

CONTROLE DE FORMAÇÃO DE UMA EQUIPE DE ROBÔS BASEADO EM TÉCNICAS NÃO LINEARES E VISÃO OMNIDIRECIONAL

CHRISTIANO COUTO GAVA*, FABRICIO BORTOLINI DE SÁ*, MARINO FRANK CYPRIANO*, RAQUEL FRIZERA VASSALLO*, TEODIANO FREIRE BASTOS FILHO*

*UFES - Universidade Federal do Espírito Santo, Depto. de Engenharia Elétrica, Vitória-ES, Brasil

Emails: christiano@ele.ufes.br, fabricio@ele.ufes.br, marino@ele.ufes.br, raquel@ele.ufes.br, tfbastos@ele.ufes.br

Abstract— In this work a mobile robot cooperation strategy based on computational vision is presented. Such strategy is applied to a mobile robot team formed by simple and cheap robots and a leader robot with more computational power. The leader has an omnidirectional visual system and uses color segmentation to obtain the pose of the follower robots. This visual information is used by a nonlinear stable controller that manages team formation. Simulations and tests were run and current results are encouraging.

Keywords— Robot Cooperation, Nonlinear Control, Omnidirectional Vision.

Resumo— Neste trabalho é apresentada uma estratégia de cooperação robótica baseada em visão computacional. Tal estratégia é aplicada a uma equipe de robôs móveis formada por robôs simples e baratos e um robô dotado de maior capacidade de processamento para liderar esta equipe. O líder possui um sistema de visão omnidirecional e através de segmentação por cores promove a realimentação visual de um controlador não linear estável de posição e orientação, cuja finalidade é controlar a formação da equipe. Simulações e experimentos foram realizados e os resultados obtidos são animadores.

Palavras-chave— Robótica Cooperativa, Controle Não-linear, Visão Omnidirecional.

1 Introdução

Atualmente, robótica móvel representa um papel cada vez mais importante na sociedade. Entretanto, tarefas do mundo real exigem, em sua maioria, grande capacidade de processamento e vários sensores para interagir com o ambiente e outros agentes. Um único robô equipado para realizar, sozinho, tarefas deste tipo, se torna muito caro. Cooperação robótica representa uma alternativa para este problema. Permite a distribuição dos sensores e do processamento entre robôs mais simples e baratos, a conclusão das tarefas em menos tempo, além da robustez a falhas, uma vez que a falha de um robô não significa, necessariamente, falha de todo o grupo (Arai et al., 2002; Cao et al., 1997).

Alguns exemplos de tarefas desempenhadas cooperativamente são resgates, transporte de cargas, vigilância, mapeamento e exploração (Matáric et al., 1995; Feddema et al., 2002; Rus et al., 1995). Durante o movimento, os robôs trocam informações entre si para se posicionar corretamente, evitar áreas já visitadas e, com isso, aumentar a eficiência do grupo, ou ainda para concentrar esforços em determinado local.

Na maioria das aplicações em grupo, é necessário o controle de formação entre os robôs para que estes assumam posturas (posição e orientação) adequadas para a execução da tarefa. Alguns trabalhos, como (Desai et al., 2001), propõem estratégias de controle descentralizado. Outros como (Feddema et al., 2002) adotam uma abordagem que pode ser estendida a vários robôs mantendo a estabilidade do controlador. Há ainda os controles

de forma centralizada, onde o grupo é coordenado por um ou mais líderes (Lewis and Tan, 1997).

Para se conseguir um bom controle de formação, é preciso estimar corretamente a postura dos robôs do grupo. Visão computacional permite a obtenção destas informações de forma fácil e sem interferência no ambiente. Além disso, sistemas de visão que fornecem amplo campo visual auxiliam ainda mais a percepção, o que os torna atrativos para aplicações baseadas em cooperação (Das et al., 2002; Vidal et al., 2004).

Uma boa maneira de se ampliar o campo de visão é usando imagens omnidirecionais (360° de campo visual horizontal) (Nayar, 1997). Estas imagens podem ser obtidas através de uma combinação de um espelho (parabólico, hiperbólico ou elíptico) com uma câmara convencional.

Este trabalho propõe uma estratégia de controle de formação baseada em visão omnidirecional. A estratégia é aplicada a uma equipe de robôs móveis cujo líder possui maior capacidade de processamento e um sistema de visão omnidirecional, enquanto que os demais componentes (seguidores) são mais simples e baratos. Os seguidores são identificados por retângulos coloridos colocados no topo de sua plataforma. A partir das imagens omnidirecionais, o líder estima a postura de cada seguidor por segmentação de cores. Em seguida, esta informação é usada para controlar a formação da equipe. Isto é feito usando-se um controlador não linear estável de posição e orientação (Carelli et al., 2006) que define as velocidades linear e angular que os seguidores devem ter para que a equipe entre e permaneça em uma determinada formação.

Assim, o foco deste trabalho é o controle de formação, enquanto os problemas de desvio de obstáculos e coordenação de tarefas não são abordados nesta etapa do projeto.

Este artigo está organizado da seguinte forma: a seção 2 apresenta a segmentação por cores baseada em imagens omnidirecionais, a seção 3 descreve o controlador de posição e orientação empregado. Na seção 4 são apresentados os experimentos e discutidos os resultados obtidos, e na seção 5, as conclusões e trabalhos futuros.

2 O Processamento das Imagens Omnidirecionais

As imagens omnidirecionais permitem ao robô líder a percepção de todo o seu entorno, facilitando a localização dos seguidores e obstáculos. As posturas dos seguidores são estimadas através da segmentação por cores nestas imagens.

Normalmente, duas cores são usadas no topo de cada robô, e sua orientação pode ser facilmente calculada (J. A. Santos-Victor and Zwaan, 2002). Entretanto, devido à distorção causada por imagens omnidirecionais, optou-se por usar apenas uma cor para cada seguidor. Para duas cores, cada área colorida seria reduzida pela metade. Isto, juntamente com a distorção da imagem (que aumenta à medida que se afasta da câmara), prejudicaria a estimativa da postura de cada seguidor.

Para identificar a posição inicial dos seguidores, o líder constrói um *background* da cena e em seguida ordena aos seguidores que se movam em linha reta por uma curta distância. Usando *Background Subtraction*, os movimentos são segmentados e, então, operações morfológicas são realizadas para reduzir a influência de ruído. Por fim, faz-se a detecção de bordas e a imagem resultante é usada para determinar as regiões de interesse, de onde são extraídas as cores de cada seguidor. Estas cores são usadas também para o processamento seguinte, onde um *software de tracking*, baseado no algoritmo CAMShift (OpenCV), utiliza a segmentação por cores para estimar continuamente a postura dos seguidores. A Figura 1 exhibe o processo de detecção da posição inicial dos seguidores e o *software de tracking* em execução.

As imagens omnidirecionais estão inicialmente representadas no espaço de cores RGB (*Red, Green, Blue*) e sua conversão para o espaço HSV (*Hue, Saturation, Value*) permite concentrar a informação de cor, antes espalhada nos canais R, G e B, apenas no canal *hue* (matiz). A partir daí, a segmentação por cores é feita usando-se apenas este canal, facilitando o processo de *tracking*. Assim, as posições dos seguidores são estimadas usando os centróides das regiões segmentadas. O centro e o tamanho dessas regiões são utilizados pelo algoritmo de *tracking* para gerar

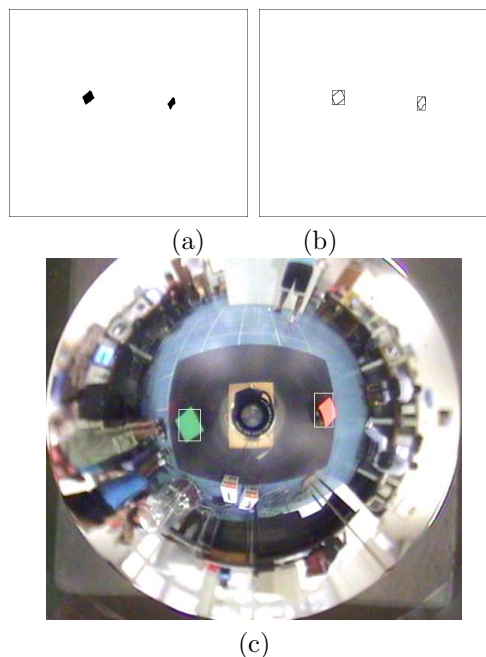


Figura 1: (a) Segmentação dos seguidores, (b) Regiões de interesse e (c) *Software de tracking*.

uma janela de busca na próxima imagem a ser processada, garantindo a rapidez e a robustez necessárias para um bom desempenho. Uma vez que estas estimativas são feitas nas imagens fornecidas pelo sistema de visão do líder, é importante notar que elas são relativas ao referencial do líder.

Para obter todas as posturas, é necessário determinar, também, as orientações dos seguidores. Calcular a orientação utilizando apenas duas imagens consecutivas é impraticável devido à baixa resolução da câmara e aos ruídos causados pela iluminação. Deste modo, foi desenvolvido um algoritmo que estima a orientação de um seguidor baseado na geometria de sua trajetória, como mostra a Figura 2.

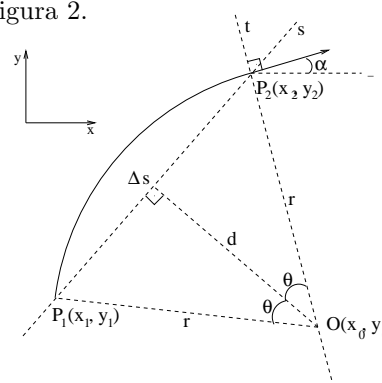


Figura 2: Geometria do movimento considerada para a estimativa da orientação, representada pelo ângulo α .

A postura estimada de cada robô é usada pelo controlador não linear para manter a formação durante a navegação. Para que o controlador seja independente das medições em *pixels* na ima-

gem, as posições dos seguidores são convertidas em metros. Uma maneira de fazer isso e também eliminar a distorção das imagens é através do remapeamento para *bird's eye view* (Vassallo et al., 2004; Pereira et al., 2005). Entretanto, este remapeamento depende de calibração do sistema de visão e representa mais etapas no processamento das imagens.

Neste trabalho, estimou-se uma conjunto de funções polinomiais que convertem distâncias em *pixels* para metros (mundo real) a partir de tabelas de correspondência entre distâncias na imagem e distâncias reais. Outras vantagens são a independência de calibração do sistema de visão e a redução no tempo de cálculo das posições reais.

3 O Controlador

Para uma equipe de robôs móveis (formada por um robô líder e n robôs seguidores) navegar em uma dada formação, é necessário um controle de postura dos robôs seguidores.

3.1 Definição

Um controlador não linear de postura dos seguidores foi definido baseado nas informações obtidas das imagens omnidirecionais capturadas pelo robô líder. Ele leva em consideração as não linearidades que descrevem o sistema. A idéia é obter uma lei de controle que seja estável em qualquer ponto de operação, e não apenas ao redor de um determinado ponto (como no caso de sistemas linearizados). Este controlador deve fornecer os valores desejados das velocidades dos seguidores baseado nos erros de posição e orientação. Para isso, é composto por um controlador de formação, que garante que a equipe entrará na formação especificada, e um segundo controlador, que compensa as velocidades linear e angular do líder. A estabilidade do controlador é provada através do método de Lyapunov (Carelli et al., 2006).

Os comandos gerados são as velocidades linear e angular de referência para os seguidores, que podem ser enviados por diferentes meios de comunicação, como uma interface RS-232 ou um link de rádio.

3.2 Descrição Teórica

Após o processamento das imagens, um vetor contendo todas as coordenadas dos seguidores pode ser definido como na Equação 1.

$$\xi = (\xi_1 \quad \xi_2 \quad \dots \quad \xi_n)^T \quad (1)$$

onde $\xi_i = (x_i \quad y_i)^T$ traduz as coordenadas no plano do chão do i -ésimo seguidor. Para encontrar uma solução genérica, o vetor de coordenadas ξ pode ser considerado como $\rho(\xi)$. Esta abordagem

é interessante porque existem casos onde é necessária a aplicação de uma transformação de coordenadas como, por exemplo, no uso de imagens perspectivas, ou definir parâmetros diretamente associados à formação (por exemplo, parâmetros geométricos, baricentro, etc.). Assim, diferenciando $\rho(\xi)$ no tempo, obtém-se a Equação 2.

$$\dot{\rho} = J(\xi) \dot{\xi} \quad (2)$$

onde $J(\xi)$ é o Jacobiano de ξ .

A partir da Equação 2 é possível definir uma lei de controle de formação dada pela Equação 3. O vetor $\dot{\xi}_{fr}$ representa as velocidades de formação desejadas, isto é, as velocidades, dadas no referencial do líder, que devem ter os seguidores para alcançar a formação.

$$\dot{\xi}_{fr} = J^{-1}(\xi) (\dot{\rho}_d + f_{\tilde{\rho}}(\tilde{\rho})) \quad \text{com } \tilde{\rho} = \rho_d - \rho \quad (3)$$

onde $\tilde{\rho}$ é o vetor de erros de formação dos seguidores (Kelly et al., 2004), ρ_d é o vetor de parâmetros desejados e ρ é o vetor de parâmetros atuais. A função $f_{\tilde{\rho}}(\tilde{\rho})$ é uma função de saturação sobre o erro e definida pela Equação 4.

$$f_{\tilde{\rho}}(\tilde{\rho}) = \text{diag} \left[\frac{k_f(\tilde{\rho})}{a + |\tilde{\rho}_j|} \right] \tilde{\rho} \quad \text{com} \quad (4)$$

$$k_f(\tilde{\rho}) = k_{f1} + k_{f2} \tanh(\tilde{\rho}_j^2).$$

$k_{f1} + k_{f2}$ representa o valor de saturação e a é tal que $\frac{k_{f1}}{a}$ representa o ganho para erros pequenos. Esta função evita a aplicação de velocidades que podem saturar os atuadores dos robôs, ao mesmo tempo que evita oscilações para erros pequenos.

Entretanto, o líder possui suas próprias velocidades linear e angular, definidas em relação a um referencial absoluto. Estas velocidades precisam ser consideradas antes de se calcular as velocidades finais de referência para os seguidores. O vetor $\dot{\xi}_{fr}$ é adicionado, na Equação 5, a $\dot{\xi}_l$ e $\dot{\xi}_\omega$, que são as compensações das velocidades linear e angular do líder, respectivamente. O vetor resultante, $\dot{\xi}_r$, traduz as velocidades que devem ter os seguidores para, ao mesmo tempo, alcançar a formação e compensar as velocidades do líder.

$$\dot{\xi}_r = \dot{\xi}_{fr} + \dot{\xi}_l + \dot{\xi}_\omega \quad (5)$$

Os valores para $\dot{\xi}_l$ e $\dot{\xi}_\omega$ são calculados de tal forma a compensar as velocidades do líder. A Figura 3 mostra um exemplo onde o líder possui velocidades linear (v) e angular (ω) e o i -ésimo seguidor é considerado já na posição desejada $(x_i \quad y_i)^T$.

Uma vez que v e ω são conhecidos, r e r_i , os raios dos círculos descritos pelo líder e seguidor, são dados pelas Equações 6 e 7.

$$r = \frac{v}{\omega} \quad (6)$$

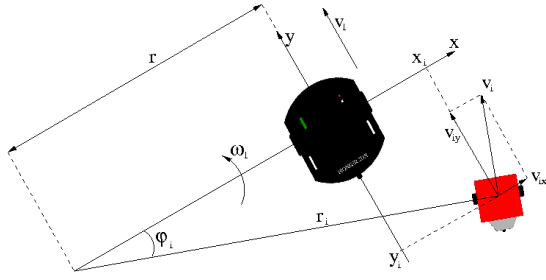


Figura 3: Compensação das velocidades do líder.

$$r_i = \sqrt{(r + x_i)^2 + (y_i)^2} \quad (7)$$

As Equações 8-11 descrevem como é calculada a compensação das velocidades para o i -ésimo seguidor. Estas equações refletem as contribuições tanto de $\dot{\xi}_l$ quanto de ξ_ω na Equação 5.

$$\varphi_i = \arctan\left(\frac{y_i}{r + x_i}\right) \quad (8)$$

$$|v_i| = \omega r_i \quad (9)$$

$$|v_{ix}| = |v_i| \cos\left(\varphi_i + \frac{\pi}{2}\right) \quad (10)$$

$$|v_{iy}| = |v_i| \sin\left(\varphi_i + \frac{\pi}{2}\right) \quad (11)$$

onde v_{ix} e v_{iy} são as componentes da velocidade de compensação do i -ésimo seguidor representada no referencial do líder.

Após o cálculo das compensações, as velocidades linear e angular a serem enviadas ao i -ésimo robô são definidas pelas Equações 12 e 13.

$$\dot{\xi}_{ci} = \|\dot{\xi}_{ri}\| \cos(\tilde{\alpha}_i) \quad (12)$$

$$\omega_{ci} = \dot{\alpha}_{ri} + f_{\tilde{\alpha}}(\tilde{\alpha}_i) + \omega \quad (13)$$

onde $\|\dot{\xi}_{ri}\|$ é a norma da velocidade desejada para o i -ésimo seguidor e $\dot{\alpha}_{ri}$ é a variação de sua orientação no tempo. O termo $\tilde{\alpha}_i = \alpha_{ri} - \alpha_i$ é a diferença entre a orientação desejada, α_{ri} , e a atual, α_i . A função $f_{\tilde{\alpha}}(\tilde{\alpha}_i)$ é uma função de saturação sobre o erro angular definida pela Equação 14.

$$f_{\tilde{\alpha}}(\tilde{\alpha}_i) = k_{\omega 1} \tanh^3(k_{\omega 2} \tilde{\alpha}_i) \quad (14)$$

A função $f_{\tilde{\alpha}}(\tilde{\alpha}_i)$ evita que erros iniciais de orientação causem comandos de velocidade angular elevados, o que comprometeria a estabilidade do controle e submeteria os motores a variações abruptas de tensão.

3.3 Simulações

Antes do teste com os robôs, algumas simulações foram realizadas a fim de avaliar o controlador proposto. Isto permite observar o comportamento do controlador mediante a variação de seus parâmetros, embora não considere a dinâmica da equipe. Sem perda de generalidade, a posição inicial do líder coincide com a origem do referencial

absoluto. A Figura 4 mostra uma simulação onde o líder possui uma velocidade linear constante de 60 mm/s e uma velocidade angular variando de acordo com a função $\omega(t) = 0.05 \tanh(t - b)$, onde b é um parâmetro auxiliar usado para controlar a forma da trajetória desejada.

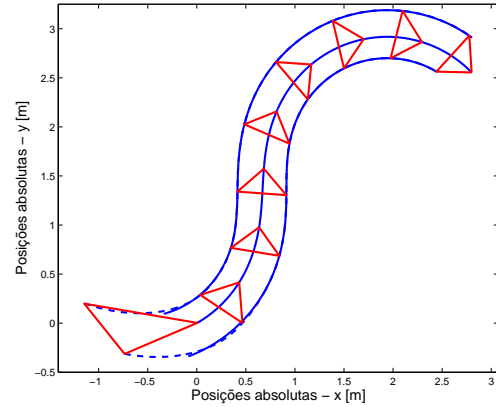


Figura 4: Simulação do controlador não linear.

A linha no centro indica a trajetória do líder. As linhas sólidas externas são as trajetórias desejadas para os seguidores, que devem se posicionar em $x_1 = -0.25$, $y_1 = -0.25$ e $x_2 = 0.25$, $y_2 = -0.25$ relativamente ao líder. Já as linhas tracejadas exibem as trajetórias descritas pelos seguidores, que partiram das posições iniciais $x_{01} = -0.75$, $y_{01} = -0.90$ e $x_{02} = -0.10$, $y_{02} = -0.80$. Os triângulos indicam como a equipe entra em formação. A orientação inicial do líder e dos seguidores foi de $\frac{\pi}{6}$, 0 e 0 rad, respectivamente. As simulações foram feitas usando-se os mesmos parâmetros do controlador utilizado nos experimentos. Os seguidores alcançam a formação, descrevendo a trajetória desejada.

4 Experimentos e Resultados

Os experimentos foram realizados no LAI - Laboratório de Automação Inteligente da UFES, com uma equipe de 3 robôs. O líder é um robô Pioneer 2DX (Pentium II, 266 MHz, 128 MB RAM) e os dois robôs seguidores correspondem a robôs celulares construídos no laboratório e baseados no microcontrolador MSP430F149 (Figura 5). O líder possui um sistema omnidirecional composto por uma câmara colorida, modelo CCS-212 da Samsung, e um espelho hiperbólico. A comunicação entre o líder e os seguidores foi feita através de cabos com comunicação serial, uma vez que um link de rádio usando a tecnologia ZigBee ainda está sendo implantado.

Entre os testes realizados, foi escolhido um experimento para ilustrar o funcionamento da estratégia de controle para a formação do grupo de robôs. As Figuras 7-10 exibem os resultados do experimento para um período de navegação de aproximadamente 1 minuto e 25 segundos.

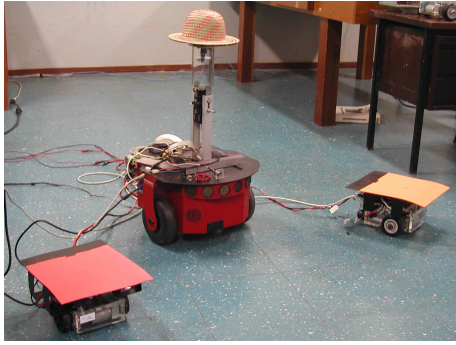


Figura 5: Robôs usados nos experimentos.

4.1 Experimento

O líder se deslocou com velocidade linear de 60 mm/s e angular de 1,5 graus/s. As posições desejadas para os seguidores no referencial do líder são dadas pelas coordenadas $x_{d1} = -0,6$ e $y_{d1} = 0,3$, $x_{d2} = 0,6$ e $y_{d2} = 0,3$, onde o seguidor-1 deverá ficar à esquerda do líder e o seguidor-2 à direita, conforme mostrado na Figura 6.

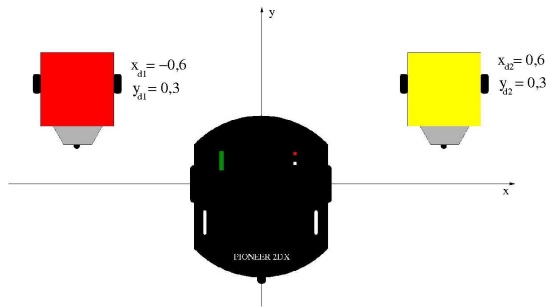


Figura 6: Formação desejada.

Vale ressaltar que, uma vez que os robôs utilizados são não holonômicos, a orientação desejada será sempre 90° . A Figura 7 mostra os erros de posição para o seguidor-1 enquanto a Figura 8 faz o mesmo para o seguidor-2. As linhas sólidas representam os erros em x e as tracejadas, os erros em y . Como pode ser visto, tanto para o seguidor-1 quanto para o seguidor-2, os erros são menores que 5 cm.

As Figuras 9 e 10 mostram as orientações calculadas (linhas pontilhadas) usando o algoritmo desenvolvido, e as orientações filtradas (linhas sólidas) antes de serem passadas ao controlador. Apesar da filtragem (Filtro de Kalman) aplicada às medidas de orientação dos seguidores, ainda assim ocorrem algumas oscilações, já que a orientação do robô é estimada usando-se apenas informação visual, a qual possui ruídos devido à baixa resolução das imagens e variações na iluminação ambiente.

5 Conclusão e Trabalhos Futuros

Este trabalho apresenta uma estratégia de cooperação entre robôs móveis baseada em técnicas de controle não linear e visão omnidirecional. A

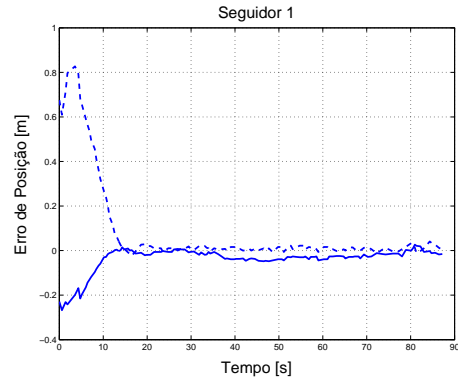


Figura 7: Erros de posição para o seguidor-1.

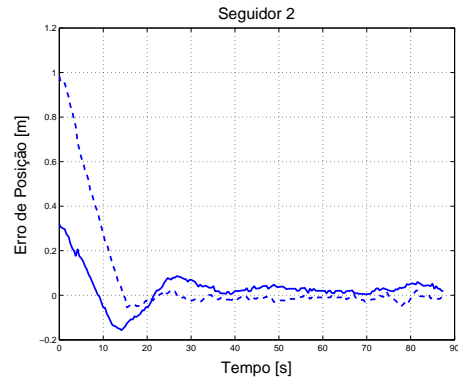


Figura 8: Erros de posição para o seguidor-2.

idéia é controlar uma equipe de robôs simples liderados por aquele que possui maior capacidade de processamento. O robô líder é o responsável pela navegação do grupo e coordenação de tarefas. Para alcançar e manter uma dada formação é utilizado um controlador não linear estável de postura com realimentação visual. Este controlador é o foco deste artigo, não sendo abordados os problemas de detecção e desvio de obstáculos.

O controlador proposto unifica um controlador de formação, projetado para garantir a formação, e um controlador de compensação, que compensa as velocidades do líder. A realimentação visual oferece a vantagem de permitir ao líder localizar todos os robôs ao seu redor capturando apenas uma imagem. Escolhendo cores diferentes para identificar cada seguidor, *Background Subtraction*, operações morfológicas, segmentação por cores e outras técnicas são usadas para localizar os seguidores em relação ao líder. Tal informação visual é utilizada pelo controlador para definir as velocidades de referência para os seguidores atingirem e manterem a formação desejada. Simulações e experimentos foram feitos para avaliar o desempenho do controlador.

Os resultados obtidos são bastante interessantes. A próxima etapa será a inclusão de módulos Zigbee para a comunicação sem fio entre os robôs da equipe.

Depois que o link de rádio entre líder e seguidores estiver funcionando e novos experimentos

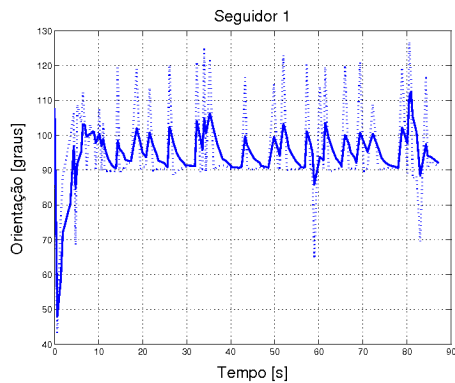


Figura 9: Orientação do seguidor-1.

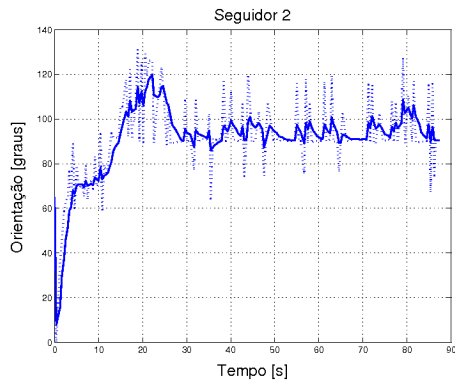


Figura 10: Orientação do seguidor-2.

forem realizados encerrando a etapa de controle de formação, espera-se adicionar algoritmos para detecção e desvio de obstáculos baseados em visão (fluxo ótico e tempo para colisão), sensores de ultra-som ou mesmo sensores *laser*. Estes últimos seriam usados apenas no líder devido, principalmente, ao seu custo. Além disso, imagens omnidirecionais serão utilizadas para um mapeamento global do ambiente. Assim, o líder será capaz de guiar toda a equipe para diferentes lugares e ordenar a execução de tarefas.

Agradecimentos

À FAPES pelo apoio financeiro através do projeto FUNCITEC/30.897.360/2005 e à CAPES e SPU (Argentina) que apoiam o convênio entre UFES e INAUT, através do programa binacional CAPG-BA, e que permitiu a estadia de três meses de Christiano Couto Gava em San Juan, Argentina, onde parte deste trabalho foi realizado.

Referências

Arai, T., Pagello, E. and Parker, L. E. (2002). Guest editorial advances in multirobot systems, *IEEE Transactions on Robotics and Automation* **18**(5): 655–661.

Cao, Y. U., Fukunaga, A. S. and Kahng, A. B. (1997). Cooperative mobile robotics: Antecedents and directions, *Autonomous Robots* (4): 7–27.

Carelli, R., Roberti, F., Vassallo, R. F. and Bastos-Filho, T. F. (2006). Estrategia de control estable de formación para robots móviles, *AADECA 2006 - XX Congreso Argentino de Control Automático*.

Das, A. K., Fierro, R., Kumar, V., Ostrowski, J. P., Spletzer, J. and Taylor, C. J. (2002). A vision-based formation control framework, *IEEE Trans. Robot. Automat.* **18**(5): 813–825.

Desai, J. P., Ostrowski, J. P. and Kumar, V. (2001). Modeling and control of formations of nonholonomic mobile robots, *IEEE Trans. Robot. Automat.* **17**(6): 905–908.

Feddema, J. T., Lewis, C. and Schoenwald, D. A. (2002). Decentralized control of cooperative robotic vehicles: Theory and application, *IEEE Trans. Robot. Automat.* **18**(5): 852–864.

J. A. Santos-Victor and, R. C. and Zwaan, S. V. (2002). Nonlinear visual control of remote cellular robots, *10th Mediterranean Conference on Control and Automation*.

Kelly, R., Carelli, R., Zannatha, J. M. I. and Monroy, C. (2004). Control de una pandilla de robots móviles para el seguimiento de una constelación de puntos objetivo, *VI Congreso Mexicano de Robótica*.

Lewis, M. A. and Tan, K. (1997). High precision formation control of mobile robots using virtual structures, *Autonomous Robots* (4): 387–403.

Matáric, M. J., Nilsson, M. and Simsarin, K. (1995). Cooperative multi-robot box-pushing, *IROS 95*, Vol. 3, pp. 556–561.

Nayar, S. K. (1997). Omnidirectional vision, *8th Int. Symposium on Robotics Research*.

Pereira, F. G., Gava, C. C., Vassallo, R. F. and Sarcinelli-Filho, M. (2005). Calibração de sistemas catadióptricos e detecção da pose de robôs móveis por segmentação de imagens omnidirecionais, *VII SBAI*.

Rus, D., Donald, B. and Jennings, J. (1995). Moving furniture with teams of autonomous robots, *IROS 95*, pp. 235–242.

Vassallo, R. F., ao, L. F. E., Santos-Victor, J. A. and Schneebeli, H. J. A. (2004). Bird's eye view remapping and path following based on omnidirectional vision, *XV CBA*.

Vidal, R., Shakernia, O. and Sastry, S. (2004). Following the flock, *IEEE Robotics & Automation Magazine* pp. 14–20.