cinza. Imagens coloridas demandam um maior tempo de processamento, pelo fato de possuírem três canais de informação de cor. Como a TH não necessita de informações de cor, para otimizar o tempo de processamento do sistema a imagem é convertida para escala de cinza. A imagem resultante contém apenas um canal de informação referente à variação de iluminação, a qual é suficiente para a determinação das bordas na imagem.

O presente trabalho utiliza uma técnica muito conhecida para detecção de bordas, o filtro Canny (Canny, 1986) resultando em uma imagem binária. A Figura 2 demonstra o resultado da detecção de bordas com o filtro Canny a partir de uma imagem resultante da subtração do fundo, convertida para escala de cinza.

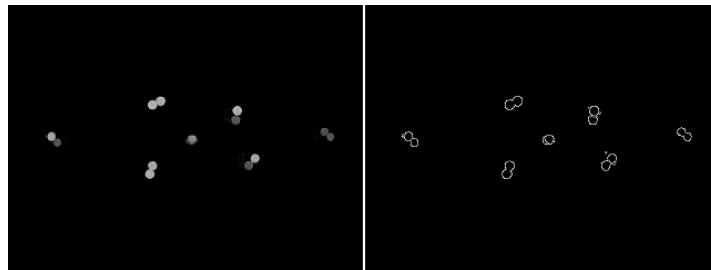


Figura 2. Resultado do filtro Canny

A partir da imagem binária resultante do filtro Canny pode-se gerar o espaço de Hough. Para cada ponto branco na imagem após o filtro Canny, o sistema traça um círculo de mesmo raio da bola e dos círculos em cima dos robôs.

Se o pixel branco for parte de um desses círculos na imagem, o círculo desenhado passará, obrigatoriamente, pelo centro do círculo da imagem. Esses círculos desenhados configuram o espaço de Hough. Cada ponto branco (borda) da imagem fará com que o espaço de Hough gere um círculo. Isso permite, portanto, que os pixels mais desenhados sejam os pixels centrais dos círculos da imagem. A figura 3 mostra um exemplo do espaço de Hough gerado.

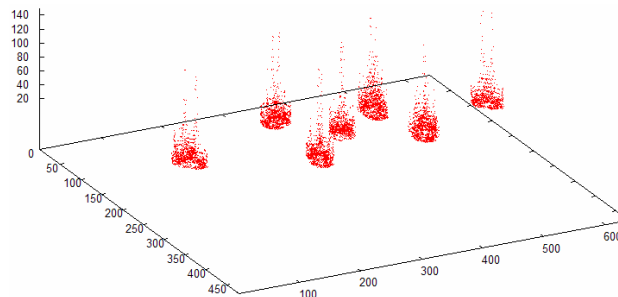


Figura 3. O espaço de Hough gerado

Os pontos de máximos locais na imagem 3 configuram círculos na imagem real. Para evitar que falsos máximos locais atrapalhem a definição de círculos, se considera a distância euclidiana dos pontos para detectar se há consideração de um círculo como sendo dois ou mais círculos no espaço de Hough.

Com os círculos detectados, passa-se a considerar a cor da região do círculo na imagem real para detectar se é um robô ou a bola.

Para se resolver problemas de círculos de dois robôs juntos, utilizam-se técnicas de satisfação de restrições que resolve o problema analisando as possibilidades de cada círculo pertencer a um robô. Vai-se eliminando as possibilidades uma a uma até se obter a classificação correta dos objetos.

Como a bola é representada por um círculo de cor laranja e essa cor não pode ser utilizada em nenhum outro objeto, todo e qualquer círculo cuja cor seja laranja é considerado uma bola.

Essas duas etapas, de determinação de centros de círculos e reconhecimento dos objetos, tornam a interpretação do espaço de Hough tolerante a ruídos, robusta a falsos máximos relativos e eficazes no reconhecimento de todos os objetos.

Para detectar os robôs da equipe adversária, temos a informação da cor a priori mas não temos informação da forma da etiqueta utilizada pelos robôs. Assim, utiliza-se a técnica de detecção de objetos por cor e de vários formatos.

Esta implementação segue a idéia de Bianchi e Reali-Costa (2000) de traçar segmentos internos aos objetos detectados de uma cor específica. Em outras palavras, o sistema percorre a imagem e, ao detectar um pixel da cor especificada, começa a traçar segmentos internos para os pixels da mesma cor na horizontal e na vertical. Para percorrer a imagem de maneira eficiente, o sistema salta pixels na horizontal e na vertical em uma imagem que o fundo foi retirado, igual dito anteriormente.

Os segmentos, por suas vezes, são traçados na horizontal, depois na metade do segmento horizontal é traçado o segmento vertical. O segmento é formado percorrendo a imagem na horizontal desde o ponto do pixel da cor específica encontrado até encontrar outro pixel que não é da cor especificada. O mesmo acontece para o segmento vertical. Esse processo de gerar segmento na horizontal e depois na vertical é chamado de processamento em cruz.

Executa-se o processamento em cruz, recursivamente, n vezes para cada pixel da cor determinada encontrado, onde o processamento m sempre começa no ponto central do segmento vertical do processamento $m-1$ anterior. A figura 4 mostra um exemplo de processamento de cruzes repetido.

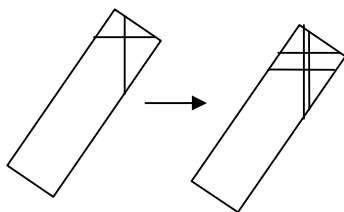


Figura 4. Processamento em cruzes: Segundo processamento em cruz a partir do centro do segmento vertical do primeiro processamento em cruz.

Esse processo em cruz é feito diversas vezes para cada objeto, visto que percorrer a imagem de saltos em saltos permite que esta varredura de pixel caia em vários lugares de um mesmo objeto. Cada processamento em cruz gera um centro provisório do objeto. Isso não é ruim, muito pelo contrário, vários centros provisórios

permitem encontrar o centro estimado mais provável do objeto com maior precisão. O centro do objeto será, portanto, estimado pela média dos pontos encontrados.

Cabe ressaltar que este método não garante encontrar o centro real de qualquer forma geométrica, ele acha apenas um centro estimado próximo ao centro real.

3. Sistema de Controle Implementado

O robô deve percorrer uma trajetória determinada pela estratégia, partindo de uma coordenada inicial (x_i, y_i) , até a coordenada final (x_f, y_f) . O sistema de controle atribuirá velocidades para cada roda, de modo que o robô possa realizar o percurso e alcançar o objetivo estipulado como mostra a figura 5.

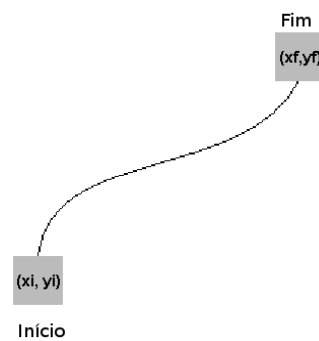


Figura 5 - Seguindo a trajetória.

A modelagem matemática é o modo de se criar uma equação que represente o sistema, podendo ser um modelo cinemático, dinâmico ou envolver ambos. Para tanto, pretende-se implementar um sistema de controle, que leve em consideração a cinemática de direção diferencial (*Differential Driver*), como mostra a figura 6 (Vieira et al, 2004).

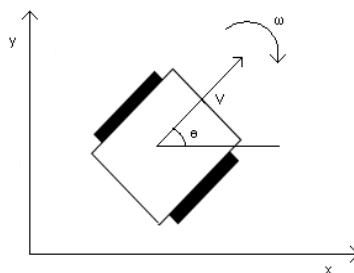


Figura 6 – Cinemática do Robô

Onde:

- ω é a velocidade angular
- v é a velocidade linear
- θ é a orientação
- (x, y) são as coordenadas da posição do robô.

Variando a velocidade das rodas independentemente, pode-se obter um movimento linear (equação. I) ou angular (equação II):

$$V = \frac{v_d + v_e}{2} \quad (\text{Equação I})$$

$$\omega = \frac{v_d - v_e}{2} \quad (\text{Equação II})$$

Caso o robô utilize as equações I e II, ele estará susceptível a erros na realização da trajetória, sendo necessário utilizar um sistema de controle servo-visual, ou controle de retroação (Ogata, 1997), que permita realizar a trajetória absorvendo os erros inerentes das limitações do robô e da trajetória, que consiste em comparar o valor da grandeza de saída com o valor da grandeza desejada, de modo a obter o valor do desvio (erro), e gerar um novo valor atenuado do mesmo.

Portanto, a equação III representa o modelamento do controle PD, para redução dos erros gerados pelo robô (Gupta et al, 2003):

$$v(t) = k_p \cdot e + k_d \cdot \left[\frac{de(t)}{dt} \right] \quad (\text{Equação III})$$

As equações IV e V, representam as equações de velocidade da roda direita e esquerda respectivamente com seus coeficientes para reduzir os erros, como mostrados por Gupta et al (2004):

$$vr(t) = (kd_p \cdot Se + kd_d \cdot ve) \cdot d + ka_p \cdot \theta \quad (\text{Equação IV})$$

$$vl(t) = (kd_p \cdot Se + kd_d \cdot ve) \cdot d - ka_p \cdot \theta \quad (\text{Equação V})$$

Para determinar as constantes das equações acima, utilizou-se testes de tentativa e erro até se obter o melhor ganho.

Mesmo utilizando o controlador PD para atenuar os erros dentro do sistema, a determinação das velocidades do robô podem ainda estar com erros, pois os cálculos de velocidade dependem de sua posição atual, mas devido à inércia e o tempo de processamento gastos pelo computador até o envio de comandos de velocidade ao robô, o mesmo não se encontra na posição onde a velocidade fora calculada. Portanto, a utilização do Filtro de Kalman (FK) (Gupta et al, 2003) que é um sistema de estimativa linear dinâmico recursivo estocástico, é capaz de estimar estados do futuro, que evolui de um estado para outro no decorrer do tempo, de acordo com seu estado atual.

Os ganhos de h e g do FK são determinados realizando testes, por tentativa e erro através de uma trajetória qualquer, nesse caso, uma trajetória em forma de “oito” onde se verificam quais são os melhores coeficientes que estimam da melhor forma a velocidade e posição respectivamente.

Após o desenvolvimento e aplicação do sistema de controle, obtem-se como resultados de testes no simulador FIRA (FIRA, 1998) (Robot Soccer v1.5a), o seguinte gráfico (Fig. 7) de saída para uma trajetória em formato de 8:

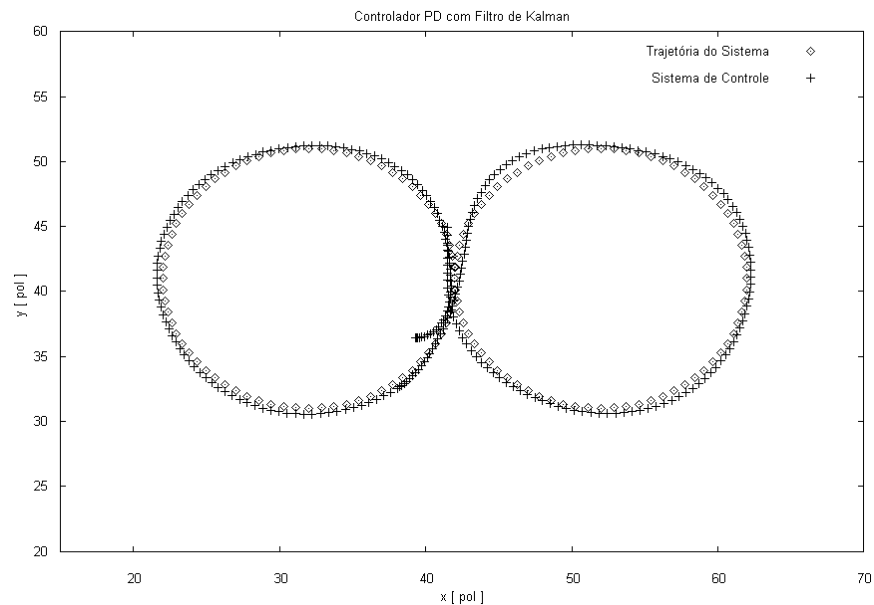


Figura 7 – Controlador PD com FK comparado com a trajetória do sistema.

4. Sistema de Estratégia Implementado

O sistema de estratégia implementado é baseado em zonas. O campo é dividido em 9 zonas, 3 na parte do ataque, 3 na parte do meio de campo e 3 zonas na área da defesa. Conforme figura 8.

Toda e qualquer estratégia de jogo começa pela análise da zona em que se encontram os robôs do time, os robôs adversários e a bola. A partir dessa análise, são gerados estados de jogo (uma máquina de estados) que analisa, para cada zona, a possibilidade de defesa ou ataque.

Zona A1	Zona M1	Zona D1
Zona A2	Zona M2	Zona D2
Zona A3	Zona M3	Zona D3
Ataque	Meio	Defesa

Figura 8 – Zonas de estratégia

Todas as zonas 1 e 3 são simétricas (A1 com A3, M1 com M3 e D1 com D3), assim, a mesma estratégia aplicada em uma é também aplicada em outra.

Para cada zona, como dito anteriormente, existe uma máquina de estado. A estratégia irá trabalhar com a zona onde se encontra a bolinha para decidir se um jogador do time irá interceptá-la para o ataque ou para a defesa. Decide-se pelo robô mais próximo da bola e que esteja atrás da posição x da bola na imagem, para interceptá-la. O outro jogador se posicionará em outra zona para rebote de bola.

Os estados são compostos por variáveis que indicam se cada robô da equipe está atrás ou a frente da bola, se a bola está de posse do jogador adversário ou do jogador da equipe, se a zona está livre ou com adversário, entre outras. Essas variáveis, em composição uma com as outras, definem a ação de cada jogador a ser tomada, que pode ser: atacar com chute a gol, passar bola para outro robô, defender ou interceptar a bola.

Dependendo da zona, uma ação tem mais peso do que outra. Por exemplo, a zona A2 a ação “atacar com chute a gol” é priorizado. Já na zona D2 a ação “defender” tem peso significativamente maior.

Para a movimentação do robô, é aplicado *Piecewise Bezier* (Hwang *et al*, 2002) de modo a traçar a trajetória a ser seguida pelos robôs da equipe. Para traçar a trajetória, usa-se zonas de desvio dos robôs. A primeira trajetória traçada é uma trajetória de retas desviando das zonas de desvio, conforme figura 9a.

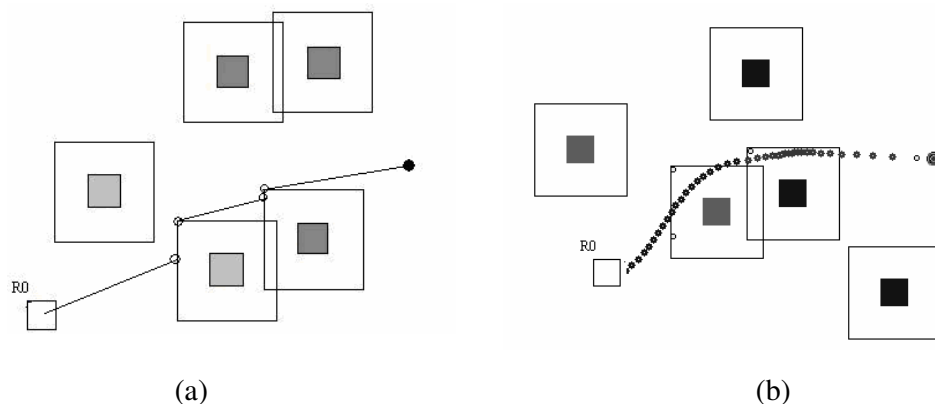


Figura 9 – Cálculo e Determinação de trajetórias usando Piecewise Bezier

Com os pontos da trajetória em retas, traça-se Bezier de modo a suavizar a trajetória que será passada, então, para o sistema de controle. A figura 9b mostra a trajetória definida com Bezier.

O controle da trajetória é feito com base no tempo estimado para o robô percorrer a trajetória. Determina-se o próximo ponto que o robô deve seguir nos próximos 30ms e isso é aplicado ao sistema de controle PD descrito anteriormente. O sistema de controle, então, atribui as velocidades de cada roda com base no ponto estimado e o ponto real do robô, fazendo o robô percorrer o espaço como desejado e seguindo sempre a trajetória. Perceba que o ciclo de visão, estratégia, trajetória e controle se repete a cada 30ms.

Já a estratégia do goleiro é bem específica. Primeiro, o goleiro utiliza a trajetória prevista para a bolinha. Essa trajetória é obtida considerando a posição atual da bolinha

e a posição da mesma na imagem anterior e é usada para determinar onde o goleiro deve ir para defender o gol antes que a bolinha chegue até ele.

A única direção do goleiro é para um ponto, chamado de ponto de defesa (pdd). Se esse ponto de defesa não existir, o goleiro deverá voltar ao ponto central do gol e ficar virado 90° sentido norte.

A determinação do ponto pdd é bem simples. Se a trajetória da bolinha interceptar o gol, o pdd é o ponto de intersecção da trajetória com uma linha imaginária que corta o meio da área do goleiro.

Se a bola estiver perto do goleiro, ele irá virar para bola e chutar. Se a bola estiver dentro da área, o pdd é exatamente o ponto da bola, para que o goleiro possa tirar a bola da área.

Essas são as diretrizes básicas da estratégia utilizada na equipe FEI06. Alguns detalhes ainda estão sendo aprimorados e, futuramente, algumas técnicas de raciocínio espacial serão incorporadas.

5. Conclusão

A equipe FEI06, ao contrário das equipes dos anos anteriores, utiliza-se de um sistema mais tolerante de visão e um sistema de controle bem mais aprimorado que os anteriores.

O sistema de visão é tolerante à variação de luz, pois depende apenas da forma (circular) para encontrar os robôs da equipe. Já a determinação dos robôs adversários pode sofrer influência da variação de luminosidade. O que deverá ser aprimorado no futuro.

Já o sistema de controle permite o robô seguir uma trajetória pré-definida e interceptar a bola em movimento. Todo o sistema permite ao robô jogadas aprimoradas de passe e drible.

As técnicas descritas neste artigo foram testadas somente em simuladores e serão testadas em robôs reais em um futuro próximo.

A estratégia implementada visa o jogo em equipe, com os robôs se posicionando em pontos estratégicos para formar jogadas e seguir trajetórias com o uso de Bezier. Entretanto, jogadas como passes e chutes a gol ainda precisam ser aprimoradas.

Referencias Bibliográficas

- Bianchi, R. A. C.; Reali-Costa, A. H.(2000). O Sistema de Visão Computacional do Time FutePOLI de Futebol de Robôs. In: CBA 2000 – Congresso Brasileiro de Automática 13., Florianópolis, 2000. Anais. Florianópolis, Sociedade Brasileira de Automática. p.2156-2161.
- Canny, J. (1986) A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, v.8, n. 6, p.679-698.
- FIRA (2005) FIRA Small League MiroSot Game Rules. Disponível em: <http://www.fira.net/soccer/mirosot/rules_slm.html>. Acesso em: 21 nov. 2005.

- Hwang, J., Arkin, R. C., Kwon, D. (2002). Mobile robots at your fingertip: Bezier curve on-line trajectory generation for supervisory control.
- Hough, P.V.C. (1959). Machine Analysis of Bubble Chamber Pictures, International Conference on High Energy Accelerators and Instrumentation, CERN.
- Hwang, J.H, Arkin, R. C. and Kwon, D.S. (2003) "Mobile robots at your fingertip: Bezier curve on-line trajectory generation for supervisory control," in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems..
- Ogata. K (1997) Engenharia de Controle Moderno, 3º Eds., LTC.
- Sen Gupta, G., Messom, C, & Demidenko, S. (2003) "Global Vision Based Optimization of Control Functions and Kalman Filtering for Robotic Applications", Proceedings of IEEE International Conference on Computational Intelligence, Robotics and Autonomous Systems, (CIRAS 2003), ISSN 0219-6131.
- Sen Gupta, G., Messom, C, Demidenko, S. (2004). Vision Assisted Measurement for Optimization of Robot Motion and Position Control Functions, IEEE Instrumentation and Measurement Technology Conference, Como, Italy, pp 297- 302, ISBN 0-7803-8248-1.
- Vieira, F. C. et al. Position and Orientation Control of a Two-Wheeled Differentially Driven Nonholonomic Mobile Robot (2004). I ICINCO - International Conference on Informatics in Control, Automation and Robotics. Setúbal, Portugal.