NeuroFSM: aprendizado de Autômatos Finitos através do uso de Redes Neurais Artificiais aplicadas à robôs móveis e veículos autônomos

Daniel Oliva Sales[1], Fernando Santos Osório[1]
[1] ICMC - USP, Av. Trabalhador São-Carlense, 400, 13566-590 São Carlos - SP, Brasil
{dsales, fosorio} @ icmc.usp.br

Abstract. Este artigo apresenta o desenvolvimento de um sistema de controle para a navegação de robôs móveis e veículos autônomos. A abordagem adotada utiliza Redes Neurais Artificiais para o aprendizado de Autômatos Finitos que representam pontos estratégicos do ambiente, de forma que os robôs sejam capazes de lidar com os dados provenientes de seus sensores mesmo estando sujeitos a imprecisão e erros e assim possam considerar as diferentes situações e estados em que se encontram (contexto) para que possam decidir como agir para realizar o controle da sua movimentação, executando tarefas de controle e navegação desde as mais simples até as mais complexas e de alto nível. Para testar e validar a metodologia proposta, foram realizados experimentos em ambientes internos e externos, com diferentes sistemas sensoriais. Os resultados obtidos mostraram que a abordagem proposta é bastante apropriada para a navegação robótica autônoma.

Keywords: Navegação Autônoma, Robótica Móvel, Redes Neurais Artificiais, Autômatos Finitos. **Nível**: Mestrado. **Data de Conclusão**: 21/06/2012.

1 Introdução

A aplicação de técnicas de Inteligência Artificial à Robôs Móveis Autônomos e Veículos Inteligentes tem um importante papel na comunidade científica internacional [9][12]. Uma das características mais desejáveis em um robô móvel é a capacidade de navegação autônoma. Existem muitos trabalhos conhecidos e relevantes nessa area, destacando-se por exemplo o Darpa Challenge (2004 e 2005 no deserto e 2007 em ambiente urbano) [7][8] e o ELROB [13][14] como dois dos mais relevantes desafios de pesquisa nessa área. Os robôs móveis autônomos geralmente executam três tarefas principais: localização, mapeamento e navegação [15]. O foco principal deste trabalho é na tarefa de navegação, com tomada de decisão sobre a atuação do robô em ambiente conhecido. São utilizadas Redes Neurais Artificiais para que o robô aprenda os possíveis estados e respectivas ações em um sistema, se locomovendo de forma segura.

1.1 Motivação e Objetivos

Há uma enorme quantidade de aplicações para robôs móveis. As aplicações desenvolvidas neste trabalho se relacionam principalmente ao monitoramento e segurança de ambientes internos e à condução autônoma de veículos em ambientes urbanos. No caso das aplicações *indoor*, o sistema proposto se adapta perfeitamente a tarefas de

patrulhamento e monitoramento, tanto com um único robô, como em sistemas multirrobóticos [2], auxiliando na segurança de setores estratégicos bem como atuando em missões não-tripuladas em ambientes hostis. Já no cenário de aplicações para ambientes urbanos, o sistema desenvolvido pode vir a integrar sistemas de direção autônoma, viabilizando a implementação de diversas aplicações comerciais.

Este trabalho se encaixa em um projeto maior realizado no Laboratório de Robótica Móvel (LRM[1]) do Inst. de Ciências Matemáticas e de Computação (ICMC) ligado ao INCT-SEC (Inst. Nacional de C&T em Sistemas Embarcados Críticos). O INCT-SEC possui atualmente 4 grupos de trabalho principais, sendo dois deles envolvendo pesquisas sobre controle e navegação de robôs e veículos autônomos: "Robôs Táticos para Ambientes Internos"; e "Veículos Terrestres Autônomos".

O principal objetivo deste trabalho é portanto a implementação de um sistema de controle para navegação autônoma adequado tanto para aplicações indoor (com robôs de pequeno porte) como para aplicações outdoor (com veículos autônomos). O sistema deve possuir tolerância a falhas, de modo que o robô consiga lidar com imprecisão e erro nos dados de entrada e capacidade de detectar o contexto em que o robô se encontra, escolhendo assim ações adequadas de acordo com cada situação.

2 Trabalhos Relacionados

A navegação autônoma é uma tarefa que requer o uso de sensors, como os de posicionamento, referencial inercial, detecção de obstáculos, imagens e orientação individualmente ou combinados [9][15][16]. É através das informações sensoriais que se constrói o mapa do ambiente e se realiza a auto-localização, assim como o planejamento de trajetórias, que se baseia no mapa e nas posições inicial e final conhecidas. Se o mapa do ambiente e a localização do robô não forem conhecidos, na maioria dos casos apenas uma navegação reativa seria possível. Para implementação de um sistema de navegação autônoma, um sistema puramente reativo não é adequado, já que a reação imediata à informação obtida dos sensores não é suficiente para garantir um controle correto do robô em um caminho mais complexo. Um sistema mais robusto deve então ser implementado, provendo informações de sequência e contexto que estão ausentes em modelos puramente reativos.

Abordagens baseadas em Autômatos Finitos (FSM) [5] tem sido bastante utilizadas. Autômatos finitos são úteis porque o sistema pode facilmente ser definido como uma sequência de estados (mudanças de contexto), levando em consideração as entradas (sensores) permitindo assim a mudança de um estado a outro, e ainda definindo para cada estado uma ação específica associada (atuação dos motores). Desta forma, para cada estado detectado o robô reage apropriadamente.

O uso de um método de aprendizado de máquina como Redes Neurais Artificiais (RNAs) é uma forma muito interessante de processar a informação sensorial, identificando e classificando os estados (corrente e transições), e determinando quais ações devem ser executadas [6]. RNAs são tolerantes a ruído e imprecisão nos dados de entrada, e capazes de detectar estados e transições entre esses estados. RNAs são também muito eficientes na generalização do conhecimento [17]. Desta forma, RNAs são adequadas para a percepção de características do ambiente e detecção de estados.

[1] LRM: http://www.lrm.icmc.usp.br/

A associação de RNAs com FSM já vem sendo discutida e desenvolvida desde os anos 90 [18][19][20]. Trabalhos mais recentes utilizando o aprendizado de autômatos por Redes Neurais em problemas de robótica foram desenvolvidos para a tarefa de estacionamento autônomo em vagas paralelas [1], em ambiente simulado. O trabalho apresentado neste artigo utiliza uma abordagem inspirada no modelo proposto em [1], porém com modificações que permitem a implementação de uma quantidade maior de aplicações em em ambientes internos e urbanos.

3 Metodologia

O sistema de navegação autônoma baseado em RNA e FSM baseia-se na idéia de que a tarefa de navegação pelo ambiente pode ser descrita na forma de um autômato com características desse ambiente (pontos-chave) interpretadas como estados e transições. Sendo assim, uma Rede Neural é treinada para aprender a reconhecer os possíveis estados desse autômato, de tal forma que o estado atual seja detectado através da leitura dos sensores. Cada estado atual tem um ou mais comportamentos específicos (reativos ou deliberativos) associados. A navegação é realizada com base em um planejamento de trajetória, de tal forma que o robô conhece a sequência de estados que descreve um determinado caminho. A RNA passa a atuar portanto como um classificador dos dados de entrada permitindo ao robô conhecer o contexto em que se encontra e percorrer qualquer trajeto, deslocando-se pelo ambiente enquanto alcança posições específicas segundo uma sequência de estados que descreve a sua tarefa/rota.

3.1 Representação do ambiente através de Autômatos Finitos

Para se descrever a trajetória/tarefas do robô na forma de autômatos finites (FSM), é desejável adotar um modelo de ambiente/mapa que favoreça esse tipo de implementação. Sendo assim, foi adotado para este sistema o modelo baseado em mapas topológicos, considerando que tanto os mapas topológicos como os autômatos são representados através de grafos, ou seja, o mapa do ambiente e os autômatos de controle são representados com a mesma estrutura de dados.

Considerando ainda que o controle de navegação é baseado justamente em uma sequência de etapas a serem seguidas (trechos a serem visitados), e que o mapa topológico armazena partes específicas do ambiente e pontos de referência, o autômato finito é o próprio mapa topológico. Cada estado do autômato corresponde a uma parte específica do ambiente, e a transição entre os estados acontece quando o robô se desloca para uma nova área do ambiente mapeado. Desta forma, a navegação de um nó a outro do mapa ocorre tomando-se como estado inicial o nó onde o robô se encontra, e como estado final o nó de destino no mapa topológico. Para que seja realizada a navegação de um ponto a outro usando este tipo de abordagem, o autômato (caminho entre os pontos origem/destino no grafo) deve ser estabelecido e armazenado em memória para o planejamento da trajetória do robô.

Uma grande vantagem deste tipo de notação é eliminar a necessidade de algoritmos mais precisos de localização e mapeamento. O robô consegue estimar sua posição aproximada (em qual ponto do mapa topológico se encontra), e com isso consegue executar a tarefa de localização ao mesmo tempo em que controla a navegação.

A percepção do estado atual (parte do mapa onde o robô se encontra) se dá pelo reconhecimento de certos padrões estruturais no ambiente (pela RNA), como por exemplo a presença de paredes em posições específicas ou o formato da área navegável à frente, minimizando o impacto causado por possíveis erros e imprecisão nos sensores, o que seria crítico em um modelo métrico.

3.2 Classificador Neural

O classificador neural é o módulo responsável por determinar qual é o estado corrente (classe) dado um conjunto de valores de entrada obtido do sistema sensorial do robô a cada instante. O método de aprendizado adotado é o supervisionado, onde os pares entrada/saída representam as possíveis situações (estados) em um determinado ambiente e suas respectivas classificações. O modelo passa por uma fase de treinamento, onde recebe os dados de treino, adaptando os pesos da rede [17].

Os pares entrada/saída utilizados no treinamento devem ser gerados a partir de uma coleta de dados feita com o robô no ambiente alvo. A saída esperada (classificação) de cada exemplo deve ser especificada manualmente pelo supervisor do sistema nesta etapa. São utilizados como dados de entrada a informação obtida do sistema sensorial (sensor), e como saída uma codificação binária que representa o estado detectado. É importante ressaltar que para obter resultados satisfatórios no treinamento da rede, deve ser coletada uma quantidade suficiente de exemplos para que a base de dados esteja balanceada, isto é, os dados devem ser coletados na mesma proporção para cada classe. Uma vez coletados e classificados esses dados, é realizado o treinamento até que se alcance uma convergência, sendo feita uma validação pelo método de *cross-validation* que define se a RNA está pronta para o uso [17]. O uso de uma base de aprendizado e outra de teste de validação permite avaliar se a rede realmente aprendeu as diferentes classes e se esta adquiriu a capacidade de generalizar o aprendizado.

3.3 Navegação Topológica Híbrida

Uma RNA treinada para classificar os dados sensoriais, reconhece os padrões que descrevem os estados. Assim o sistema pode começar a navegação autônoma. As fases iniciais de preparação do sistema são necessárias para que com um único treinamento o sistema seja capaz de reconhecer as diferentes características do ambiente e assim navegar por qualquer caminho (sub-grafo do mapa topológico).

Dada uma rota desejada (nó de destino e posição inicial/atual do robô), a geração do autômato de navegação é feita selecionando-se um caminho entre esses dois nós (sub-grafo) no mapa topológico, e associando a cada estado intermediário ações específicas que levem o robô a alcançar o estado seguinte. Com essa lista de estados/ações devidamente armazenada, a Navegação Topológica Híbrida pode ser iniciada. Esta etapa consiste em seguir a sequência pré-estabelecida de estados, ativando para cada estado um comportamento reativo adequado para esta situação.

O sistema sensorial fica responsável por capturar dados do ambiente que são processados pelo classificador, indicando qual é o estado corrente. A partir do momento em que as detecções correspondem ao próximo estado armazenado na lista de estados, há uma troca do context atual, alterando o estado corrente e consequentemente o comportamento associado. A Figura 1 ilustra este processo.

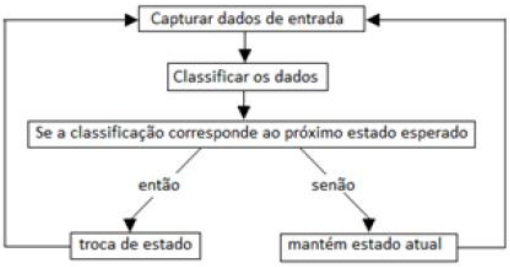

Fig. 1. Fluxograma do controle de navegação.

O comportamento deliberativo (sequenciamento de ações) por si só não é suficiente para garantir uma navegação segura em cada parte do ambiente, já que os mapas topológicos não oferecem informações detalhadas sobre a distância e posicionamento absoluto dos elementos no ambiente. Por este motivo é que comportamentos reativos são associados a cada estado, permitindo ao robô realizar uma navegação local segura e adequada para cada diferente configuração do ambiente (ex. manter-se no centro de um corredor ao seguir em frente).

No caso de situações que exigem tomadas de decisão, como nos cruzamentos, esta é realizada na etapa de planejamento de trajetória (na geração do autômato), deliberadamente associando ao estado o comportamento correto desejado (curva à direita, esquerda ou seguir em frente). Conforme citado anteriormente, a tarefa de auto-localização é realizada simultaneamente à própria navegação, já que a posição aproximada do robô no ambiente sempre corresponde ao estado atual do autômato.

4 Experimentos e Resultados

Para testar e avaliar a abordagem proposta, diversos experimentos foram realizados aplicando a metodologia descrita na seção anterior em diferentes situações. Os testes foram conduzidos tanto em ambiente simulado como com robôs reais. Diversos tipos de sensores foram utilizados, entre eles Câmera RGB monocular, Laser SICK e Sensor Kinect. O método foi avaliado tanto em ambientes internos como em vias reais de trânsito. Para a análise do classificador, utilizou-se o método *Stratified 5-fold Cross-Validation*, particionando a base de dados na proporção 80-20, (80% dos dados usados para treinamento e 20% para teste), separados em 5 conjuntos de treinamentos. A implementação da RNA foi feita com a ferramenta SNNS e JavaNNS, onde ao concluir o treinamento com sucesso as redes foram convertidas para linguagem C para que fossem incluídas no programa de controle do robô. Os principais experimentos realizados e os resultados obtidos estão descritos a seguir.

4.1 Sistema de navegação topológica autônoma em ambientes internos

Neste primeiro experimento utilizando a metodologia proposta [4], foi implementado o controle de navegação autônoma em uma tarefa de patrulhamento e vigilância em um ambiente interno: um andar de um prédio composto por diversas salas e

corredores. Foram realizados testes inicialmente no ambiente simulado do Player-Stage, e em seguida utilizando um robô real de pequeno/médio porte, um Pioneer P3-AT equipado com um sensor Laser SICK e câmera termal.

O sistema sensorial consistia em um laser SICK LMS291 como sensor principal. A leitura de 20 feixes, com intervalo de ângulos de 10 graus entre si, foram a informação utilizada como entrada do classificador, definindo portanto uma camada de entrada da RNA com 20 neurônios. Foi utilizada ainda uma câmera termal Flir PathFindIR como sensor secundário, realizando a detecção de instrusos humanos (calor) no ambiente, porém a leitura desse sensor não foi incluída no classificador.

Teste 1	Teste 2	Teste 3	Teste 4	Teste 5
248 2 0 0 0 2 185 0 0 0 0 0 207 1 0 0 0 1 235 0 0 0 0 0 208	250 0 0 0 0 0 187 0 0 0 1 0 205 0 1 0 0 0 235 0 0 0 0 0 208	249 0 0 0 0 2 184 0 0 0 0 0 204 2 1 0 0 0 235 0 0 0 0 0 208	248 0 1 0 0 1 185 0 0 0 0 0 205 1 1 0 0 0 235 0 0 0 0 0 207	249 0 0 0 0 4 182 0 0 0 0 0 206 0 1 0 0 0 235 0 0 0 0 0 207
Acertos: 1083 Erros: 6	Acertos: 1085 Erros: 2	Acertos: 1080 Erros: 5	Acertos: 1080 Erros: 4	Acertos: 1079 Erros: 5

Fig. 2. Matriz de confusão dos testes da RNA 20-20-5.

(a) Curva à direita (b) Interseção (c) Reta (corredor) (d) Curva à direita (porta)

Fig. 3. Trechos do cenário utilizado nos testes (videos em: http://www.youtube.com/lrmicmc/)

Foram adotados cinco estados para representação das situações possíveis nesse ambiente. Desta forma, a camada de saída da RNA foi composta por 5 neurônios. Por fim, a quantidade de neurônios na camada escondida foi definida de forma empírica.

Foram gerados no total 5429 pares entrada/saída desejada para treinamento. O algoritmo de treinamento utilizado foi o Resilient Propagation (R-Prop), com seus parâmetros padrão. A convergência foi alcançada antes de 500 ciclos. As médias de acerto da RNAs testadas ficaram acima de 90%, e foi escolhida a RNA 20-20-5 para integrar o sistema de controle do robô real. As matrizes de confusão dos testes dessa RNA são apresentadas na Figura 2, onde é possível observar o alto índice de acerto do sistema em todas as classes. A Figura 3 apresenta alguns trechos percorridos pelo robô durante a navegação no ambiente utilizado.

O uso da câmera termal Flir PathFindIR foi proposto como uma funcionalidade adicional ao sistema para detecção de agentes (humanos ou não). O frame capturado é processado utilizando OpenCV, e permite avaliar se há um agente no ambiente ou não, já que que a temperatura do agente (quente) é bem diferente da temperatura do ambiente (frio). Esta funcionalidade é bastante útil em tarefas de vigilância.

Os resultados obtidos foram satisfatórios, tanto pela análise do desempenho do classificador como pela conclusão bem sucedida da tarefa de navegação, demonstrando a viabilidade da abordagem topológica adotada neste trabalho.

4.2 Sistema de patrulhamento autônomo multi-agente

Este experimento implementou o sistema de navegação topológica obtido no trabalho anterior em um problema mais complex, em um sistema de patrulhamento multirrobótico [2]. Os experimentos foram conduzidos em um ambiente simulado, cuja representação é ilustrada na Fig. 4. Os cinco estados implementados foram: Detecção de agente (a), Reta (b), Curva à esquerda (c), Curva à direita (d) e Interseção (e).

Os robôs representados na simulação bem como seus sistemas sensoriais são os mesmos utilizados no experimento anterior (Robô Pioneer P3-AT com Laser Sick). A única diferença está no sistema de percepção, com relação a quantidade de feixes laser observados: foram usados 180 feixes com intervalo de 1 grau entre si, ao invés dos 20 feixes usados anteriomente.

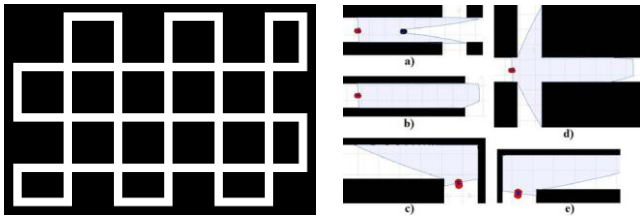

Fig. 4. Mapa do ambiente utilizado no experimento e estados implementados.

O sistema foi composto por quatro robôs autônomos com comunicação entre si, e sem estação central de controle. O comportamento de cada robô era determinado de acordo com uma combinação do estado corrente detectado com as mensagens recebidas dos agentes aliados (que compõem o grupo robótico patrulheiro).

Dois comportamentos principais foram implementados: "Patrulhando" e "Cercando inimigo". Na etapa de patrulhamento, os robôs podiam navegar pelo ambiente de forma aleatória ou seguindo um caminho cíclico pré-estabelecido (rota de patrulha). Se durante a patrulha um agente qualquer fosse detectado por um robô, uma mensagem é enviada em broadcast para o grupo com a posição do agente encontrado no mapa topológico. Cada robô responde com sua posição atual, e se nenhum dos robôs aliados é esse agente, o comportamento "cercar inimigo" é acionado. Caso contrário, mantém-se o comportamento de patrulha.

O planejamento de trajetórias nesse experimento ocorreu de duas formas: definir a rota a ser seguida por cada robô em sua patrulha (cíclica ou aleatória); planejar o menor caminho entre cada robô e o invasor detectado para cercá-lo. No comportamento de patrulha, a geração do caminho é realizada da mesma forma que no experimento anterior, definindo-se um ponto de destino para criação do autômato de navegação. Já no comportamento de neutralização do inimigo o caminho é calculado com o algoritmo A* (A Star). Cada robô calcula o menor caminho entre ele e o invasor levando em consideração a posição dos aliados de forma a cercar esse invasor. Para isto, o mapa do ambiente é convertido em um grid. Um algoritmo é responsavel por analisar esse caminho obtido com o A* e gerar a partir dele um autômato (grafo) de navegação. Essa implementação permitiu a geração de caminhos topológicos que priorizam a distância mínima entre origem/destino.

Neste experimento, a base de dados continha 2835 exemplos, e os parâmetros de treino foram os mesmos do experimento anterior. Foi alcançado 100% de acerto nos conjuntos de teste usados, e a tarefa de cercar o invasor concluída com sucesso.

4.3 Sistema de navegação baseada em visão 3D usando sensor Kinect

Os experimentos anteriores realizados com sistema sensorial baseado em laser tiveram um importante papel nesta pesquisa, validando a metodologia proposta tanto nas simulações como no ambiente real. Uma vez que o sistema de navegação topológica autônoma tinha se tornado uma proposta madura, foram projetados experimentos para avaliação da aplicação dessa metodologia com outros sistemas sensoriais, ambientes e estados, como é o caso do experimento descrito nesta seção.

Neste experimento, foi implementado um sistema de navegação autônoma para robôs de serviço em ambientes internos [10]. Procurou-se adotar um sistema sensorial de baixo-custo (em oposição ao modelo baseado em laser), mas que possuísse bom volume de dados detectados e diversidade no tipo de informação extraída. Foi adotado para este fim o sensor Kinect, um sensor 3D de baixo custo desenvolvido pela Microsoft para o vídeo-game XBOX 360 que permite uma estimação da profundidade dos elementos capturados no ambiente (nuvem de pontos 3D – sensor RGB+*Depth*).

As leituras do Kinect são armazenadas em mapas de profundidade (matrizes) de dimensões 640x480. Sendo assim, este sensor para detecção de profundidades fornece 307200 pontos por leitura. O uso direto da matriz de dados sensoriais como entradas da RNA tornaria muito complexo o treinamento da rede (excess de entradas), portanto foi implementada uma etapa de pré-processamento, onde foi selecionada uma área de interesse nessa matriz de captura (mapa de profundidade) do Kinect.

Definiu-se uma área de interesse com 80 linhas da área central da matriz, sendo calculada a média entre essas 80 linhas, e resultando assim em um vetor de 632 posições, usado como entrada da RNA. O ambiente foi representado através de 8 estados: corridor livre, curva à esquerda, curva à direita, bifurcação esq/dir, corredor com porta a esquerda, corridor com porta a direita, cruzamento e final do corridor.

A base de dados possuía 1421 pares entrada/saída (aproximadamente 180 exemplos por classe). O algoritmo de treinamento mais uma vez foi o R-Prop, com parâmetros padrão e critério de parada de 1000 ciclos. Dentre as cinco topologias testadas, os melhores resultados foram obtidos pela RNA 632-316-8, que obteve média de acerto total de 92,2%. O erro final foi minimizado após a implementação de um filtro que impede oscilação na troca de estados, o que permitiu que mais uma vez a tarefa de navegação fosse concluída corretamente, usando um robô real (P3-AT).

4.4 Sistema de navegação baseada em visão em ambientes urbanos

Este experimento avaliou a implementação da metodologia proposta em um ambiente urbano (vias de trânsito reais) para o controle de navegação de um veículo autônomo, complementando a gama de aplicações que faziam parte dos objetivos deste trabalho.

Adotou-se um modelo sensorial baseado em visão monocular para aquisição dos dados de entrada, devido a características como baixo custo, peso e consumo de energia. O sistema desenvolvido divide-se em duas etapas principais: detecção na imagem das áreas navegáveis do ambiente, e reconhecimento de estados através do modelo topológico. A primeira etapa consiste na geração de mapas de navegabilidade através da classificação do frame capturado pela câmera, e a segunda consiste no reconhecimento de padrões nesses mapas de navegabilidade para detecção de estados.

Inicialmente o frame capturado foi processado para determinar quais eram as áreas navegáveis e não-navegáveis. Para isto, foi utilizado o classificador neural desenvolvido por Shinzato [3]. Este classificador usa um comitê de 5 RNAs, onde cada RNA avalia atributos diferentes da imagem, e gera como saída uma matriz com valores entre 0 e 1, indicando a certeza sobre a navegabilidade dos pontos representados na imagem capturada. Essa matriz é usada como entrada do classificador de estados. A imagem de 320x240 pixels é processada em blocos de 10x10 pixels, e a matriz de navegabilidade gerada possuía dimensões 32x24, resultando em 768 valores. Como nos exemplos adquiridos a metade superior da imagem estava relacionada sempre à informação acima da linha do horizonte, apenas a metade inferior da matriz foi usada como entrada da RNA, definindo uma camada de entrada com 384 neurônios.

Cinco estados foram implementados, ilustrados na Figura 5. Foram gerados 11186 exemplos para treinamento, obtidos através da captura de aproximadamente 1 minuto de video a 30fps para cada classe. Dentre as topologias testadas, a RNA 384-384-5 foi a que apresentou melhores resultados. A Figura 6 apresenta as matrizes de confusão dos testes realizados, onde é possível ver que o índice de acerto por classe foi bastante satisfatório. A acurácia média ficou em torno de 99,44%, superando os resultados obtidos nos sistemas implementados para ambientes internos, o que demonstra a viabilidade do modelo baseado em visão para detecção de estados. Os resultados obtidos são bastante promissores, demonstrando a viabilidade do uso da abordagem topológica no controle de veículos autônomos.

Fig. 5. Estados possíveis no ambiente utilizado.

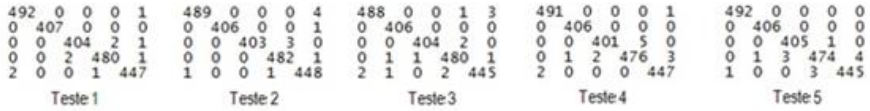

Fig. 6. Matrizes de confusão nos 5 testes realizados.

5 Conclusão

As pesquisas realizadas durante este projeto de mestrado possibilitaram o desenvolvimento de uma metodologia para implementação de Sistemas de Navegação Autônoma baseados no aprendizado de Autômatos Finitos por Redes Neurais Artificiais. A abordagem utilizada dispensa o uso de mapas métricos e de algoritmos de localização e de mapeamento muito precisos, complexos e com alto custo computacional, possibilitando a implementação do método mesmo em sistemas com recursos computacionais limitados.

Com os diversos experimentos realizados, foi ressaltada a capacidade de adaptação do método a diversos tipos de sensores, aplicações e ambientes, servindo de referência para o desenvolvimento de trabalhos futuros, tanto em implementações baseadas em sistemas sensoriais de baixo-custo como em implementações que privilegiam volume e riqueza de informação sensorial.

Concluindo, os resultados obtidos neste trabalho foram muito satisfatórios, tanto do ponto de vista quantitativo (análise do nível de acerto das RNAs nas implementações desenvolvidas) como do ponto de vista qualitativo (avaliação da tarefa de navegação nos experimentos realizados).

Agradecimentos. Os autores agradecem à FAPESP e ao CNPq pelo suporte ao INCT-SEC (Inst. Nacional de C&T em Sistemas Embarcados Críticos), processos 573963/2008-9 e 08/57870-9, e à FAPESP pelo suporte financeiro desta pesquisa (bolsa de mestrado).

Referências

1. Heinen, M.; Osório, F.; Heinen, F.; Kelber, C. SEVA3D: Autonomous Vehicles Parking Simulator in a three-dimensional environment, 2007.
2. Sales, D.; Feitosa, D.; Osório, F.; Wolf, D. Multi-agent autonomous patrolling system using ANN and fsm control. In: II CBSEC: Sist. Embarcados Criticos, Campinas, Brasil, 2012
3. Shinzato, P. Y, Wolf, D. F. Features Image Analysis for Road Following Algorithm Using Neural Networks. September, 2010.
4. Sales, D; Osório, F; Wolf, D. Topological Autonomous Navigation for Mobile Robots in Indoor Environments using ANN and FSM. In: Proceedings of the I Brazilian Conference on Critical Embedded Systems (CBSEC), São Carlos, Brazil, 2011.
5. Hopcroft, J.E., Ullman, J.D. (1979) “Introduction to Automata Theory, Languages and Computation”. Addison - Wesley, 1979.
6. Marino, A.; Parker, L.; Antonelli, G. and Caccavale, F. Behavioral Control for Multi-Robot Perimeter Patrol: A Finite State Automata approach. In: ICRA, 2009.
7. Thrun, S. et al. (2006) "Stanley: The Robot that Won the DARPA Grand Challenge," Journal of Field Robotics, Vol. 23, No. 9, June 2006, p.661-692.
8. Urmson, Chris et al. (2008). “Autonomous driving in urban environments: Boss and the Urban Challenge”. In: Journal of Field Robotics. V.25 , N.8 (August 2008). Pages 425-466.
9. Buehler, M.; Iagnemma, K.; Singh, S. (Editors). The 2005 DARPA Grand Challenge: The Great Robot Race (Springer Tracts in Advanced Robotics). Springer. (October, 2007).
10. Sales, D.; Correa, D.; Osório, F. S.; Wolf, D. F. 3D vision-based autonomous navigation system using ann and kinect sensor. In: 13th Engineering Applications of Neural Network Conference (EANN2012), London, UK, 2012
12. Bishop, Richard. (2000). "A Survey of Intelligent Vehicle Applications Worldwide". Proceedings of the IEEE intelligent Vehicles Symposium 2000. pp.25-30.
13. Kuhnert, K.-D. (2008). "Software architecture of the Autonomous Mobile Outdoor Robot AMOR". Proceedings of the IEEE Intelligent Vehicles Symposium, 2008. pp. 889-894.
14. Schilling, Klaus. (2008) “Assistance Systems for the Control of Rovers”. SICE Annual Conference, Tokyo, Oct. 2008.
15. Wolf, D.; Osório, F.; Simões, E.; Trindade Jr., O. Robótica Inteligente: Da Simulação às Aplicações no Mundo Real. [Tutorial] In: André de Carvalho; T. Kowaltowski. (Org.). JAI - SBC. RJ: SBC - Editora da PUC. 2009, v. 1, p. 279-330.
16. Goebl, M.; Althoff, M.; Buss, M.; et al. (2008) "Design and capabilities of the Munich Cognitive Automobile". IEEE Intelligent Vehicles Symposium, 2008. Page(s): 1101 – 1107
17. Haykin, S. Neural Networks: A Comprehensive Foundation. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.
18. Giles, C. Lee; Horne, Bill G.; LIN, Tsungnan. Learning a class of large finite state machines with a recurrent neural network. Neural Networks 8(9): 1359-1365. 1995.
19. Omlin, Christian W.; Giles, C. Lee. Constructing Deterministic Finite-State Automata in Recurrent Neural Networks. Jornal of the ACM 43(6): 937-972. 1996
20. Frasconi, P.; Gori, M,; Maggini, M.; Soda, G. Representation of finite state automata in Recurrent Radial Basis Function networks. Machine Learning 23:1, 5-32 1996.