

SISTEMA DE NAVEGAÇÃO PARA ROBÔ MÓVEL BASEADO EM MAPEAMENTO DE AMBIENTES E CONTROLADOR FUZZY

MÁRIO LUIZ TRONCO

Laboratório de Automação e Computação Evolutiva - LACE, Departamento de Ciências de Computação e Estatística - DCCE, Universidade Estadual Paulista - UNESP
Rua Cristóvão Colombo, 2265 – Jardim Nazareth - CEP 15054 – 000 – São José do Rio Preto - SP
E-mail: mariot@ibilce.unesp.br

ARTHUR JOSÉ VIEIRA PORTO

Laboratório de Simulação de Sistemas a Eventos Discretos, Departamento de Engenharia Mecânica – Escola de Engenharia de São Carlos, Universidade de São Paulo - USP
Rua Trabalhador São-Carlense, 400 – CEP 13566-590, São Carlos - SP
E-mail: ajvporto@sc.usp.br

Abstract— Autonomous robots must be able to learn and maintain models of their environments. In this context, the present work considers techniques for the classification and extraction of features from images, using artificial neural networks. This techniques are used in the LACE (Automation and Evolutive Computer Laboratory) mobile robot mapping and localization system. This is based on a sensorial sub-system composed by ultrasound sensors and a omnivision system composed by a camera and a conical mirror. The system uses a hierarchical neural network to do the classification, structured in two layers, reason and intuition, and is trained to classify each place explored by the robot. A fuzzy system, used to generate variables associated with speed and direction of the robot is also presented. Experimental results are presented.

Keywords — Computer Vision, Hierarchical Neural Network, Mobile Robot Vision System

Resumo— Robôs autônomos devem ser capazes de aprender e manter modelos dos ambiente explorados. Neste contexto, o presente trabalho apresenta técnicas para a classificação e extração de atributos de imagens, usando redes neurais artificiais. Estas técnicas são utilizadas no sistema de mapeamento e localização do robô móvel do LACE (Laboratório de Automação e Computação Evolutiva). Este sistema é baseado em um sub-sistema sensorial composto por sensores de ultra-som e um sistema de visão omnidirecional. O sistema usa uma rede neural hierárquica para realizar a classificação, estruturada em duas camadas: razão e intuição, e é treinada para classificar cada local explorado pelo robô. Um sistema fuzzy, usado para gerar variáveis associadas à velocidade e direção do robô é também apresentado. Resultados experimentais são apresentados e analisados.

Palavras-chave — Visão Computacional, Rede Neural Hierárquica, Sistema de Visão para Robô Móvel

1. Introdução

Navegação baseada em mapas requer um processo de reconhecimento e análise de alto nível a fim de interpretar o mapa e estabelecer sua correspondência com o mundo real. Além disso, o problema de adquirir modelos é complexo e difícil de ser resolvido devido a limitações práticas sobre a habilidade dos robôs de aprender e usar modelos precisos. Estas limitações surgem a partir de problemas inerentes aos próprios sensores e ao ambiente. Classicamente, a construção de mapas segue duas abordagens: geométrica e topológica. A abordagem adotada neste trabalho é a topológica, a qual é definida por Kuipers (KUIPERS and BYUN, 1991). De acordo com ela, o ambiente é representado por um número de lugares distintivos, e de modo que o robô possa se locomover de um lugar a outro. Em outras palavras, o ambiente é representado por um grafo, onde os lugares identificados for-

mam o conjunto de vértices ou nós, e todas as passagens entre cada nó formam o conjunto de arestas.

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 a lógica fuzzy, proposições podem ser representadas com graus de verdade e falsidade.

No presente trabalho são apresentados um método de mapeamento e um módulo fuzzy usados pelo robô móvel do Laboratório de Automação e Computação Evolutiva (LACE) do Departamento de Ciências de Computação e Estatística – DCCE – UNESP.

2. Sistema de Visão Omnidirecional do Robô Móvel do LACE

O robô móvel do LACE é equipado com um Módulo de Visão Omnidirecional, no qual imagens omnidirecionais são capturadas por uma câmera com seu foco alinhado a um espelho cônico. Estas imagens são retificadas e transformadas em imagens panorâmicas da cena. Elas são então pré-processadas e usadas pelos procedimentos de classificação e caracterização de nós. As equações e a metodologia adotada na captura e processamento das imagens estão definidas em Spacek (SPACEK, 2003). A figura 1 mostra esquematicamente o sistema de visão omnidirecional do robô.

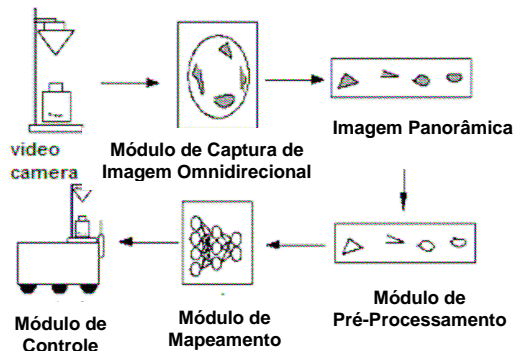
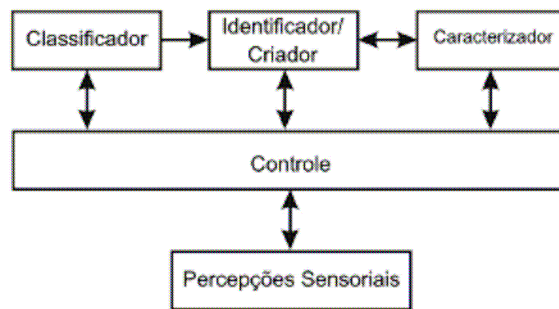


Figura 1. Sistema de Visão Omnidirecional do Robô Móvel do LACE.

3. Mapeamento Topológico

O método de mapeamento de ambientes proposto neste trabalho tem como objetivo construir um modelo topológico de ambientes interiores explorados pelo robô móvel do LACE. Para percepção do ambiente o robô utiliza seu sistema sensorial formado por sensores de ultra-som e um sistema de visão omnidirecional, sendo tais informações fornecidas como dados de entrada ao sistema de mapeamento, o qual é composto por três módulos: classificador, identificador/criador e caracterizador de nós. A principal tarefa do módulo classificador é identificar cada lugar visitado pelo robô, classificando-os dentre quatro classes pré-definidas, e assim criar os nós do mapa com suas respectivas relações de adjacência. Uma rede neural hierárquica estruturada em duas camadas, *razão* e *intuição*, é usada como principal ferramenta do módulo classificador. A tarefa de identificação e criação de um novo nó é realizada por um procedimento que recebe a informação sobre a classe do lugar e cria um novo nó da mesma classe. A Figura 2 ilustra a relação entre os módulos do sistema de mapeamento.



3.1. Módulo Classificador

A função do módulo classificador é classificar os lugares explorados pelo robô dentre quatro classes pré-definidas: corredor, porta, intersecção e sala; e assim passar esta informação ao módulo criador, o qual efetivamente cria e insere um novo nó no mapa topológico. O objetivo da classificação é atribuir parâmetros a cada classe para que, com base neles, o robô saiba quais decisões tomar durante a navegação. Dessa maneira, as decisões de navegação do robô podem ser definidas como uma máquina de estados, onde cada estado (classe) está associado a um conjunto de possíveis ações. A Figura 3 mostra este procedimento.

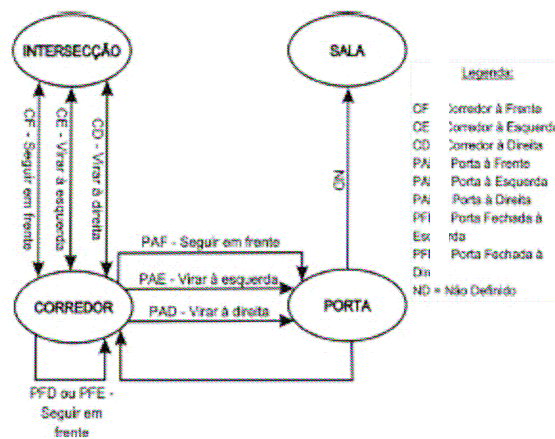


Figura 3. Diagrama em máquina de estados das decisões de navegação do robô.

Para implementar a tarefa de classificação usou-se uma Rede Neural Artificial Hierárquica (RNAH) treinada para reconhecer as classes de lugares definidas. A RNAH é formada por duas camadas, *razão* e *intuição*, como ilustrado na figura 4. A RNAH recebe como entrada os dados lidos pelo sistema sensorial do robô: medidas de distância e imagens do ambiente fornecidas pelos sonares e sistema de visão omnidirecional, respectivamente.

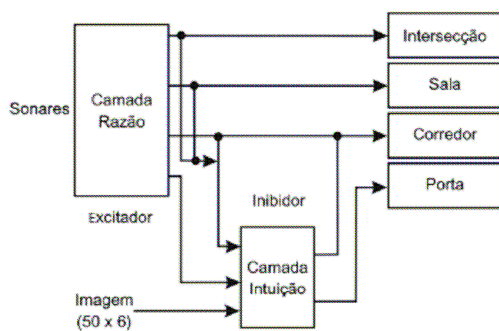


Figura 4. Rede Neural Hierárquica.

A rede razão recebe os dados lidos pelos sonares e com base neles é capaz de identificar a classe *corredor*, quando o robô navega através deste, e a classe *intersecção*, quando detecta o encontro de dois ou mais corredores de modo que o robô possa eventualmente mudar a direção do seu percurso. As informações de distância são suficientes para identificar as referidas classes nas condições descritas. Quando o robô se locomover através de um corredor e seus sonares detectarem a existência de uma abertura lateral, esta pode consistir de uma porta ou um corredor. Neste caso as informações de distância não são suficientes para identificar a classe correta. É neste momento que a rede intuição é ativada.

Nesta situação, um dos neurônios da saída da rede razão é ativado, o qual consiste no neurônio excitador da rede intuição. Esta executa então seu procedimento de classificação utilizando imagens do local atual com o objetivo de identificar a classe correta: *porta* ou *corredor*. Para isso, a segunda camada da RNAH é treinada com imagens destas duas classes de lugares para adquirir a capacidade de distingui-las.

3.2 Treinamento da RNAH

Para identificar as classes definidas, ambas as camadas da RNAH precisam ser treinadas para adquirir tal capacidade. Foram definidos então parâmetros para cada classe, os quais as caracterizam, tornando-as diferentes das demais. Tais parâmetros são levados em consideração durante a construção do conjunto de padrões de treinamento. O conjunto de treinamento fornecido à camada *razão* da RNAH permite que a mesma classifique com certo grau de certeza as classes cujos parâmetros que as caracterizam são bem definidos e modelados com base nas distâncias lidas pelos sonares. As classes onde isso não acontece são caracterizadas por suas imagens, e é responsabilidade da segunda rede identificá-las, a qual é ativada sempre que houver qualquer dúvida no processo de reconhecimento da primeira camada. Por exemplo, um

corredor é definido como um lugar limitado continuamente por obstáculos em ambos os lados do robô.

Uma intersecção é um local onde dois ou mais corredores se interceptam, de modo que o robô possa eventualmente alterar a direção do seu percurso. Essas duas classes são então modeladas levando-se em consideração tais características e as informações disponíveis para a construção dos modelos: leitura das distâncias dos sonares. No caso da rede intuição, o conjunto de treinamento é formado por diferentes imagens de portas e corredores, com o objetivo de capacitar a rede a distingui-las quando estes forem encontrados durante a etapa de mapeamento.

A arquitetura da rede neural adotada é a Perceptron Multicamadas (MPL), que são redes adequadas para a tarefa de classificação. Os sonares localizados nas laterais do robô se movem na horizontal enquanto realizam a leitura das distâncias. O resultado é a criação de um vetor de números reais, onde cada número representa a distância do robô em relação a um objeto, calculada de acordo com o ângulo formado entre a respectiva direção de leitura e a posição central de leitura. Já o sensor dianteiro se move tanto na horizontal quanto na vertical, resultando em uma matriz de números reais. Dessa maneira, cada neurônio da camada de entrada da rede *razão* recebe o valor armazenado em uma das posições do vetor ou matriz de leituras. Assim esta camada tem tantos neurônios quanto o número de leituras realizadas pelos três sonares. Sua camada de saída é composta por quatro neurônios: os três primeiros correspondem às classes *sala*, *intersecção* e *corredor*, enquanto o último consiste no neurônio excitador da rede *intuição*.

A rede intuição recebe como entrada imagens dos locais a serem classificados. Dessa maneira, o número de neurônios de sua camada de entrada é igual à resolução das imagens, ou seja, o número total de pixels utilizados para representar as imagens. Este número deve ser "grande" o suficiente para garantir a qualidade no processo de classificação, e "pequeno" o suficiente para garantir eficiência durante as fases de treinamento e validação da rede, etapas estas que demandam o maior tempo de execução.

3.3 Módulo Caracterizador

A função do módulo *caracterizador* de nós do sistema de mapeamento é caracterizar os nós identificados pelo módulo classificador, a fim de torná-los únicos e distintos dos demais nós pertencentes à mesma classe. O objetivo é criar marcos naturais para os nós a partir de cenas de um conjunto de imagens, as quais serão usadas como marcos visuais, o que difere da abordagem onde marcos são definidos como objetos individuais da cena.

Na abordagem adotada neste artigo, os marcos são definidos por vetores de atributos e invariantes de

momento afim extraídos a partir das imagens dos nós. A técnica usada para extrair os vetores de atributos e diminuir o tamanho dos mesmos é a PCA (Principal Component Analysis) (DUDA, 1973), cuja aplicação resulta em uma representação de menor dimensão da imagem, levando em consideração a variância dos atributos. As equações usadas para calcular os momentos afins foram derivadas por Zitová (ZITOVÀ et. al, 1999), as quais são invariantes sobre transformações afins geral.

A abordagem proposta para implementação da PCA é definida como segue. Considere a imagem I fornecida pelo sistema de visão omnidirecional, representada por uma matriz de dimensão $m \times n$, de modo que cada elemento representa o nível de intensidade de cinza naquele ponto. A imagem pode ser representada por um vetor através da leitura, coluna a coluna, da matriz da imagem e armazenando cada pixel num vetor coluna. Assim,

$$x(l) = I(i, j), \quad (1)$$

para $i = 1, \dots, n, j = 1, \dots, m$ e $l = i + (j \times 1) \times m$. O tamanho do vetor de atributos é dado por $d = m \times n$. Considerando t padrões de treinamento conhecidos, x_1, x_2, \dots, x_t , o conjunto de treinamento pode ser visto como uma matriz, onde cada coluna contém um padrão de treinamento,

$$X = \begin{bmatrix} x_1(1) & x_2(1) & \dots & x_t(1) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(d) & x_2(d) & \dots & x_t(d) \end{bmatrix} \quad (2)$$

A matriz de covariância Σ x da matriz de treinamento X , pode ser definida como:

$$\Sigma = (X - \mu)(X - \mu)^T, \quad (3)$$

onde μ é a matriz média de dimensão $d \times t$, e cada coluna da matriz contém o valor esperado do padrão X .

$$\mu(i, j) = \frac{1}{t} \sum_{j=1}^t (X(i, j)), \quad (4)$$

Dada a matriz de covariância Σ x, estima-se seus auto-vetores e auto-valores de maneira que:

λ_i e v_i , para $i = 1, \dots, d$, são os auto-valores e auto-vetores de Σx , respectivamente. Os autovalores indicam a relevância dos autovetores. O conjunto ordenado de autovetores compõe a matriz de transformação H a seguir:

$$Y = H^T X, \quad (5)$$

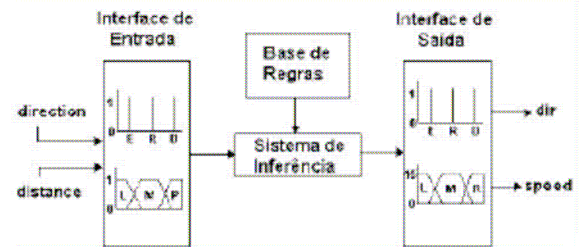
e H^T é a matriz transposta de H .

A matriz de novos atributos Y obtida não apresenta redução na sua dimensão, sendo somente uma mudança de base no espaço de atributos. Entretanto, a matriz de covariância H pode ser construída somente com os autovetores que possuem os maiores autova-

lores. Escolhendo-se k autovalores, $k < d$, a dimensão do vetor de atributos é reduzida para uma representação k -dimensional.

4. Módulo 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 5 mostra a integração dos componentes do controlador fuzzy e suas regras.



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

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

Figura 5. Módulo 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). 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.

5. Resultados Experimentais

As tabelas 1 e 2, a seguir, mostram os resultados experimentais obtidos com o Módulo de Mapeamento.

Tabela 1. Taxa de Classificação Geral das Camadas Razão e Intuição.

Camada	Classificação Geral	Padrões Desconhecidos	Padrões Alterados
Razão	100%	100%	–
Intuição	94%	90%	100%

Tabela 2. Classificação de Corredor e Porta na camada Intuição.

Classe	Classificação Correta	Classificação Incorreta	Classificação Correta com baixo Valor de Ativação
Porta	93%	7%	7%
Corredor	95%	5%	0%

A figura 6 mostra a saída do controlador fuzzy para a variável speed, que denota a velocidade do robô no trajeto de rota testado.

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.

6. Conclusões

Este trabalho contemplou uma técnica de modelagem de ambientes para navegação do robô móvel do Laboratório de Automação e Computação Evolutiva – LACE. A arquitetura de rede neural utilizada mostrou-se adequada para a implementação do sistema de mapeamento do robô. Os resultados obtidos no teste da camada intuição foram também satisfatórios, pois a mesma obteve em geral uma taxa de acertos de 94%. Assim, esta camada alcançou o objetivo para a qual foi criada: resolver possíveis conflitos no processo de classificação da primeira rede. A técnica de visão omnidirecional utilizada no sistema de navegação do robô móvel mostrou-se adequada, fornecendo sua autolocalização em tempo hábil para as ações de controle do mesmo. 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.

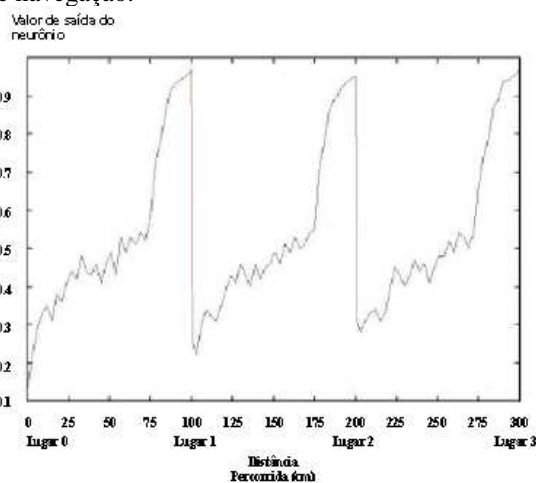


Figura 6. Saída do Controlador Fuzzy para a variável speed.

Referências Bibliográficas

- KUIPERS, B. J., BYUN, Y. T., 1991. "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations", *Robotics and Autonomous Systems*, Vol. 8, 47–63.
- SPACEK, L., 2003. "Omnidirectional Catadioptric Vision Sensor with Conical Mirrors", *Journal of Robotics e Auonomous System*.
- DUDA, R. O.; HART, P. E., 1973. "Pattern Classification and Scene Analysis". John Wiley E Sons.
- ZITOVÀ, B.; FLUSSER, J. 1999, "Landmark recognition using invariant features", *Pattern Recognition Letters*, Vol 20, pp. 541-547.