CONTROLADOR NEURAL NEBULOSO E PROBABILÍSTICO PARA A NAVEGAÇÃO AUTÔNOMA DE ROBÔS MÓVEIS

Rodrigo Calvo*, Roseli Aparecida Francelin Romero*

*Laboratório de Inteligência Computacional, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo
Av. do Trabalhador São-Carlense, 400 - Centro - Cx. Postal 668
São Carlos, São Paulo, Brasil

Emails: rcalvo@icmc.usp.br, rafrance@icmc.usp.br

Abstract— In this research an intelligent hybrid architecture is proposed based on neuro-fuzzy systems that is able to guide the robot in unknown environments. Initially, the system is not able to navigate, but after a trial and error period and some collisions, it improves in guiding the robot to the goal efficiently. This architecture is hierarquical and consists in two modules that generate innate behaviors, like obstacles avoiding and target reaching. The approach is consolidated in simulation and validated en real environments. To this end, this system has been implemented using Saphira simulator. Experiments in a real environments show the efficiency and learning capabilities of the navigation system, validating the intelligent hybrid architecture for mobile robots applications.

Keywords— robot autonomous navigation, neuro-fuzzy networks, fuzzy systems, robotics, reinforcement learning.

Resumo— Nesta pesquisa é proposta uma arquitetura híbrida inteligente baseada em sistemas neuro-fuzzy para capacitar o robô a alcançar alvos, ou pontos metas, em ambientes desconhecidos. Inicialmente, o sistema não tem habilidade para a navegação, após uma fase de experimentos com algumas colisões, o mecanismo de navegação aprimora-se guiando o robô ao alvo de forma eficiente. A arquitetura é hierárquica e costitiui-se de dois módulos responsáveis por gerar comportamentos inatos de desvio de obstáculos e de busca ao alvo. A abordagem já consolidada em simulação foi validada em ambientes reais neste trabalho. Para tanto, este sistema foi implementado e testado no simulador Saphira e experimentos em ambientes reais demonstraram a eficiência e a capacidade de aprendizagem do sistema de navegação, validando a arquitetura híbrida inteligente para aplicação em robôs móveis.

Keywords— navegação autônoma de robôs, redes neuro-fuzzy, sistemas fuzzy, robótica, aprendizagem por reforço.

1 Introdução

A necessidade da criação de sistemas e desenvolvimento de novas tecnologias para a substituição do trabalho humano é uma vertente em ascensão na humanidade. Tais sistemas devem ser capazes de executar tarefas sem que haja alguma interferência externa (sistemas independentes). Incluindo-se nesta classe, o problema de navegação autônoma de robôs consiste no desenvolvimento de mecanismos de tomada de decisão para um ou mais robôs móveis em um ambiente arbitrário no qual devem realizar determinadas tarefas de forma autônoma, como atingir alvos, desviar de obstáculos, explorar ambiente, etc (Calvo and Figueiredo, 2003) (Cazangi and Figueiredo, 2002) (Crestani et al., 2002). Diversas aplicações para tais sistemas vêm motivando os pesquisadores desta área, como por exemplo, limpeza de ambientes, embarcações, transporte de materiais, sistema de vigilância e tarefas consideradas arriscadas para o ser humano (Rao and Fuentes, 1996).

Algumas abordagens de sistemas de navegação consideram que o ambiente (topologia e eventos) é totalmente conhecido e todos os possíveis eventos pertencem a um número finito de classes conhecidas (ambientes estáticos e controlados). Neste caso, o sistema de navegação pode ser programado com antecedência, com poucos ou nenhum parâmetro adaptável. Há grandes expectativas para aplicações da robótica onde o ambiente é totalmente desconhecido. Pelo menos, neste caso, a autonomia é uma característica essencial para sistemas de navegação, em particular, e para sistemas de controle em geral (Crestani et al., 2002). A aprendizagem de sistemas de navegação autônoma inteligente é baseada nas iterações com os ambientes de navegação tornando o conhecimento prévio da topologia do ambiente desnecessário. A aprendizagem ocorre se o sistema de navegação não exibe um desempenho aceitável ou se novas experiências ocorrem. Essas premissas são investigadas em uma simples classe de problemas de navegação: o ambiente é desconhecido, há somente um alvo no ambiente e o robô precisa alcançá-lo evitando colisões contra obstáculos (Vershure, 1998).

Sistemas de navegação são propostos utilizando robôs móveis, adicionando a estes capacidade de mobilidade e autonomia para interagir adequadamente com diversos ambientes ampliando o campo de novas aplicações. Medeiros e Romero (2004) apresentam uma rede neural do tipo multi-camadas para controle de um robô

móvel em um chão de fábrica. Métodos probabilísticos de mapeamento e localização são utilizados para a navegação do robô móvel Pioneer I, capacitando-o a transportar documentos entre locais distantes (Bianchi and Romero, 2003). Quiles (2005) propõe um sistema de visão computacional baseado em redes neurais multi-camadas para capacitar um robô móvel a seguir um objeto de determinada cor e forma evitando colisões com obstáculos.

O presente trabalho estende a idéia proposta em (Calvo and Figueiredo, 2003) visto que algumas modificações estão sendo propostas para tornar o sistema operacional em tempo real. Um sensor a laser é acoplado ao sistema para a iteração do robô com o ambiente. O sistema modificado é testado utilizando-se o simulador *Saphira* e testes em tempo real realizados com o robô móvel Pioneer I do LABIC-ICMC-USP. Os experimentos apresentam comparações de desempenho entre as simulações efetuadas no sistema proposto em (Calvo and Figueiredo, 2003) e os testes realizados com o *Saphira* e com o robô móvel.

Este artigo é organizado como segue. Seção 2 descreve o sistema de navegação. O modelo do robô ideal utilizado em (Calvo and Figueiredo, 2003) e do robô móvel Pioneer I é definido na Seção 3. O módulo de localização de alvos para auxiliar na caputura dos mesmos é descrito na Seção 4. Resultados de simulação e dos testes em tempo real são mostrados, além de uma comparação para propósitos de avaliação de desempenho, na Seção 5. Uma breve discussão sobre os resultados obtidos e possibilidades para trabalhos futuros são apresentados na Seção 6.

2 Sistema de Navegação Autônoma

O sistema de navegação autônoma consiste de três módulos principais, conectados a um neurônio de saída. Dois destes módulos, Sistema Nebuloso de Busca ao Alvo (SBA) e Sistema Nebuloso de Desvio de Obstáculos (SDO) geram comportamentos instintivos. Um repertório de coordenação (RC) estabelece (após o processo de aprendizagem) pesos adequados para ponderar os comportamentos gerados pelos módulos SBA e SDO e combiná-los no neurônio de saída (Figura 1).

2.1 Repertório de Coordenação

Se o sistema de navegação consistisse somente dos sistemas nebulosos e se eles operassem juntos sem nenhuma coordenação, haveria muitas situações conflitantes e o desempenho da navegação seria certamente insatisfatório. A função do repertório de coordenação (RC) é coordenar os comportamentos instintivos, gerados pelos sistemas SDO e SBA.

O módulo RC é formado por três redes neurais nebulosas não-supervisionadas: Rede de Desvio de Obstáculos (RDO), Rede de Busca ao Alvo (RBA) e Rede de Distância ao Alvo (RDA). As redes são conectadas a diferentes tipos de sensores: sensores de distância a obstáculos, direção ao alvo e distância ao alvo, respectivamente (Figura 1).

A arquitetura das redes consiste de duas camadas de neurônios nebulosos (neurônios modelados de acordo com a Teoria Nebulosa (Gomide and Pedrycz, 1998)). A primeira camada é construtiva, ou seja, no instante em que o ciclo de aprendizagem é acionado, alguns neurônios são inseridos na primeira camada. Cada neurônio da primeira camada se conecta com o campo sensorial de distancia a obstáculos, direção ao alvo e distancia ao alvo (rede RDO, RBA e RDA, respectivamente). Há somente um neurônio na segunda camada, que é conectado com todos os neurônios da primeira camada.

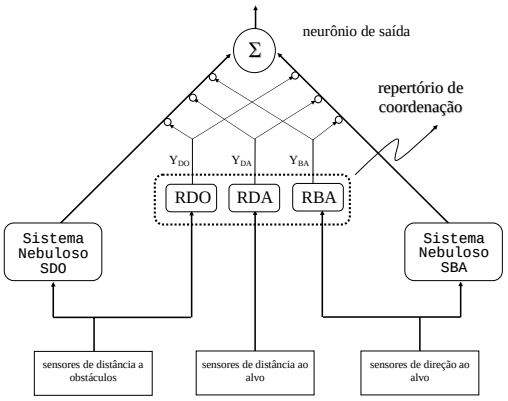


Figura 1: Arquitetura do sistema de navegação autônomo

As redes do RC influenciam no balanceamento de cada comportamento instintivo do robô (busca de alvo e desvio de obstáculos), baseando-se no mapa do ambiente (vetores de sinais fornecidos pelos sensores). Assim, a rede RDO aprende a associar classes de mapas do ambiente, com situações de eminência de colisão. O processo de aprendizado é ativado a todo o instante que o robô colide contra um o obstáculo (t_c). A rede RBA aprende a associar classes de sinais de direção ao alvo com o comportamento adequado de busca ao alvo. O processo de aprendizagem é ativado a todo instante em que o robô captura um alvo (t_a). A rede neural RDA aprende a associar classes de sinais de distância ao alvo com momentos em que o robô deve adotar o comportamento de busca ao alvo. O processo de aprendizagem é ativado a todo instante t_a. Os detalhes das arquiteturas, do processamento neural e do processo de aprendizagem são detalhados em (Calvo and Figueiredo, 2003).

Após o processo de aprendizagem, o módulo de coordenação pondera os sinais provenientes dos

módulos inatos, balanceando os comportamentos instintivos. As saídas das redes do módulo de coordenação estabelecem os pesos sinápticos para o neurônio de saída. O método de aprendizagem por reforço é utilizado para estas redes. A todo instante t_c um neurônio é inserido na rede RDO. Os pesos sinápticos, então, recebem os sinais dos sensores de distância a obstáculos. Analogamente, a todo instante t_a um neurônio é inserido nas redes RBA e RDA e os pesos sinápticos assumem os valores dos sinais dos sensores de direção ao alvo e de distância ao alvo, respectivamente.

3 Modelo dos Robôs

Para o sistema de navegação em (Calvo and Figueiredo, 2003) é definido um modelo de robô ideal para avaliar a estratégia de aprendizagem através de simulações.

O robô ideal utilizado nas simulações do sistema não apresenta dinâmica interna (Figura **??**), sua velocidade é constante e sua direção pode assumir valores entre $[-15^o, 15^o]$. O robô contém três classes de campos sensoriais com o objetivo de obter informações provenientes do ambiente. Tais classes são: 1) Os sensores de obstáculo estão dispostos na parte frontal do robô (de -90º a 90º) em um número de 50 sensores com a finalidade de obterem a distância entre o robô e o obstáculo mais próximo; 2) Os sensores de direção ao alvo estão distribuídos ao longo do perímetro do robô em uma quantidade de 100 sensores. A saída de cada sensor está relacionada com o ângulo entre sua direção e o alinhamento do robô com o alvo; 3) Os sensores de distância ao alvo fornecem ao sistema a distância entre o robô e o alvo. Um único sensor captura a distância desejada. Tal distância é mapeada e convertida para todos os 51 sensores de distância ao alvo através de uma função gaussiana.

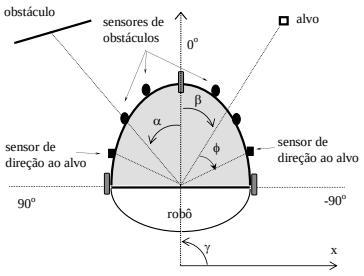


Figura 2: Pionner I

Tornando possível a validação do sistem neural nebuloso em experimentos reais, todo o desenvolvimento do sistema de navegação e experimentos foram realizados sob a plataforma do robô móvel Pioneer I [1] (Figura 3). Este robô está montado sobre um eixo de duas rodas que lhe permite fazer rotações e movimentos para frente e para trás. O robô é movido por dois motores com movimentação por diferencial. Suas capacidades de odometria variam bastante dependendo da superfície por onde ele está se locomovendo.

O robô móvel Pioneer I possui sete sonares ultra-sônicos sendo cinco deles localizados na parte frontal, um na lateral direta e um na lateral direita. Devido aos sonares não apresentarem medições satisfatórias para as tarefas de desvio de obstáculo, foi acoplado ao robô um sensor laser denominado *Proximity Laser Scanner* (PLS). O sensor é capaz de rastrear o ambiente com um ângulo de visão de 180 graus, resolução angular de 0,5 graus, distância máxima de 50 metros.

Figura 3: Pionner I

4 Módulo de Localização de Alvos (LA)

O módulo LA consiste no método de mapeamento (MM) do ambiente e no método de localização (ML), com o intuito de gerar o mapa do ambiente e localizar o robô no ambiente mapeado, respectivamente. No sistema neural nebuloso, o módulo LA supre a necessidade de informação a respeito da direção e distância do alvo, pois através deste módulo a posição do robô passa a ser conhecida (com erro de odometria minimizado).Com a posição do alvo já fornecida, é possível obter informações da direção e distância do mesmo. Essas informações, então, são fornecidas para os sensores de direção e de distância ao alvo, respectivamente. O módulo LA requer os sinais dos sensores de distância de obstáculos (sensor laser) para efetuar a tarefa de mapeamento do ambiente que auxilia na determinação da posição do robô. De acordo com as modificações citadas, a arquitetura do controlador autônomo sofre alterações como é mostrado na Figura 4.

4.1 Método de Localização (ML)

O método ML incorporado neste trabalho é a Localização de Markov (Fox et al., 1998), um método probabilístico que mantém várias hipóteses a respeito da posição do robô em um ambiente que

[1] *Activ*Media Robotics - http://www.activmedia.com

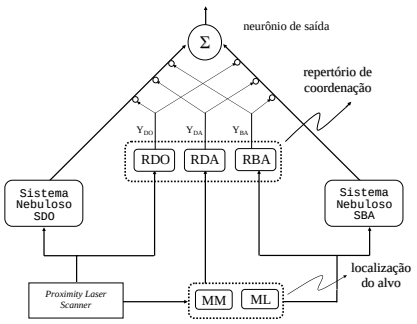


Figura 4: Arquitetura do sistema de navegação autônomo para ambientes reais

possuem um peso associado representando um fator probabilístico numérico. Este método de localização tem como objetivo encontrar a distribuição de probabilidades sobre todos os possíveis locais do ambiente.

Seja $l = \langle x, y, \theta \rangle$ uma posição no espaço de estados, em que x e y são coordenadas cartesianas do robô e θ é a sua orientação, estabelece-se a distribuição $Bel(l)$ sobre todos os locais l que indica a probabilidade do robô estar nas posições l. No entanto, a distribuição $Bel(l)$ expressa a crença subjetiva de que o robô possui em estar na posição l. Se a posição inicial do robô é conhecida, então a distribuição é $Bel(l)$ é centrada na posição correta (valor modal da distribuição), caso contrário, esta é uniformemente distribuída pra refletir tal incerteza.

A Localização de Markov aplica dois modelos probabilísticos diferentes para atualizar $Bel(l)$: 1) um modelo de ação para incorporar os movimentos do robô; 2) um modelo perceptual para atualizar a crença baseando-se na entrada sensorial. A movimentação do robô é modelada pela probabilidade condicional $p(l|l', a)$ que especifica a probabilidade de que uma ação a, quando executada em l', leve o robô para l. Dessa forma, $Bel(l)$ é atualizada de acordo com a fórmula geral proveniente do domínio das cadeias de Markov, dada pela equação 1:

$$Bel(l) = \sum_{l'} p(l|l', a) Bel(l) \qquad (1)$$

em que o termo $p(l|l', a)$ representa um modelo da cinemática do robô. Na implementação proposta por (Fox et al., 1998), foi assumido que os erros de odometria são normalmente distribuídos.

As leituras dos sensores são integradas de acordo com uma fórmula de atualização Bayesiana. Seja s uma leitura de sensor e $p(s|l)$ a probabilidade de se perceber s dado que o robô está na posição l, então $Bel(l)$ é atualizada de acordo com a regra da seguinte equação:

$$Bel(l) = \alpha p(s|l) Bel(l) \qquad (2)$$

em que o α é um fator de normalização que garante que $Bel(l)$ totalize 1 sobre todo l.

O método de Localização de Markov implementado em (Bianchi and Romero, 2003) utilizou o algoritmo de Monte Carlo que consiste na representação da distribuição (crença) $Bel(l)$ por um conjunto de amostras s associadas a um fator numérico de importância p. Tal fator indica a probabilidade da amostra ser relevante para a determinação da posição do robô. A crença inicial é obtida gerando-se aleatoriamente N amostras da distribuição prévia $P(s_o)$, e atribuindo-se o fator de importância uniforme N^{-1} para cada amostra.

4.2 Método de Mapeamento

O mecanismo de locomoção de um robô de maneira autônoma em ambientes não estruturados ocorre por meio do conhecimento, a cada instante, da sua localização no ambiente. A tarefa de localização de um robô móvel é alcançada em (Bianchi and Romero, 2003) utilizando o algoritmo de Monte Carlo citado na Seção 4.1. Porém, esse método requer a existência de um mapa do ambiente, que pode ser de difícil construção de forma manual.

O método para o mapeamento de ambientes utiliza uma representação probabilística e reticulada da informação espacial chamada *Occupancy Grid* (OC). A construção de mapas de ambientes é efetuado através dos sensores de distância ao alvo que apresentam ruídos. As células do espaço reticulado onde o método OC é aplicado armazenam uma estimativa probabilística de seu estado definido como livre ou ocupado (Elfes, 1989).

Um procedimento de estimativa Bayesiano permite a atualização incremental do método OC usando leituras realizadas por vários sensores sobre múltiplos pontos de vista. A variável de estado associada a uma célula $m_{\langle x,y \rangle}$ da representação espacial é definida como uma variável aleatória discreta com dois estados, livre e ocupada, denotadas EMP e OCC, respectivamente. Dessa forma, o método OC corresponde a um campo aleatório de estados discretos e binários. Uma vez que os estados das células são exclusivos e exaustivos, $P(\langle x, y \rangle = OCC) + P(\langle x, y \rangle = EMP) = 1$.

5 Resultados

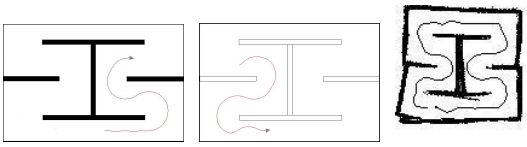
Nesta seção é apresentada a validação do sistema de navegasção proposto em ambientes reais. Para tornar válida a abordagem de aprendizagem adotada em (Calvo and Figueiredo, 2003) foi concebido um simulador de robô móvel (SR). Desta forma, pode-se notar a eficiência da proposta de acordo com os resultados obtidos em simulação.

A validação do sistema de navegação adotado no sistema neural nebuloso ocorre através da análise de comparação do desempenho dos ex-

perimentos realizados no simulador de robô móvel (SR), no simulador *Saphira* (SS), designando um estágio anterior aos testes em ambientes reais por oferecer comportamentos do robô semelhantes aos comportamentos reais, e em ambientes reais (AR). O sistema de navegação é validado com sucesso se este apresentar, em SS e AR, desempenho semelhante ou melhor ao desempenho apresentado no simulador SR. O desempenho do sistema caracteriza-se não somente pela proporção do número de colisões pelo número de caputras ocorridas e sim pelo número de capturas entre duas colisões consecutivas.

Vale ressaltar que algumas adaptações foram necessárias no sistema de navegação para que este seja aplicado em AR. O sensor laser (PLS) é acoplado ao robô Pioneer I para desempenhar a tarefa dos sensores de distância de obstáculos. No robô móvel não há a presença física de sensores que detectam o alvo. Assim como ocorre no simulador SR, a posição do alvo é fornecida ao sistema e então é encontrado o sensor de direção ao alvo que se encontra mais alinhado com o alvo dentre todos os sensores posicionados, virtualmente, no perímetro do robô. Ainda com a posição do alvo fornecida e juntamente com a posição do robô (por ele conhecida), é encontrada a distância entre o robô e o alvo pelo sensor virtual de distância ao alvo.

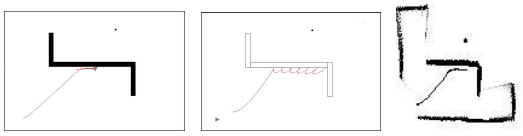
Primeiramente são apresentados os resultados de simulação quando somente o sistema SDO é executado. Neste caso, os alvos presentes no mundo são desconsiderados e a tarefa do robô é somente explorar o mundo sem que ocorram colisões. Para que o sistema SDO opere adequadamente, ou seja, gere o ângulo de rotação do robô, são necessárias as informações daquele sensor de distância a obstáculo que detectar o obstáculo mais próximo do robô. Tais informações são: o sinal detectado do sensor e sua posição angular no perímetro do robô. Observa-se na Figura 5 que não houve colisões contra obstáculos, mostrando a eficiência do sistema SDO em AR.



(a) (b) (c)

Figura 5: Trajetória do robô quando somente o sistema SDO é executado (a) simulador SR; (b) simulador SS; (c) AR

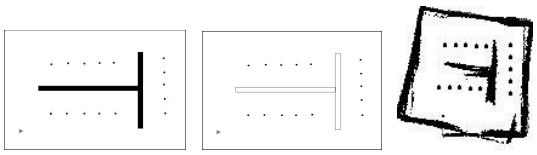
Em seguida, são realizados experimentos quando o sistema SBA é executado isoladamente e validando-o para a aplicação em AR. Com este módulo em execução, o robô não torna capaz de desviar dos obstáculos presentes no ambiente e sua tarefa passa a ser, somente, alcançar o(s) alvo(s). Para que o módulo inato SBA opere adequadamente, é necessária a informação do sensor de direção ao alvo que detectar o alvo com mais intensidade, ou seja, o sensor que estiver mais alinhado com o alvo. Tal informação se refere à posição angular do sensor no perímetro do robô. A Figura 6 ilustra experimentos do sistema SBA nos simuladores (SR e SS) e em AR. Nota-se que a captura do alvo torna-se impossível quando um obstáculo intercepta o caminho do robô ao alvo confirmando a eficácia do sistema SBA.



(a) (b) (c)

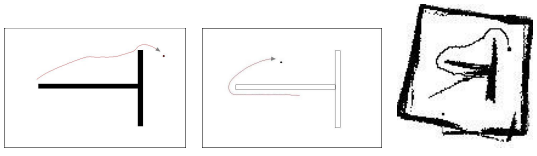
Figura 6: Trajetória do robô quando somente o sistema SBA é executado (a) simulador SR; (b) simulador SS; (c) AR

O módulo RC é o último componente da arquitetura do sistema neural nebuloso a ser validado. o módulo RC agrupa os sistemas SDO e SBA de modo a ponderar os comportamentos conflitantes gerados, tornando o robô capaz de adaptar a um mundo desconhecido e, após a etapa de aprendizagem, alcançar alvos evitando colisões contra obstáculos. Os experimentos para o módulo RC são realizados no ambiente ilustrado na Figura 7. Durante a simulação, somente um alvo está presente no mundo. Após a captura, o alvo (capturado) é eliminado e um novo alvo é inserido no mundo. As Figuras 8 e 9 mostram a trajetória para a captura de um alvo e o gráfico de desempenho do sistema de navegação, respectivamente.



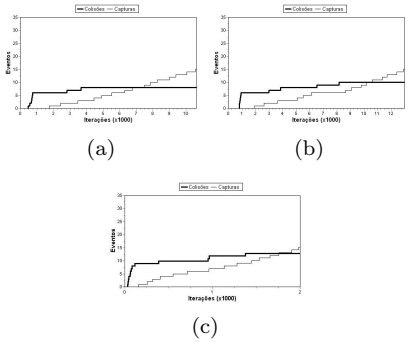
(a) (b) (c)

Figura 7: Modelo de ambiente para o experimento do módulo RC (a) simulador SR; (b) simulador SS; (c) AR



(a) (b) (c)

Figura 8: Trajetória do robô quando o módulo RC é executado (a) simulador SR; (b) simulador SS; (c) AR



(a) (b)

(c)

Figura 9: Desempenho do sistema de navegação autônomo (a) simulador SR; (b) simulador SS; (c) AR

Para avaliar o desempenho do sistema de navegação, apresenta-se os gráficos da Figura 9 em que a quantidade de colisões sofridas pelo robô e as capturas dos alvos são representadas pela curva mais espessa e pela curva fina, respectivamente. As curvas são plotadas em um plano que representa a quantidade de eventos (colisão e captura) na ordenada pelo número de iterações ocorridas na abscissa.

Verifica-se por meio dos gráficos que o desempenho do sistema de navegação foi semelhante entre os simuladores (SR e SS) e no AR. Durante o processo de adaptação ao ambiente, o robô sofre seguidas colisões e a captura dos alvos o estimula a capturar novos alvos. Isto mostra que o sistema proposto anteriormente (Calvo and Figueiredo, 2003), executado num ambiente idealizado (simulador SR), é robusto e funciona em aplicações de tempo real. Isto foi conseguido graças ao recurso agregado ao robô: o sensor laser (PLS). Os resultados obtidos pelos experimentos realizados valida o sistema neural nebuloso para a sua aplicação em AR pelo fato de apresentar um desempenho semelhante ao simulador SR.

6 Conclusão

O sistema de navegação foi testado e validado por um extenso conjunto de experimentos de navegação agrupados em 3 classes: 1) simulação em SR; 2) simulação no ambiente *Saphira*; 3) experimentos em AR. Os experimentos demonstraram que o sistema é aplicável em ambientes reais apresentando resultados semelhantes aos de simulação não perdendo sua característica de adaptabilidade e autonomia. Uma das contribuições desse trabalho está associada à integração do módulo LA, oferencendo autonomia ao sistema apresentado anteriormente (Bianchi and Romero, 2003).

Como trabalhos futuros pode-se citar a utilização do modelo da arquitetura do controlador autônomo juntamente com o mecanismo de aprendizado por reforço para aplicação de tarefas envolvendo múltiplos robôs de forma cooperativa e/ou competitiva, como por exemplo, exploração de ambientes. Assim, pretende-se dotar cada robô de um contolador autônomo semelhante ao apresentado neste trabalho. Outro ponto a ser pesquisado é a implantação de sensores reais que detectam a orientação do alvo, como por exemplo, sensores de luminosidades.

Referências

Bianchi, R. E. and Romero, R. A. F. (2003). Robô mensageiro baseado em navegação probabilística, Universidade Estadual de Campinas, VI Simpósio Brasileiro de Automação Inteligente, BauruCampinas - SP, Brasil.

Calvo, R. and Figueiredo, M. (2003). Reinforcement learning for hierarchical and modular neural network in autonomous robot navigation, IJCNN'03,International Joint Conference on Neural Networks, Portland - Oregon, USA.

Cazangi, R. and Figueiredo, M. (2002). Simultaneous emergence of conflicting basic behaviors and their coordination in an evolutionary autonomous navigation system, *Proceedings of 2002 Congress on Evolutionary Computation*, CEC'02, Congress on Evolutionary Computation, Honolulu - Hawai, USA.

Crestani, P. R., Figueiredo, M. and Zuben, F. V. (2002). A hierarchical neuro-fuzzy approach to autonomous navigation, *Proceedings of 2002 International Joint Conference on Neural Networks*, IJCNN'02, International Joint Conference on Neural Networks, Honolulu - Hawai, USA.

Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation, *Computer* **22**(6): 46–57.

Fox, D., Burgard, W., Thrun, S. and Cremers, A. B. (1998). Position estimation for mobile robots in dynamic environments, *AAAI/IAAI*, pp. 983 – 988.

Gomide, F. and Pedrycz, W. (1998). *An introduction to fuzzy sets: analysis and design*, The MIT Press, Cambridge, USA.

Rao, R. N. and Fuentes, O. (1996). Learning navigational behaviors using a predictive sparse distributed memory, *In From Animals to Animats: Proceedings of the 4th International Conference on Simulation of Adaptive Behavior* pp. 382 – 390.

Vershure, P. (1998). A bottom up approach towards the acquisition and expression of sequential representations applied to a behaving real-world device: Distributed adaptive control iii, *Neural Networks* **11**: 1531–1549.