

Sistema de Navegação para Robô Móvel Baseada em Visão Omnidirecional e Controlador Fuzzy

Mário Luiz Tronco¹
Arthur José Vieira Porto²

¹*Departamento de Ciências de Computação e Estatística – Instituto de Biociências. Letras e Ciências Exatas -Universidade Estadual Paulista – UNESP*

²*Departamento de Engenharia Mecânica – Escola de Engenharia de São Carlos – Universidade de São Paulo - USP*

mariot@ibilce.unesp.br ajvporto@sc.usp.br

Abstract

Omnidirectional Vision are presented as appropriate to real time decision systems, providing information about the work environment with 360° panoramic images. This work presents a omnidirectional robot navigation system, using fuzzy controller, based on low cost camera (WebCam) and a mirror constructed in polishing metal (conical mirror), under a LINUX Platform.

I. Introdução

Um importante problema relacionado à navegação autônoma de robôs é a necessidade de lidar com a grande quantidade de incertezas inerentes aos ambientes em que o mesmo deve atuar. Devido às imprecisões que estão associadas aos sistemas de sensoramento de ambientes reais e da inexistência de uma formulação matemática para o problema da navegação e guiagem do robô móvel em ambientes interiores reais, torna-se necessário a utilização de uma técnica que utilize uma base de conhecimento humana. Sistemas *fuzzy* apresentam-se como uma técnica adequada para tratar tais problemas.

O presente trabalho descreve o desenvolvimento de um sistema de navegação para um robô móvel, que utiliza um sensor de visão omnidirecional associado a um controlador *fuzzy* para navegar em um ambiente interno controlado. O objetivo principal buscado foi desenvolver um sistema de navegação o qual, a partir de informações de auto-localização fornecidas em tempo real por um sistema de visão e de uma rota de navegação obtida a partir de um mapa do ambiente de trabalho do robô, possibilite que este se desloque do local em que se

encontra até o local de destino, fornecendo as direções de conversão e velocidades adequadas para seu controle durante o percurso da rota de navegação.

II. Navegação de Robôs Móveis

Navegação é a técnica de encontrar a posição, o curso e a distância percorrida por um veículo qualquer. Um sistema de navegação consiste em um dispositivo que responda basicamente três questões: Onde estou? Para onde vou? Como irei até lá? [1].

No contexto de robôs móveis a navegação pode ser definida como “a tarefa de guiar um robô móvel de uma posição inicial para outra posição desejada através de um caminho com certas características”. Navegação neste contexto implica que o problema da localização seja resolvido. A tarefa de navegação assume que a posição do robô seja conhecida o tempo todo com um nível adequado de precisão.

Os principais objetivos, em relação ao contexto da navegação de robôs móveis envolvem:

- Minimizar o tamanho do caminho;
- Minimizar a distância percorrida;
- Minimizar o uso de combustível / eletricidade;
- Maximizar a segurança (distância dos objetos).

A navegação, além da extração de dados do mundo real com uma boa fidelidade, necessita de [2]: **Mapa de localização**: uma representação do ambiente de navegação é necessária para que através de informações vindas dos sensores seja possível determinar a localização do robô móvel no ambiente; **Planejador**: utilizando as informações do mapa será construído o plano de

navegação, que irá traçar a rota a ser percorrida; **Navegador**: utilizará as informações do plano para navegar e cumprir o objetivo esperado.

Robôs móveis navegam e realizam tarefas com certo grau de autonomia. Tarefas para robôs autônomos consistem em planejamento de trajetória, autolocalização, rastreamento visual de marcos do ambiente, detecção e desvio de obstáculos, construção de mapas do ambiente, dentre outras. Os mapas do ambiente são construídos para serem utilizadas em posteriores tarefas de navegação, fornecendo ao robô subsídios para o reconhecimento do ambiente de navegação, planejamento de trajetória, autolocalização e realização de demais tarefas. Para a construção de um mapa, pode-se classificar um ambiente segundo a sua estrutura, podendo ser estruturado ou não estruturado. Essencialmente, em um ambiente estruturado existem restrições geométricas que podem ser utilizadas para a construção de modelos, tal como retas e planos. No caso não estruturado, não existem restrições geométricas que possam ser parametrizadas, ou utilizadas, para representar um mapa, não sendo possível construir mapas usando retas, planos ou outras primitivas geométricas. Neste caso, além de não existirem hipóteses auxiliares, que simplifiquem a construção dos mapas, o movimento do robô é mais geral, isto é, em vez de andar sobre um pavimento plano, pode andar sobre terreno irregular, por isso é mais difícil de ser tratado. Uma outra classificação do tipo de ambiente é se este é externo ou interno, podendo ser estruturado ou não [3], [4], [5]. Ambientes internos como salas, corredores e laboratórios podem ser considerados como estruturados. Na sua maioria contêm retas (verticais e horizontais) e planos, os quais podem ser coletados para a criação de um modelo representativo do ambiente. Assim, no caso de ambientes internos, a navegação pode ser auxiliada por um mapa que descreve a estrutura ou a posição dos objetos no ambiente. Os mapas de ambientes internos podem ser construídos em uma etapa de exploração anterior à etapa de navegação [6], ou podem ser construídos simultaneamente ao problema de navegação [7]. Têm-se também situações onde se realizam navegações sem a construção ou utilização de mapas do ambiente [4], [5].

Tipicamente, existem duas abordagens para a representação de mapas do ambiente: mapas geométricos e topológicos. **Mapas Geométricos**: contêm informações precisas da geometria do ambiente, da posição dos objetos e distâncias entre esses. Como exemplo, citam-se as grades de ocupação mostrado na figura 1a, conexão de polígonos e gráfico de Voronoi; **Mapas Topológicos**: não possuem qualquer informação sobre a geometria do ambiente, sendo representados por nós conectados a nós, como na figura 1b. Nos nós o robô pode tomar certas ações, como virar à esquerda ou à direita, e reconhecer marcos para se localizar no ambiente [2], [8].

O reconhecimento dos marcos visuais dá ao robô uma localização qualitativa no ambiente, obtendo sua posição em termos do quanto está mais próximo ou mais distante do alvo. Na navegação com mapas geométricos a localização é quantitativa, sabendo o robô qual é sua exata posição no ambiente. Algumas abordagens utilizam uma seqüência de imagens como mapa do ambiente, pois esta pode determinar uma rota de navegação [9].

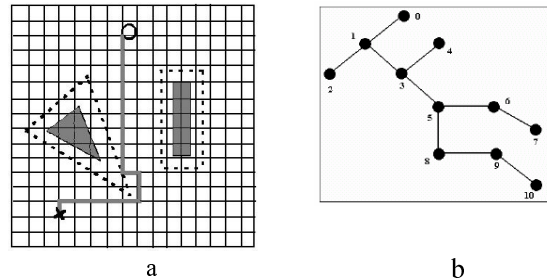


Figura 1. Mapa geométrico baseado em grades de ocupação (a) e Mapa Topológico (b) [10].

Nas aplicações de robótica móvel é interessante ter um campo visual mais abrangente do que o obtido com uma câmera e lente comuns. Um sistema de visão panorâmica ou omnidirecional fornece uma imagem de 360° do ambiente e vem sendo utilizada para a navegação de robôs móveis [9], [11], [12]. O uso desta técnica permite: a aquisição de imagens panorâmicas, a detecção de obstáculos, a verificação de espaço livre para navegação, a não colisão do robô com obstáculos e o reconhecimento de marcos no ambiente.

A navegação, embora auxiliada por técnicas como a de visão omnidirecional, é uma tarefa muito complexa para ser automatizada através do uso de técnicas convencionais de computação. A lógica *fuzzy* é uma técnica que incorpora, em um determinado sistema de controle, a forma humana de pensar. Um controlador *fuzzy* típico pode ser projetado para comportar-se conforme o raciocínio dedutivo, inferindo conclusões baseadas em informações que ele já conhece. De forma simples, pode-se caracterizar a lógica *fuzzy* como um tipo de lógica que reconhece mais do que simples valores de verdadeiro ou falso. Com sistemas que utilizam a lógica *fuzzy*, proposições podem ser representadas com graus de verdade e falsidade.

III. Desenvolvimento Experimental

O Sistema de Navegação descrito no presente trabalho, baseado em visão omnidirecional, foi implementado em uma Plataforma de robô móvel autônomo, do Laboratório de Automação e Computação Evolutiva - LACE – do Departamento de Ciências de Computação e Estatística - DCCE – IBILCE – UNESP.

Este veículo tem como controlador de bordo um Microcomputador Pentium 200MHz com Sistema Operacional LINUX, o qual, via Rádio Modem, conecta-se a um Microcomputador Gerente, como mostrado na Figura 2, a seguir.

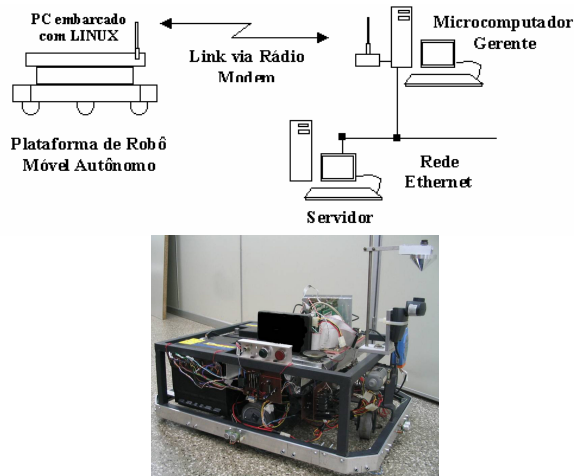


Figura 2. Plataforma de Robô Autônomo do LACE – DCCE – IBILCE – UNESP.

O sistema de navegação desenvolvido proporciona ao robô móvel a capacidade de deslocar-se entre dois lugares, em um ambiente interno estruturado. O ambiente onde o robô móvel navega é previamente conhecido e representado através de um mapa, que é utilizado para determinar as rotas de navegação. Os módulos que compõem o sistema de navegação são integrados conforme mostrado na figura 3.

O módulo de mapeamento e determinação de rotas é responsável por interpretar o mapa do ambiente e determinar a rota de navegação e as direções de conversão do robô a partir da sua auto-localização fornecida pelo sistema de visão; o módulo de controle *fuzzy* é responsável por controlar a direção e a velocidade do robô durante seu percurso na rota determinada.

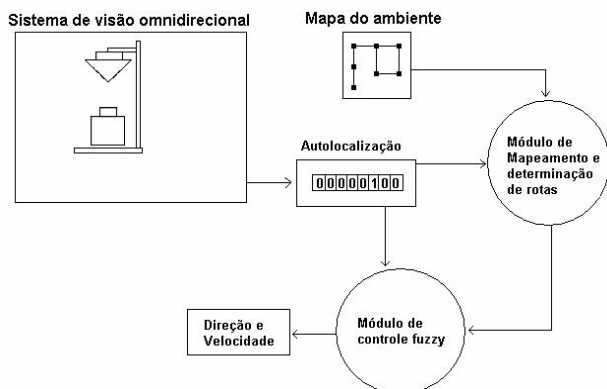


Figura 3: Módulos do Sistema de Navegação.

O sistema de visão omnidirecional é composto por quatro módulos: Módulo de Captura, o qual faz a aquisição da imagem omnidirecional e a converte em uma imagem panorâmica, Módulo de Pré-processamento, responsável pela detecção de bordas e binarização da imagem panorâmica, Módulo Gerador de Padrões, responsável por associar a imagem capturada a um lugar e posição previamente definidos para processamento pela Rede Neural e Módulo de Localização que usa uma Rede Neural para gerar as informações necessárias à navegação (posição e direção do robô). A figura 4, a seguir, mostra um exemplo de imagem omnidirecional e de imagens panorâmicas processadas pelo sistema.

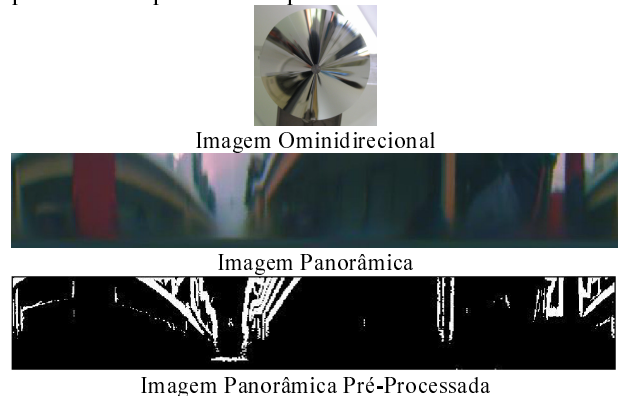


Figura 4. Imagens processadas pelo sistema de visão omnidirecional.

O módulo de localização tem como objetivo associar as imagens geradas pelo módulo de pré-processamento com uma posição pré-determinada em um ambiente interno no qual o robô desloca-se. Para tal, uma Rede Neural é treinada com imagens capturadas em lugares pré-determinados desse ambiente e usada nesse módulo para gerar informações de auto-localização do robô móvel em relação a esses lugares. A Rede Neural utilizada possui 200 neurônios na camada de entrada (a imagem gerada pelo módulo de pré-processamento possui 50 pixels de largura por 4 pixels de altura). Na camada de saída, foi adotada a associação de um neurônio com cada local a ser reconhecido. Entretanto, a orientação do robô também deve ser determinada. Assim, para cada lugar existem quatro neurônios de saída sendo que cada um dos quatro é uma direção em relação a um referencial fixo no ambiente. Em um ambiente com N lugares diferentes para serem reconhecidos, a saída da RNA é composta por $(4 \times N)$ neurônios. Foram utilizados conjuntos de treinamento para cada lugar (15 imagens panorâmicas representando cada lugar) representado no Ambiente de Navegação do Robô. Estas imagens foram capturas utilizando o conjunto espelho/câmera e pré-processadas antes de serem apresentadas à Rede Neural.

Para a construção do mapa do ambiente de navegação para o robô móvel utilizou-se a técnica de mapeamento topológico, no qual um mapa global que define todo o ambiente de navegação do robô é representado através de um grafo. Os vértices do grafo representam os lugares estratégicos para os quais o sistema de visão omnidirecional foi treinado a reconhecer. Os lugares estratégicos para a navegação correspondem aos possíveis locais de conversão de direção do robô móvel e também os possíveis locais de destino final das rotas de navegação. As arestas do grafo representam as relações de adjacência entre os lugares, ou seja, um lugar é adjacente a outro se existe um caminho livre entre eles.

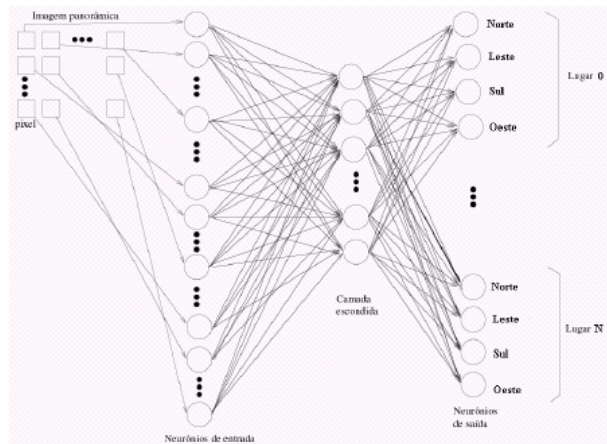


Figura 5. Rede Neural do Módulo de Localização.

A figura 6 é uma representação gráfica de um ambiente com 10 lugares treinados para serem reconhecidos pelo sistema de visão.

Para que o robô móvel, a partir do lugar em que ele se encontra, possa se deslocar até um lugar de destino, percorrendo sempre o menor caminho possível, foi utilizada uma estratégia de determinação de rotas ótimas usando o algoritmo de Dijkstra.

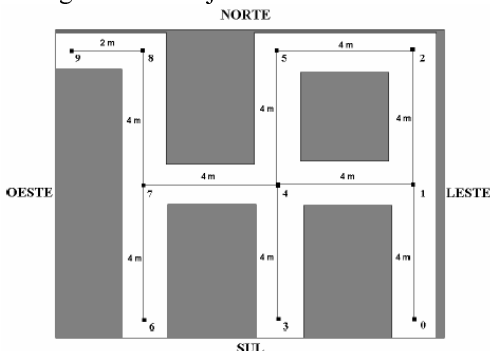


Figura 6. Representação Gráfica do Mapa do Ambiente.

O módulo de mapeamento e determinação de rotas gera, através deste algoritmo, um vetor com uma

seqüência de lugares que o robô móvel deve percorrer para alcançar o destino final. No entanto, além de conhecer essa seqüência de lugares, o robô móvel deve saber qual direção tomar em cada lugar da rota de navegação. Para isso considerou-se as direções das arestas que foram definidas no grafo e armazenadas em uma matriz de adjacência. No ambiente de navegação foi definida uma direção de referência, que passou a ser considerada como sendo o norte geográfico do ambiente; as demais direções leste, sul e oeste foram definidas a partir da direção norte adotada. Essas mesmas direções são adotadas tanto para a construção do mapa do ambiente quanto para o treinamento da rede neural do sistema de visão omnidirecional. A tabela 1 mostra as relações das direções das arestas com o movimento que o robô deve executar.

Próxima Anterior	Norte	Sul	Leste	Oeste
Norte	Reto em Frente	Reto para trás	Direita	Esquerda
Sul	Reto para trás	Reto em frente	Esquerda	Direita
Leste	Esquerda	Direita	Reto em Frente	Reto para trás
Oeste	Direita	Esquerda	Reto para trás	Reto em frente

Tabela 1. Direções de conversão considerando as arestas anterior e próxima.

Módulo de controle fuzzy

O módulo de controle fuzzy implementa o subsistema que controla a direção e a velocidade do robô móvel durante o percurso da rota de navegação, considerando-se que o robô só deve efetuar uma conversão quando estiver muito próximo do lugar em que ele deve mudar de direção, e também a velocidade do robô móvel deve ser adequada para cada trecho da rota de navegação. A figura 7 mostra a integração dos componentes do controlador fuzzy.

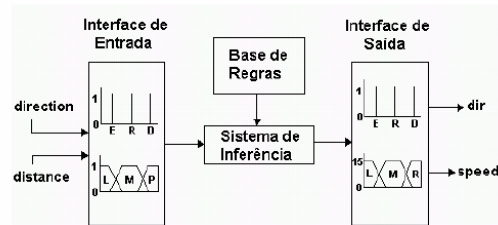


Figura 7: Controlador fuzzy

O módulo de controle fuzzy utiliza duas variáveis de entrada, *distance* e *direction* e fornece duas variáveis de saída, *dir* e *speed*. O número de funções que abrangem o universo de discurso da variável, os tipos de funções de pertinência (trapezoidal e *singleton fuzzy*) utilizados e o grau de sobreposição das funções, foram definidos a partir

da observação do comportamento das variáveis de entrada e também pelas ações propostas para a realização da tarefa de navegação do robô. A variável *distance* representa a distância relativa do robô móvel até o lugar de destino da aresta que ele está percorrendo (L – longe: 0.0 a 0.75, P - perto 0.6 a 0.95 e MP - muito perto: 0.8 a 1.0). O valor desta variável, que é adimensional, é proporcional à distância do robô ao próximo lugar de destino no grafo representando sua rota de deslocamento. A variável *direction* denota a direção que o robô móvel deve tomar ao atingir o lugar de destino da aresta que ele está percorrendo (E – esquerda: 1.0, R – reto: 2.0 e D – direita: 3.0). A variável de saída *speed* indica qual deve ser a velocidade do robô para determinado trecho da rota. Essa velocidade é inferida a partir da direção de conversão e da distância do robô em relação ao local de conversão (SLW – lento: 0.0 a 0.06, MD – médio: 0.03 a 0.12 e FST – rápido: 0.09 a 0.15). A conversão da variável de saída *fuzzy* para um valor discreto é feita pela interface de saída do módulo de controle *fuzzy* utilizando-se o método do centróide que calcula o ponto que divide a área composta em duas partes iguais. A velocidade de saída encontra-se na faixa $0.0 \text{ m/s} \leq \text{speed} \leq 0.15 \text{ m/s}$. A variável de saída *dir* indica qual deve ser o posicionamento das rodas do robô para cada trecho da rota, indicando as direções de conversão direita ou esquerda somente quando a distância do robô em relação ao local de conversão for muito pequena (E – esquerda: 1.0, R – reto: 2.0 e D – direita: 3.0). A definição das regras *fuzzy* baseou-se nos objetivos propostos para o controlador, ou seja, de que o robô móvel só deve efetuar uma curva quando estiver a uma distância muito próxima do lugar de conversão, uma vez que o robô pode identificar este lugar mesmo estando a uma distância considerável do ponto ideal de conversão. O robô também deve mover-se de forma mais lenta quando estiver próximo aos lugares em que deve fazer uma conversão ou próximo aos destinos finais da rota de navegação. Com base nestas metas, as tabelas 2 e 3 apresentam respectivamente numa forma condensada, as regras utilizadas para determinar o correto posicionamento das rodas do robô e as regras utilizadas para determinar a velocidade adequada do robô para cada trecho da rota de navegação.

distance / direction	E - Esquerda	R - Reto	D - Direita
L - longe	R	R	R
P - perto	R	R	R
MP - muito perto	E	R	D

Tabela 2. Regras para a direção do robô.

distance / direction	Esquerda	Reto	Direita
L - longe	FST	FST	FST
P - perto	MD	FST	MD
MP - muito perto	SLW	FST	SLW

Tabela 3. Regras para a velocidade do robô.

IV. Resultados

A figura 8, a seguir, mostra o gráfico com a variação dos valores dos neurônios do Módulo de Localização, correspondentes aos lugares de destino em cada trecho da rota de navegação em um teste realizado no ambiente do LACE.

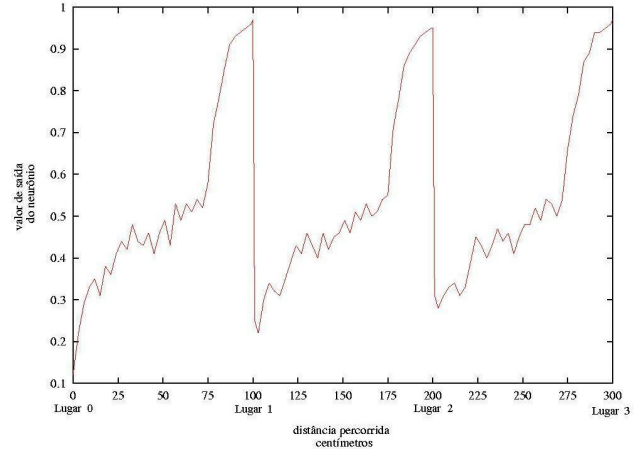


Figura 8. Saída dos Neurônios do Módulo de Localização.

Na figura anterior, pode-se notar que os valores de saída variaram entre 0 e 1, assumindo valores menores quando o robô está mais próximo do lugar de origem e aumentando gradativamente à medida que se aproxima do lugar de destino de cada trecho da rota. Estes valores são observados nas saídas da Rede Neural relativas à posição do ambiente que está sendo monitorada (são quatro saídas – quatro neurônios – para representar cada lugar do ambiente). O mapa topológico do ambiente do LACE foi construído para testar o deslocamento do robô pelo ambiente do laboratório. O ambiente foi mapeado com 6 vértices igualmente espaçados, como mostrado na figura 9.

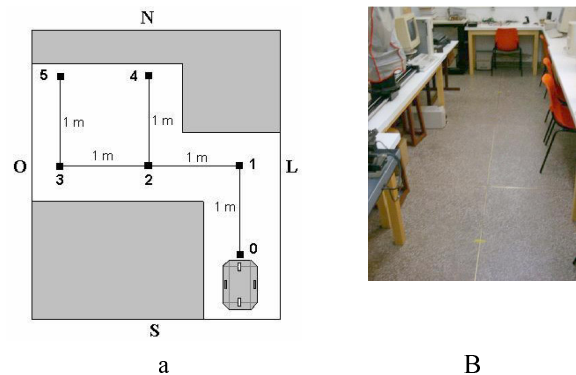


Figura 9. Representação gráfica (a) do mapa (b)

A figura 10, a seguir, mostra a saída do controlador *fuzzy* para a variável *speed*, que denota a velocidade do robô no trajeto de rota testado.

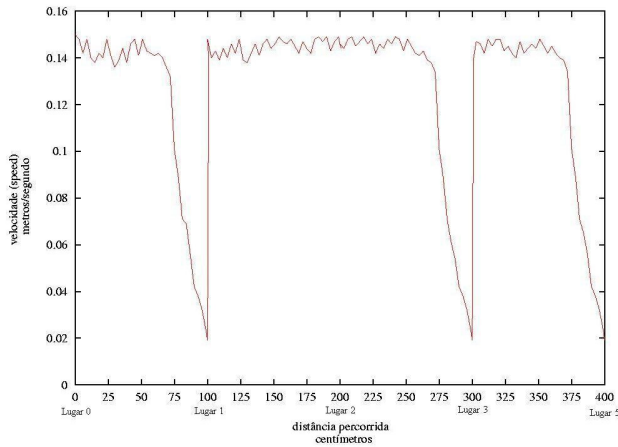


Figura 10. Saída do controlador fuzzy para a variável *speed*

Da a figura 10, pode-se perceber que o robô inicia o deslocamento pela rota com velocidade muito próxima da máxima, que foi definida nas funções de pertinência para a variável *speed*, mantendo esta velocidade enquanto se encontra distante do lugar 1, onde deve executar uma conversão à esquerda. Ao se aproximar do lugar 1 a velocidade é reduzida rapidamente, ficando muito baixa no lugar em que o robô efetua a conversão. Após a conversão a velocidade é restabelecida para a máxima e mantida assim enquanto o robô se aproxima do próximo lugar (lugar 2). Neste lugar, percebe-se que o controlador não reduz a velocidade do robô na aproximação, pois ele não efetua uma conversão de direção neste local. A velocidade é mantida alta até a aproximação do lugar 3 onde a variação da velocidade se comporta como no lugar 1, pois o robô executa uma conversão à direita neste local. O mesmo ocorre para o lugar 5 que corresponde ao destino final da rota de navegação.

V. Conclusões

A técnica de visão omnidirecional mostrou-se adequada para a aplicação de navegação do robô, fornecendo sua autolocalização em tempo real para as ações de controle do robô. O valor deste tempo foi suficientemente pequeno quando comparado ao tempo de deslocamento do robô de um lugar ao próximo lugar em sua rota de tráfego. O comportamento do valor da saída dos neurônios possibilitou sua utilização no controlador *fuzzy*, fornecendo uma relação de distância do robô aos lugares. A estratégia de mapeamento topológico, utilizada para construir o mapa do ambiente de navegação, em conjunto com o módulo de mapeamento e determinação de rotas funcionou corretamente em todas as situações

testadas, mostrando ser uma solução adequada para o ambiente de navegação proposto. O módulo de controle *fuzzy* funcionou corretamente nos testes, fornecendo a direção e velocidade adequada para o robô em cada trecho da rota de navegação.

O Robô do LACE conta com diversos níveis de sensores (de ultra-som, ópticos reflexivos, etc) os quais vem sendo utilizados, em uma estratégia de fusão sensorial, para dotá-lo de maior robustez nas atividades de controle de deslocamento. Resultados experimentais utilizando esta técnica vem sendo obtidos e analisados para validar a nova estratégia de controle adotada.

VI. Agradecimentos

Os autores agradecem à FUNDUNESP, pelo apoio financeiro de parte dos módulos desenvolvidos e à MANET pelo trabalho colaborativo proporcionado.

Referências

- [1] Borenstein, J.; Koren, Y. Sensors for Mobile Robots Navigation, *IEEE Journal of Robotics and Automation*, Vol. RA-4, No. 2, p. 213-218, 1996.
- [2] Thrun, S.; Bucken A. Learning Maps for Indoor Mobile Robot Navigation, Technical Report CMU-CS-96-121, Carnegie Mellon University, School of Computer Science, Pittsburgh, 1996.
- [3] Giachetti, A.; Campani, M.; Torre, V. The Use of Optical Flow for Road Navigation, *IEEE Transactions on Robotics and Automation*, vol. 14, nº 1, p.34-48, 1998.
- [4] Goldberg, S. B.; Maione, M. W.; Matthies, L. Stereo Vision and robot navigation software for planetary exploration, *IEEE Aerospace Conference Proceedings*, vol.5 p. 2025-2036, 2002.
- [5] Van der Zwaan, S.; Santos-Victor, J. A. Real-Time Vision Based station keeping for Underwater Robots. *MTS/IEEE Conference on Exhibition OCEANS*. Vol.2, p. 1058-1065. Nov. 2001
- [6] Yamauchi, B. A Frontier-Based Approach for Autonomous Exploration, *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 146-151, 1997.
- [7] Rekleitis, I. M.; Dujmovic, V. Efficient Topological Exploration, *Proceedings of the IEEE International Conference on Robotics and Automation*, Detroit, Michigan, p. 676-681, 1999.
- [8] Thrun, S. Robot Mapping: A Survey, CMU-CS-02-111, Feb. 2002, http://www-2.cs.cmu.edu/afs/cs.cmu.edu/user/thrun/public_html.
- [9] Matsumoto, Y.; Ikeda, K.; Inaba, M.; Inoue, H. Exploration and Navigation in Corridor Environment Based on Omni-View Sequence, *Proceeding of the IEEE International Conference on Intelligent Robots and Systems*, p. 1505-1510, 2000.
- [10] Duckett, T.; Saffioti, A. Building Globally Consistent Gridmaps from Topologies, Center for Applied Autonomous Sensor Systems Department of Technology, The University Örebro, SE-70182, Sweden, 2000.
- [11] Gaspar, J.; Santos-Victor, J. Visual Path Following with a Catadioptric Panoramic Camera, *International Symposium on Intelligent Robotic System*, SIRS'99, Coimbra, Portugal, 1999.
- [12] Gaspar, J.; Decó, C. C. G.; Okamoto, J.; Santos-Victor, J. Constant Resolution Omnidirectional Cameras, *OMNIVIS'02 Workshop on Omni-directional Vision*, Copenhagen, 2002.