

ESTIMAÇÃO DE PARÂMETROS PARA NAVEGAÇÃO DE UM ROBÔ MÓVEL A PARTIR DE DADOS OBTIDOS POR UM SISTEMA DE VISÃO COMPUTACIONAL

ELYSON CARVALHO¹, ALAN SÁ¹, ELINE SANTOS¹, EDUARDO FREIRE¹, LEONARDO MATOS¹, BENEDITO LUCIANO²

¹*Centro de Ciências Exatas e Tecnologia – Universidade Federal de Sergipe
Av. Marechal Rondon, s/n, 49100-000, São Cristovão-SE*

²*Departamento de Engenharia Elétrica – DEE – Universidade Federal de Campina Grande
Rua Aprígio Veloso, 882, Bodocongó, 58109-900 – Campina Grande-PB*

E-mails: elyson_ecase@yahoo.com.br; alanvss@gmail.com; elinealvessantos@gmail.com; efreire@ufs.br; lmatos@ufs.br; benedito@dee.ufcg.edu.br

Abstract— The correct operation of a control system depends on the reliable measure of its state variables and other necessary parameters. Cameras are increasingly being used due to their capacity to provide a greater amount of information about the operation environment of the mobile robot. On the other hand, the presence of noise and other inherent problems of the algorithms often used to perform the image processing affects the reliability of the extracted data. A feasible approach to improve the reliability of the parameters extracted from images consists on the use of estimation techniques. In this work an Extended Information Filter was used to estimate the mobile robot pose, and a $\alpha\beta$ filter was used to estimate its angular and linear velocities.

Keywords— Mobile Robot; Control Architecture; Vision Feedback, Parameter Estimation.

Resumo— A correta operação de um sistema de controle depende da medição confiável de suas variáveis de estado e outros parâmetros necessários. O uso de câmeras tem crescido devido à sua capacidade de obter uma maior quantidade de informações a respeito do ambiente de operação do robô móvel. No entanto, a presença de ruído e outros problemas inerentes aos algoritmos usados para o processamento das imagens afetam a confiabilidade dos dados extraídos. Uma possível abordagem para melhorar a confiabilidade dos parâmetros extraídos a partir de imagens consiste no uso de técnicas de estimação. Nesse trabalho foi utilizado um Filtro de Informação Estendido para estimar a posição e orientação do robô móvel e um filtro $\alpha\beta$ para estimação de suas velocidades angular e linear.

Palavras-chave— Robô Móvel; Arquitetura de Controle; Realimentação Visual, Estimação de Parâmetros.

1 Introdução

A robótica vem se tornando cada vez mais importante e presente no cotidiano da sociedade. Robôs manipuladores são muito usados na indústria. Observam-se robôs móveis sendo utilizados em situações de risco para o homem, como, por exemplo, desarmar bombas, e em locais de difícil acesso, como no fundo do mar ou exploração espacial. Existem outras aplicações para os robôs, por isso nota-se um crescente interesse na realização de pesquisas nessa área.

Um dos focos de pesquisa em robótica consiste no desenvolvimento de melhores sistemas de navegação para o controle de robôs móveis. Como não estão presos num único local de operação, tais robôs são potencialmente mais úteis do que os atuais robôs manipuladores utilizados na indústria. No entanto, para que as suas potencialidades possam ser aproveitadas, os robôs móveis, muitas vezes, necessitam de sistemas de sensoriamento e controle muito mais complexos do que os robôs manipuladores.

Em Carvalho *et al* (2005) um sistema de controle para navegação de um robô móvel baseado em fusão de sinais de controle e realimentação visual foi apresentado. O sistema de navegação que foi desen-

volvido permite que o robô navegue de forma segura e eficiente em ambientes fechados e planos, semelhantes aos encontrados em escritórios e residências. Tal sistema de navegação é composto por um controlador de posição final e um controlador de evitar obstáculos, e foi projetado de acordo com a arquitetura de controle para navegação de robôs móveis apresentada em (Freire *et al.*, 2004), que realiza a fusão das saídas dos controladores. Um sistema de visão computacional recebe imagens do ambiente de operação do robô móvel adquiridas por uma *webcam* e as processa, a fim extrair parâmetros como posição e orientação do robô móvel, e a detecção de possíveis obstáculos. Esses dados são passados ao sistema de controle, que com base neles, gera em sua saída as velocidades linear e angular necessárias para que ele possa alcançar o seu ponto de destino.

Apesar de funcionar bem na maioria dos experimentos, ruídos de alta frequência e outros problemas inerentes aos algoritmos usados para o processamento das imagens afetavam o cálculo das variáveis de estado usadas pelo controlador, o que o induzia a erros. Por essa razão, foram utilizadas técnicas de estimação de parâmetros na tentativa de melhorar a confiabilidade das variáveis de estado e de outros parâmetros necessários ao sistema de controle.

A posição e orientação do robô móvel são extraídas da imagem pelo sistema de visão computacional, no entanto, muitas vezes apresentavam variações bruscas, impossíveis de ocorrer devido ao fato de que as velocidades linear e angular do robô são limitadas. Para evitar problemas como esse, elas foram estimadas usando um filtro de informação estendido (Mutambara, 1998). Por outro lado, as velocidades linear e angular do robô são obtidas a partir da imagem pelo sistema de visão computacional, mas tal processo é extremamente ruidoso. O sistema implementado em Carvalho *et al* (2005) não realizava a medição da velocidade linear e angular do robô. Neste trabalho, as velocidades linear e angular do robô foram estimadas usando um filtro α - β (Kalata e Murphy, 1997), o que produziu melhores resultados.

O artigo está estruturado da seguinte forma: a seção 2 apresenta brevemente o sistema de controle utilizado. A seção 3 trata do sistema de visão computacional implementado. Os estimadores utilizados são apresentados na seção 4. A seção 5 trata do robô móvel desenvolvido. Os resultados obtidos são mostrados na seção 6. Na seção 7 são apresentadas as conclusões deste trabalho e sugerem-se alguns trabalhos futuros. Ao final estão os agradecimentos e as referências bibliográficas.

2 O Sistema de Controle Proposto

A arquitetura de controle para navegação proposta (Carvalho *et al*, 2005) é apresentada na Fig. 1. Ela é composta por dois controladores não lineares: um controlador de posição final, e um controlador para evitar obstáculos. Detalhes sobre o projeto dos controladores, como demonstração de estabilidade assintótica segundo o critério de Lyapunov, equações de malha fechada, ações de controle, etc. podem ser encontrados em (Secchi, 1999), (Secchi *et al*, 2001), (Freire *et al*, 2004).

Em arquiteturas de controle para navegação de robôs móveis baseadas em fusão de comandos, como a que é usada neste trabalho, um conjunto de controladores ou comportamentos contribui para gerar os sinais de controle de velocidade linear e angular (Pirajanian, 2000). A cada execução do laço de controle, cada controlador calcula uma velocidade linear e uma velocidade angular. Um método ou esquema de fusão é então utilizado para combinar as diversas velocidades lineares e angulares, produzindo os sinais de controle que são efetivamente enviados para o robô móvel. Arquiteturas de controle baseadas em fusão possuem a característica de produzirem ações de controle mais suaves, melhorando o desempenho do robô, inclusive sob o aspecto do consumo de baterias. Detalhes sobre a arquitetura proposta e as razões que levaram à sua escolha podem ser encontrados em (Freire *et al*, 2004).

Utilizou-se uma simples média ponderada como método de fusão. Obviamente não se trata de um método de fusão otimizado sob nenhum aspecto e so-

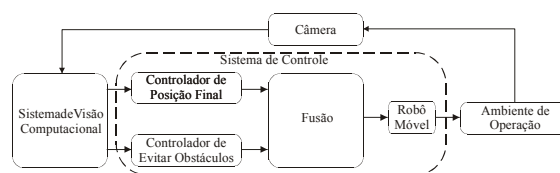


Figura 1. Esquema do sistema de controle para navegação.

mente foi utilizado devido à sua simplicidade, o que permitiu uma implementação imediata. Apesar do método de fusão utilizado ser precário, os resultados obtidos são satisfatórios, o que reflete a funcionalidade da arquitetura de controle utilizada.

Os pesos atribuídos a cada controlador são calculados de forma determinística. O método utilizado para a determinação dos pesos usa o valor da menor distância calculada entre o robô e os obstáculos. Assim, quanto menor for a distância (d_{obst}), maior será o peso do controlador de evitar obstáculos e menor será o peso do controlador de posição final.

$$Ppd = \frac{d_{obst}}{d_{max}} \quad (1)$$

$$Peo = \frac{(d_{max} - d_{obst})}{d_{max}} \quad (2)$$

Ppd e Peo são os pesos associados aos controladores de posição final e evitar obstáculos respectivamente, e d_{max} é a distância a partir da qual o controlador de evitar obstáculos atua.

3 Sistema de Visão Computacional

O sistema de visão computacional desenvolvido realiza a extração de características relevantes sobre as imagens capturadas e alimenta o sistema de controle. As imagens fornecidas pela câmera ao sistema de visão são policromáticas e frequentemente corrompidas com algum tipo de ruído, como ruído aleatório aditivo ou ruído de iluminação. É necessário, portanto, ser realizado um processamento preliminar antes de iniciar a extração de características. O pré-processamento consiste na segmentação dos objetos presentes na cena, que tipicamente é composta pelo robô, pelos obstáculos e pelo fundo.

A segmentação do fundo é realizada por um algoritmo de limiarização global (Gonzalez e Woods, 1992). Primeiro procura-se na imagem a cor predominante, isto é, a que ocorre com maior frequência. Em seguida faz-se uma nova varredura da imagem desta vez rotulando os *pixels* cuja distância no espaço tricromático à cor predominante seja inferior a um dado limiar. A segmentação do robô também é feita pela distância vetorial entre *pixels*. O robô é identificado como o objeto cuja distância no espaço tricromático entre sua cor e uma determinada cor de referência é inferior a um limiar estabelecido. Para visualizar o robô apropriadamente, sua parte superior é coberta com uma capa cuja parte dianteira é de cor verde e a parte traseira é de cor vermelha (vide Fig. 2). Dessa forma, é possível distinguir a parte diantei-

ra da traseira, o que é fundamental para a determinação da orientação do robô móvel. Uma vez que a câmera é posicionada acima do ambiente de operação do robô, nas imagens capturadas vê-se na realidade sua capa de revestimento. Para evitar confundir o robô com outros objetos de mesma cor presentes na imagem, na inicialização do sistema de controle, o usuário especifica a localização do robô clicando primeiro sobre a sua parte dianteira e depois sobre a sua parte traseira. Então, a cada novo *frame* enviado, a nova posição do robô é procurada somente em uma vizinhança local em torno de sua posição anterior.

Tendo sido segmentado o robô, é possível obter suas características, como área, posição e orientação. A área do robô é calculada pelo somatório dos *pixels*. Sua posição é definida como sendo as coordenadas de seu centro de área (Horn, 1986). A orientação do robô é obtida segundo o seguinte processo: determina-se o centro de área da parte dianteira do robô (cor verde) e o centro de área da parte traseira do robô (cor vermelha); a orientação do robô móvel é dada pela orientação da reta que conecta os centros de área de sua parte dianteira e de sua parte traseira.

O processo utilizado na verificação de obstáculos é baseado num método de sonar visual como apresentado a seguir. São utilizados feixes que saem do centro de massa do robô em diferentes ângulos e que vão percorrendo a imagem *pixel* a *pixel*. A varredura de cada feixe termina quando ele encontra um obstáculo ou quando atinge um dos limites da imagem. Um *pixel* é considerado pertencente a um obstáculo quando ele não foi rotulado como fundo. A fim de eliminar a presença de ruído, que é um *pixel* rotulado indevidamente como objeto, verifica-se se em uma vizinhança 3x3 ele foi isoladamente rotulado como objeto entre outros rotulados como fundo. As coordenadas referentes ao ponto de parada de cada feixe são usadas no cálculo da distância ao centro de massa do robô. Esses valores são guardados num vetor, que é passado para o sistema de controle.

Para a implementação dos feixes, foi utilizado o algoritmo do ponto médio (Foley *et al.*, 1992). Este algoritmo traça retas *pixel* a *pixel*, e permite que sejam feitas operações em cada um deles no momento em que são alcançados pelo feixe. Isso viabiliza a checagem do valor do *pixel*, para saber se ele pertence ao fundo, ao robô ou aos limites da cena.

4. Estimação de Parâmetros

Nesta seção são descritos o filtro de informação estendido, usado para estimar a posição e orientação do robô, e o filtro α - β , usado para estimar as suas velocidades linear e angular.

4.1 O Filtro de Informação Estendido

O Filtro de Informação estendido é o equivalente algébrico do Filtro de Kalman estendido e foi introdu-

zido pela primeira vez em Mutambara, (1998). Uma das principais vantagens do Filtro de Informação com relação ao Filtro de Kalman é o fato de que o Filtro de Informação é inicializado de forma mais simples. Isto é relevante, sobretudo, para sistemas não lineares devido ao fato de que uma inicialização inadequada pode impedir a convergência do filtro.

Seja o modelo para o sistema

$$\begin{cases} \mathbf{x}(k) = \mathbf{F}(\mathbf{x}(k-1), (k-1)) + \mathbf{w}(k) \\ \mathbf{z}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{v}(k) \end{cases} \quad (3)$$

onde $\mathbf{x}^T(k) = [x(k) \ y(k) \ \varphi(k)]$ é o vetor de estados $n \times 1$ do processo no instante t_k ; \mathbf{F} o vetor de transição de estados e possui a mesma dimensão de \mathbf{x} ; $\mathbf{z}(k)$ é o vetor de observação $m \times 1$ no instante t_k ; $\mathbf{H}(k)$ é a matriz de observação $m \times n$ que descreve a conexão ideal (sem ruído) entre as medições e o vetor de estados no instante t_k . Assume-se que o vetor $n \times 1$ $\mathbf{w}(k)$ consiste em uma seqüência branca de ruídos gaussianos com covariância conhecida, a qual representa o ruído associado ao sistema, e $\mathbf{v}(k)$ é um vetor $m \times 1$ representando o erro de medição, também consistindo em uma seqüência branca de ruídos gaussianos, com covariância conhecida. Também assume-se que a correlação cruzada entre $\mathbf{v}(k)$ e $\mathbf{w}(k)$ é zero, e as matrizes de covariância associadas a eles são dadas por:

$$E[\mathbf{w}(k)\mathbf{w}^T(k)] = \begin{cases} \mathbf{Q}(k) & i = k \\ 0 & i \neq k \end{cases} \quad (4)$$

$$E[\mathbf{v}(k)\mathbf{v}^T(k)] = \begin{cases} \mathbf{R}(k) & i = k \\ 0 & i \neq k \end{cases} \quad (5)$$

$$E[\mathbf{w}(k)\mathbf{v}^T(i)] = 0, \quad \forall k, i \quad (6)$$

Para a estimação da posição e orientação do robô os parâmetros do modelo do sistema são dados por:

$$\mathbf{F} = \begin{bmatrix} x(k-1) + |\bar{u}(k-1)|\Delta t \cos[\varphi(k-1)] \\ y(k-1) + |\bar{u}(k-1)|\Delta t \sin[\varphi(k-1)] \\ \varphi(k-1) + \Delta t \omega(k-1) \end{bmatrix}, \mathbf{z} = \begin{bmatrix} x_c(k) \\ y_c(k) \\ \varphi_c(k) \end{bmatrix} \quad (7)$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} R_1 & 0 & 0 \\ 0 & R_2 & 0 \\ 0 & 0 & R_3 \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} Q_1 & 0 & 0 \\ 0 & Q_2 & 0 \\ 0 & 0 & Q_3 \end{bmatrix} \quad (8)$$

onde x_c , y_c e φ são as coordenadas e orientação do robô obtidas pelo sistema de visão computacional.

A matriz \mathbf{Q} é calculada usando um algoritmo recursivo para o cálculo de variância, enquanto que a matriz \mathbf{R} é calculada usando o mesmo algoritmo recursivo, porém com a introdução de um fator de esquecimento. As equações do Filtro de Informação estendido são (Mutambara, 1998):

Predição:

$$\hat{\mathbf{y}}(k|k-1) = \mathbf{Y}(k|k-1)\mathbf{F}(k, \hat{\mathbf{x}}(k-1|k-1), (k-1)) \quad (9)$$

$$\mathbf{Y}(k|k-1) = [\nabla \mathbf{F}_x(k)\mathbf{Y}^{-1}(k-1|k-1)\nabla \mathbf{F}_x^T(k) + \mathbf{Q}(k)]^{-1} \quad (10)$$

Estimação:

$$\hat{\mathbf{y}}(k|k) = \hat{\mathbf{y}}(k|k-1) + \mathbf{i}(k) \quad (11)$$

$$\mathbf{Y}(k|k) = \mathbf{Y}(k|k-1) + \mathbf{I}(k) \quad (12)$$

onde $\mathbf{I}(k) = \mathbf{R}^{-1}(k)$, $\mathbf{i}(k) = \mathbf{R}^{-1}\mathbf{z}(k)$ e $\nabla \mathbf{F}_x$ é o Jacobiano de \mathbf{F} .

4.2 O Filtro α - β

O filtro α - β propõe constantes α e β ótimas, a partir do filtro de Kalman, para medição e filtragem de parâmetros de trajetórias em condições estacionárias, onde pode-se assumir como constantes (Kalata e Murphy, 1997; Rawicz *et al*, 1998):

- O período de amostragem;
- Variância do ruído da medida;
- Aceleração.

A partir do modelo Newtoniano, é possível equacionar um sistema de posição (x,y) e ângulo φ , como:

$$\begin{bmatrix} x(k) \\ y(k) \\ \varphi(k) \end{bmatrix} = \begin{bmatrix} x(k-1) \\ y(k-1) \\ \varphi(k-1) \end{bmatrix} + \begin{bmatrix} T \cos(\varphi) & 0 \\ T \sin(\varphi) & 0 \\ 0 & T \end{bmatrix} \cdot \begin{bmatrix} \bar{u}(k-1) \\ \omega(k-1) \end{bmatrix} \quad (13)$$

onde T é o período de amostragem e \bar{u} e ω são respectivamente as velocidades linear e angular.

O vetor de observação z é dado por:

$$z = \begin{bmatrix} x_c(k-1) \\ y_c(k-1) \\ \varphi_c(k-1) \end{bmatrix} + n(k-1) \quad (14)$$

onde $n(k-1)$ representa o ruído da medição.

O processo do filtro α - β é dado por:

Predição:

$$\begin{bmatrix} x(k|k-1) \\ y(k|k-1) \\ \varphi(k|k-1) \end{bmatrix} = \begin{bmatrix} x(k-1|k-1) \\ y(k-1|k-1) \\ \varphi(k-1|k-1) \end{bmatrix} + \begin{bmatrix} T \cos(\varphi(k-1|k-1)) & 0 \\ T \sin(\varphi(k-1|k-1)) & 0 \\ 0 & T \end{bmatrix} \cdot \begin{bmatrix} \bar{u}(k-1) \\ \omega(k-1) \end{bmatrix} \quad (15)$$

Correção:

$$\begin{bmatrix} x(k|k) \\ y(k|k) \\ \varphi(k|k) \end{bmatrix} = \begin{bmatrix} x(k-1|k-1) \\ y(k-1|k-1) \\ \varphi(k-1|k-1) \end{bmatrix} + \alpha \begin{bmatrix} x_c(k) - x(k|k-1) \\ y_c(k) - y(k|k-1) \\ \varphi_c(k) - \varphi(k|k-1) \end{bmatrix} \quad (16)$$

$$\begin{bmatrix} \bar{u}(k|k) \\ \omega(k|k) \end{bmatrix} = \begin{bmatrix} \bar{u}(k-1|k-1) \\ \omega(k-1|k-1) \end{bmatrix} + \beta \begin{bmatrix} \rho_c(k) - \rho(k|k-1) \\ \varphi_c(k) - \varphi(k|k-1) \end{bmatrix} \quad (17)$$

onde:

$$\rho_c = \sqrt{x_c^2 + y_c^2} \quad (18)$$

$$\rho(k|k) = \sqrt{x(k|k)^2 + y(k|k)^2} \quad (19)$$

De Kalata e Murphy, (1997) e Rawicz *et al*, (1998), obtém-se o índice de seguimento A , dado por:

$$A^2 = \frac{\beta^2}{(1-\alpha)} \quad (20)$$

A relação entre os parâmetros α e β ótimos pode ser obtida por:

$$\beta = 2(2-\alpha) - 4\sqrt{1-\alpha} \quad (21)$$

Combinando (19) e (20), é possível explicitar o valor de α , dado por:

$$\alpha = \frac{-L^2 + \sqrt{L^4 + 16L^2}}{8} \quad (22)$$

onde $L^2 = A^2 + 8A$.

5 O Robô Móvel Desenvolvido

A plataforma de locomoção do robô utilizado nos experimentos foi construída utilizando-se placas de acrílico. Na base, foram fixados dois motores LEGO® que possuem caixa de redução embutida. Para o acionamento dos mesmos utilizou-se o CI SN754410 que é constituído de duas pontes H.

O controle embarcado do robô é feito por um circuito baseado no microcontrolador PIC16F877 da Microchip®. Este possui dois pinos em uma de suas portas que têm a técnica de PWM implementada e por isso são utilizados, juntamente com as pontes H, para o acionamento dos motores. Uma outra característica de relevância nesse microcontrolador é que possui um pino específico para a comunicação serial assíncrona, e através deste ele recebe os dados.

A comunicação entre o computador e o robô foi feita através de um enlace infravermelho.

6 Resultados

Vários experimentos foram realizados, porém, por questão de espaço, somente apresenta-se um deles.

O sistema de controle do robô, incluindo o sistema de visão computacional, foi desenvolvido em C++ Builder®. A Fig. 3 mostra a tela de execução desse aplicativo.

A Fig. 4 mostra a trajetória do robô saindo do ponto (0,54; 0,46), com orientação de aproximadamente 60° com relação ao eixo x, em direção ao ponto (1,35; 1,30), tendo um obstáculo conforme mostrado na Fig. 4. Pode-se observar que para alcançar o seu ponto de destino o robô faz uma curva para desviar do obstáculo, evitando assim uma colisão. É importante salientar que o sistema de controle considera os limites da imagem como obstáculos para evitar que o robô saia do campo de visão da câmera. As figuras 5, 6 e 7 apresentam, respectivamente, os valores medidos pelo sistema de visão computacional e estimados pelo filtro de informação estendido (descrito na seção 4.1), das coordenadas x e y do robô e de sua orientação φ . Analisando as figuras 5 e 6 é possível notar que apesar de um erro de medição do sistema de visão computacional para as coordenadas x e y do robô, as respectivas coordenadas estimadas não são afetadas de forma significativa. As figuras 8 e 9 apresentam, respectivamente, os valores medidos e estimados das velocidades linear e angular desenvolvidas pelo robô ao longo do experimento.

7 Conclusões

O filtro de informação estendido e o filtro α - β se mostraram eficientes para realizar a tarefa de estimação das coordenadas, orientação e velocidade angular e linear do robô.

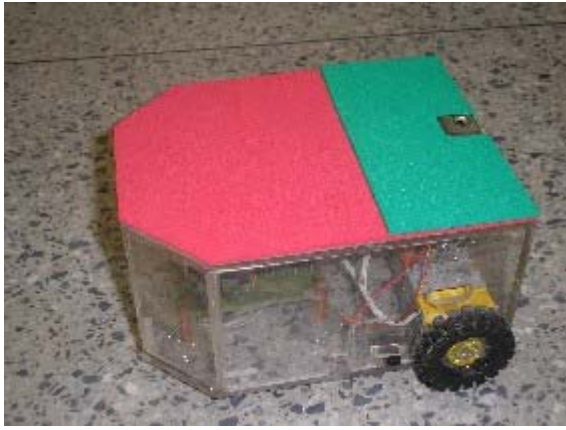


Figura 2: Robô Móvel

Tendo como base valores de variáveis de estado e parâmetros mais confiáveis o sistema de controle do robô pôde funcionar melhor, lavando o robô de seu ponto de origem ao seu ponto de destino de forma segura e eficiente, realizando trajetórias curtas e evitando obstáculos presentes no caminho do robô.

O robô móvel desenvolvido, apesar de passar por constantes mudanças para adequar-se às necessidades do projeto, apresenta um ótimo desempenho, servindo atualmente de plataforma de testes para outros projetos de iniciação científica.

Como trabalhos futuros pretende-se acrescentar uma segunda câmera para captura de imagens para que se possa ter uma visão do ambiente de operação do robô sob dois ângulos distintos ou para ampliar a área de operação do robô móvel. Na primeira aplicação as duas câmeras apontam para a mesma área de trabalho, a partir de pontos e ângulos diferentes. O objetivo neste caso é o de detectar obstáculos que não podem ser detectados sob determinados ângulos por ter uma de suas dimensões muito reduzida. Na segunda aplicação as duas câmeras são colocadas no teto e orientadas perpendicularmente com relação ao piso, porém em pontos diferentes. Neste caso, o objetivo consiste em ampliar a área de operação do robô móvel. Deve-se também, substituir o enlace de infravermelho por um enlace de rádio.

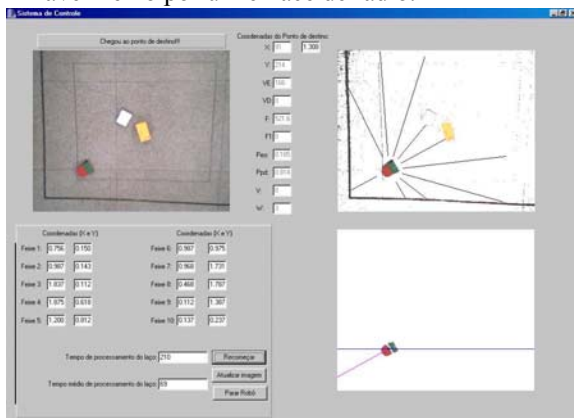


Figura 3: Tela de execução do sistema de controle integrado com o sistema de visão e interface de comunicação com o robô.

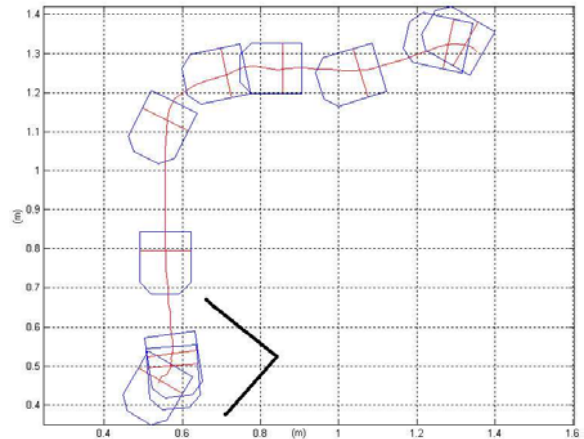


Figura 4: Trajetória descrita pelo robô móvel.

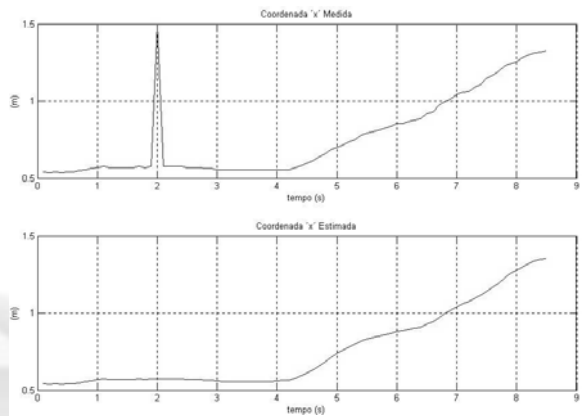


Figura 5: Coordenada x do robô medida pelo sistema de visão e estimada pelo filtro de informação descentralizado.

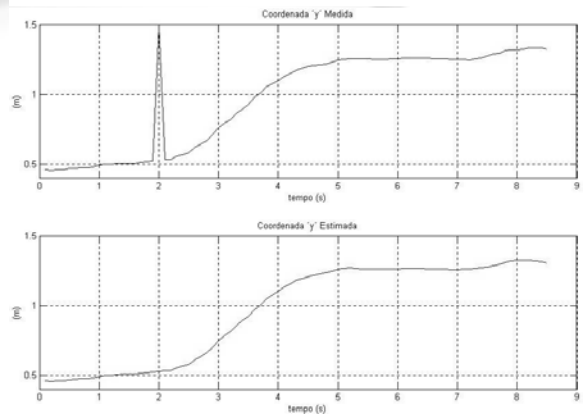


Figura 6: Coordenada y do robô medida pelo sistema de visão e estimada pelo filtro de informação descentralizado.

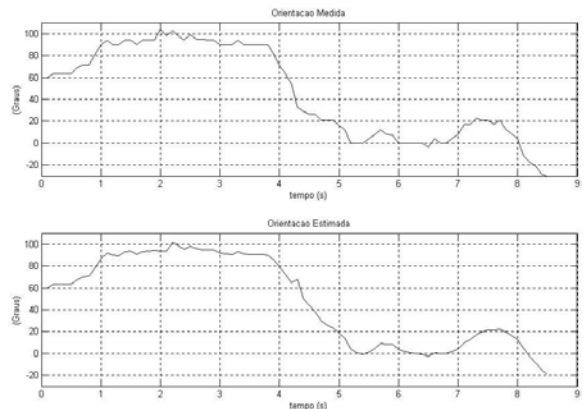


Figura 7: Orientação do robô medida pelo sistema de visão e estimada pelo filtro de informação descentralizado.

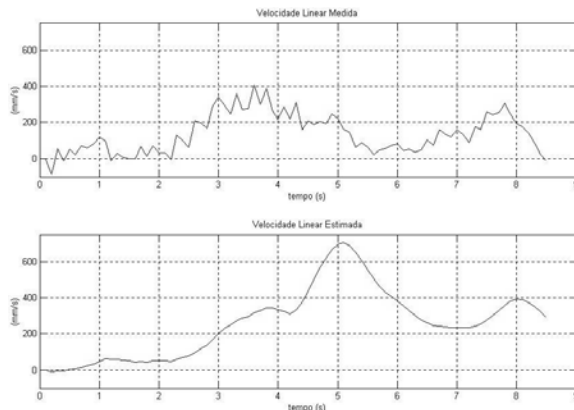


Figura 8: Velocidade linear medida pelo sistema de visão e estimada pelo filtro α - β .

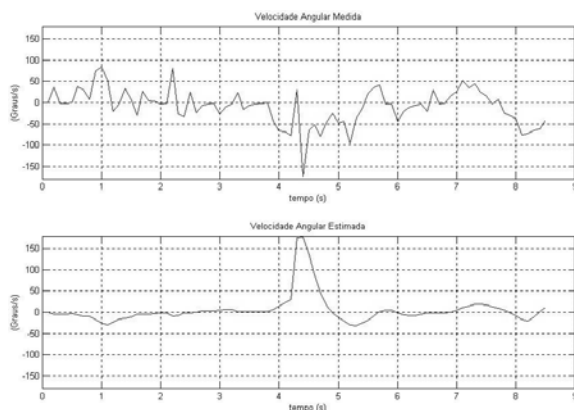


Figura 9: Velocidade angular medida pelo sistema de visão e estimada pelo filtro α - β .

Agradecimentos

Este trabalho foi realizado com o apoio do CNPq, através do programa de bolsas de iniciação científica do PIBIC.

Os autores desejam agradecer também aos colegas Lucas Molina, Jânio Coutinho Canuto, João Jenisson Vieira da Silva, Bruno Bittencourt da Costa, Beatriz Trinchão Andrade e Jugurta Rosa Montalvão Filho por todo suporte prestado em diversas fases deste trabalho.

Finalmente, os autores desejam agradecer ao CENPES/PETROBRÁS, CNPq e CAPES pelo apoio concedido ao desenvolvimento desse trabalho.

Referências Bibliográficas

Bemporad, R.; Di Marco, M. and Tesi, A. (1997). Wall-following controllers for sonar-based mobile robots. *Proc. 36th. IEEE Conf. on Decision and Control*, San Diego, Dec.

Carelli, R. and Freire, E. (2003). Corridor Navigation and Wall-Following Stable Control for Sonar-Based Mobile Robots. *Robotics And Autonomous Systems*, v. 45, p. 235-247.

Carvalho, E. A. N., Costa, B. B., Andrade, B. T., Freire, E. O. e Matos, L. N. (2005). Desenvolvimento de um Sistema de Controle para Navegação de um Robô Móvel Baseado em Fusão de

Sinais de Controle e Realimentação Visual. 7^o *Simpósio Brasileiro de Automação Inteligente – VII SBAI*. São Luis, Maranhão.

Foley, J. D., Dam, A.; Feiner, S. K e Hughes, J. F. (1992). *Computer Graphics: Principles and Practice*. Editora Addison-Wesley, EUA.

Freire, E.; Bastos-Filho, T.; Sarcinelli-Filho, M. and Carelli, R. (2004). A New Mobile Robot Control Architecture: Fusion of the Output of Distinct Controllers. *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*, v. 34, n. 1, p. 419-429.

Gonzalez, R.C. e Woods, R.E. (2002). *Digital Image Processing*. Editora Pearson Education.

Horn, B. K. P. (1986). *Robot Vision*. Editora McGraw-Hill, Massachusetts.

Jones, J. L.; Seiger, B. A. e Flynn, A. M. (1999). *Mobile Robots: Inspiration to Implementation*. Massachusetts, A K Peters.

Kalata, P. R., Murphy, K. M. (1997). Alpha-Beta Target Tracking and Track Rate Variations. *29th Southeastern Symposium on System Theory (SSST '97)*. p. 70.

Mccomb, G. (1987). *The Robot Builder's Bonanza: 99 Inexpensive Robotics Projects*. USA: McGraw-Hill.

Mutambara, A. G. O. (1998). *Decentralized Estimation and Control for Multi-sensor Systems*. EUA: *CRC Press*.

Pirjanian, P. Multiple objective behavior-based control. *Robotics and Autonomous Systems*, 31, pp. 53-60 (2000).

Rawicz, P. L., Kalata, P. R., Chmielewski, T. A. e Murphy, K. M. (1998). On α - β Target Tracking Initiation. *Proceedings of the 1998 American Control Conference*. Vol. 4, p. 2434-2438.

Sasiadek, J. Z and Hartana, P. (2000). Odometry and sonar data fusion for mobile robot navigation. *6th. IFAC Symposium on Robot Control, SYROCO '00*. Vienna, Austria. Preprints, Vol.II, p. 531-536.

Secchi, H. (1999). *Control de Vehículos Autoguiados con Realimentación Sensorial*. Dissertação de Mestrado – Universidad Nacional de San Juan – Argentina.

Secchi, H., Carelli, R. e Mut, V. (2001). Discrete Stable Control of Mobile Robots with Obstacles Avoidance, *International Conference on Advanced Robotics, ICAR'01*, Budapeste, Hungria.

Souza, C. A. B. (2003). *Desenvolvimento de um Sistema de Visão Computacional para Aplicações de Controle Servo-Visual de Robôs Móveis*. Monografia. Universidade Tiradentes, Aracaju.