

# SISTEMAS CLASSIFICADORES APLICADOS À NAVEGAÇÃO DE ROBÔS AUTÔNOMOS

André Mendeleck<sup>1,2</sup>, Douglas Eduardo Zampieri<sup>1</sup>, Arthur Miranda Neto<sup>1,2</sup>,

<sup>1</sup>UNICAMP - Universidade Estadual de Campinas

<sup>2</sup>FIPEP-Faculdades Integradas do Instituto Paulista de Ensino e Pesquisa

[andre.mendeleck@uol.com.br](mailto:andre.mendeleck@uol.com.br)

[douglas@fem.unicamp.br](mailto:douglas@fem.unicamp.br)

[arthur@fem.unicamp.br](mailto:arthur@fem.unicamp.br)

**RESUMO:** A geração de trajetórias para robôs móveis pode ser macro classificada em duas grandes categorias, tendo o ambiente (área de trabalho) como elemento principal: conhecida e determinística; ou desconhecida e dinâmica. Enquanto a navegação de robôs por ambiente conhecido e determinístico pode-se resumir à aplicação de algoritmos de busca com critérios de otimização, a navegação por ambientes desconhecidos e dinâmicos exige a utilização de algoritmos adaptativos. Neste artigo analisamos a navegação de robôs em ambiente desconhecidos utilizando-se aprendizado evolutivo com sistemas classificadores.

**Palavras-Chave:** Inteligência Artificial, Sistemas Classificadores, Algoritmos Genéticos, Wireless, Robótica.

## I. INTRODUÇÃO

A geração de movimentos para robôs móveis em ambientes desestruturados é um dos tópicos mais estudados na área de robótica e automação. Se o ambiente for conhecido e estático, os problemas podem ser primeiramente reduzidos à modelagem do ambiente e à busca por caminhos seguindo algum critério de otimização (distância, energia, processamento, número de movimentos, qualidade dos deslocamentos, ...). Para ambientes desestruturados, o cenário para estudo é dinâmico, com vários elementos em movimentos. Assim, conduzir um robô de um ponto inicial até o seu destino, envolve a execução de operações complexas e não determinísticas, como por exemplo, a interação com o ambiente, identificação dos elementos ambientais e tomadas de decisões. Neste caso, a geração de trajetória exige o tratamento de uma série de fatores desconhecidos pelo software de planejamento, tais como, o volume da área de trabalho e os elementos móveis e fixos. Sem estas

informações, o custo computacional para a geração de trajetória torna-se bastante elevado, principalmente se considerarmos as formas convencionais de programação de robôs. No entanto, em algumas situações não é possível caracterizar, com antecedência o ambiente de trabalho. Uma situação mais complexa é a de um robô móvel que não conhece a trajetória a seguir, devendo desenvolver ("determinar"), em tempo real de execução, a trajetória a ser seguida, além de deslocar-se desviando de obstáculos.

Por outro lado, o ser humano é capaz de realizar a navegação por ambientes desestruturados com alto grau de adaptação, uma vez que sua estrutura neuronal propicia um certo grau de plasticidade e adaptabilidade nas ações, capacitando o indivíduo a identificar, analisar, propor soluções, atuar e a aprender com os problemas que lhe são apresentados. As aleatoriedades da navegação ativam mecanismos de compensação, que permitem a adaptação às novas necessidades. Por exemplo, um ser humano cujo sistema neural não recebe informações visuais pode suprir essa falta de informação com o auxílio de sinais de outros sensores (audição, tato, ...), adaptando o seu processamento neural. Situação semelhante ocorre com um robô móvel que perde os seus valores de referência de trajetória. O mecanismo executará os movimentos sem ter a certeza da sua validade. A orientação e o posicionamento absoluto do robô são facilmente resolvidos com um sistema de GPS interconectado ao sistema de navegação. Os sistemas robóticos também podem incorporar estas características, tornando-os flexíveis e adaptáveis à dinâmica do deslocamento.

Contudo, existem outros aspectos que necessitam ser considerados. A capacidade de deslocamento do robô, de reconhecimento do ambiente e de desvio de obstáculos fornece ao sistema de navegação um volume grande de informações, permitindo gerar mapas físicos do ambiente. Estes sistemas são, geralmente, dedicados e dificilmente podem ser transferidos para outros sistemas.

Associado ao deslocamento livre de fios e problemas de processamento e armazenamento de informações, podemos utilizar um ambiente cliente/servidor com um sistema de comunicação wireless para atualização do status dos sensores do robô.

Neste artigo analisamos o problema da navegação e desvio de obstáculos para robôs móveis deslocando-se em ambientes não conhecidos. Abordaremos uma solução computacional que permita a auto-evolução e a auto-escolha de regras baseada nos princípios de computação evolutiva e sistemas classificadores [5][6][7][9].

Propomos a arquitetura do sistema em um modelo cliente/servidor, na qual concentraremos todo processamento pesado no servidor, aliviando o sistema embarcado, utilizando recursos de conectividade sem fio para propiciar melhor mobilidade.

## II. SISTEMAS CLASSIFICADORES

Ações de um robô podem ser elaboradas e implementadas a partir de um mecanismo do tipo ação-reação-poderação. Os movimentos do robô são planejados a partir de macro informações (origem e destino, por exemplo) e por informações fornecidas por sensores. Pode-se estabelecer uma macro trajetória, ou uma direção preferencial de movimento, e durante a realização das ações, o sistema de navegação, interagindo com o ambiente, efetua as devidas correções de trajetória e "aprende" as características ambientais. Uma forte ferramenta computacional para abordar este tipo de problema são os sistemas computacionais baseados em redes neurais com supervisão. Para o desvio de obstáculos, estamos estudando a utilização de sistemas classificadores (SC). Estes sistemas permitem a definição, através do treinamento evolutivo e a geração de um conjunto de regras [11][12][13][14][15], a seleção e classificação de ações que envolve certo grau de aleatoriedade.

Para o sistema biológico, um dos principais elementos para o controle das atividades celulares é o código genético estruturado no DNA. Esta estrutura é formada por um conjunto de bases protéicas devidamente organizadas. Utilizaremos uma estrutura semelhante para gerar o conjunto de regras para o sistema classificador.

As regras são formadas por duas partes: identificador (antecedente) e ação (conseqüente) [16][20]:

- **IDENTIFICADOR** - é um vetor de tamanho fixo cujos elementos pertencem ao alfabeto {0,1,#}, onde " #" (don't care) significa que aquela posição poderá assumir tanto o valor 1 quanto o valor 0.

- **AÇÃO** - é um outro vetor de tamanho fixo, os elementos deste vetor pertencem ao alfabeto {0, 1}.

A representação de uma regra é a seguinte: **001##00 : 01**, sendo a parte antecedente separada da parte conseqüente por " : ". Cada elemento (ou uma seqüência) do identificador deve ser relacionado a algum elemento do robô, como por exemplo, às informações de sensores. A parte conseqüente deve ser associada a uma ação (movimento, mudança de status) do robô. A interpretação da regra é semelhante à utilizada em estruturas condicionais do tipo SE/ENTÃO/SENÃO. Se o código IDENTIFICADOR for ativado, então se executa a AÇÃO.

O elemento "0" indica ausência do sinal, o elemento "1" indica presença de sinal e o elemento "#" indica qualquer valor (0 ou 1).

O sistema classificador tem três subsistemas:

- Subsistema de Tratamento de Regras e Mensagens;
- Subsistema de Apropriação de Crédito;
- Subsistema de Descoberta de Novas Regras;

O subsistema de Tratamento de Regras e Mensagens recebe informações (mensagens) dos sensores e efetua a sua codificação, gerando uma parte antecedente específica para um determinado instante de amostragem. Essa codificação é comparada com a parte antecedente de todos os classificadores previamente cadastrados (gerados), os indivíduos compatíveis são escolhidos e são enviados ao Subsistema de Apropriação de Crédito. A comparação ocorre elemento a elemento e se todos os elementos de uma regra existem (ou são coincidentes - "matching" com os dados de entrada), então a regra é selecionada.

No subsistema de Apropriação de Crédito todos os classificadores compatíveis participarão de uma competição. Cada competidor (regra) deverá fazer uma aposta "**Bid**". O classificador que vencer a competição atuará sobre o ambiente.

Todos os classificadores escolhidos pagam uma taxa para participar da competição. O classificador vencedor enviará ao robô uma "ação" (parte conseqüente) que deverá ser executada. Dependendo dos resultados conseguidos pela execução desta ação, o sistema será realimentado e a regra vencedora será punida ou recompensada, alterando o seu nível de energia (strength).

A energia de uma regra indica o seu nível de aceitabilidade para atuação no meio ambiente.

Os classificadores (regras) são criados inicialmente de forma aleatória e para que possam ter um comportamento

adequado, será necessário que o Sistema Classificador descubra quais classificadores geram comportamentos apropriados, criam-se novos classificadores e descartam-se aqueles que sejam ou tornem-se inapropriados, considerando a variabilidade do ambiente. Este mecanismo de aprendizagem de regras é implementado pelo algoritmo de *Atribuição de Créditos* e por um *Algoritmo Genético (AG)* [1][2][3].

O Sistema Classificador [10][11] interage com o ambiente em que atua através da troca de mensagens entre os dados dos sensores e a ação desenvolvida pelo robô. O ambiente manda mensagens sobre seu estado atual para o sistema, e o sistema envia para o ambiente a ação que deve ser executada. Por exemplo: um robô está navegando por um ambiente, encontra um obstáculo, manda esta mensagem para o Sistema Classificador, e este dirá ao robô que direção tomar para evitar o obstáculo. Dependendo do efeito que a ação escolhida teve sobre o ambiente (boa ou má), o conjunto de classificadores responsáveis por aquela ação será premiado ou punido. Desta forma os Sistemas Classificadores conseguem determinar novas regras através de regularidades verificadas no ambiente, e com isto refinar o seu modelo de ambiente.

Após um processo de atribuição de créditos seguido por etapas evolutivas, estes classificadores podem apresentar (incorporar) soluções para o problema proposto. Estes sistemas se guiam por metas implícitas, e promovem a evolução baseada em recompensas ou punições.

O subsistema de Descoberta de Novas Regras gera aleatoriamente novas regras. Adotamos a técnica clássica de algoritmos genéticos. O Algoritmo Genético selecionará os classificadores com os maiores strengths (**fitness**), gerando novos indivíduos por meio da recombinação e mutação destes. Os novos classificadores gerados tomam o lugar dos mais fracos, modificando o conjunto de classificadores do sistema [5].

Neste tipo de abordagem, o ajuste dos parâmetros, a geração de novas regras e sua seleção para a formação de uma lista de mensagens apropriadas para as tarefas em análise são os pontos críticos do processo, exigindo uma etapa de treinamento e simulação.

### III. SIMULAÇÕES

Para as simulações do sistema classificador definimos um robô retangular (o robô que utilizaremos para os testes é retangular) capaz de deslocar-se para frente, para trás, girar sobre o próprio eixo e possui 7 sensores de proximidade na parte frontal. O ambiente de navegação é desconhecido para o sistema classificador, contudo para facilitar a visualização dos movimentos criamos um “macro ambiente” utilizando-se primitivas gráficas como

retas e polígonos, que pode ser observado somente pelo usuário. A atividade operacional básica é deslocar-se de um ponto inicial a um ponto final definidos pelo usuário, no simulador. Este procedimento é realizado clicando-se com o mouse sobre a área do “macro ambiente”, em tempo de execução. O sistema classificador deve conhecer o ponto objetivo da trajetória, contudo, ele não tem qualquer informação sobre o ambiente, a não ser as fornecidas em tempo de execução, pelos sensores do robô. A cada deslocamento do robô efetua-se uma amostragem dos valores dos sensores, formando-se uma mensagem que é enviada ao sistema classificador. Este retorna a ação a ser executada.

O classificador é representado por um vetor de 13 posições, sendo que as 11 primeiras posições representam a parte antecedente formada pelas informações dos sensores (7 sensores) e pela direção preferencial de movimento (4 bits). A parte conseqüente é formada pelos dois últimos bits restantes, correspondendo às seguintes decisões a serem tomadas: mover para frente, girar para esquerda, girar para a direita e girar 180 graus. O ângulo de giro é definido durante as simulações. Realizamos simulações com valores variando de 30° a 110°, sendo que os melhores resultados foram obtidos com giros de 80°, 90° e 110°.

Para as simulações definimos um ambiente com obstáculos fixos e móveis, desconhecidos para o sistema classificador. Utilizamos inicialmente 1000 regras geradas aleatoriamente, posicionamos o robô em um ponto de origem, definimos um ponto final de trajetória e estabelecemos uma direção preferencial de movimento.

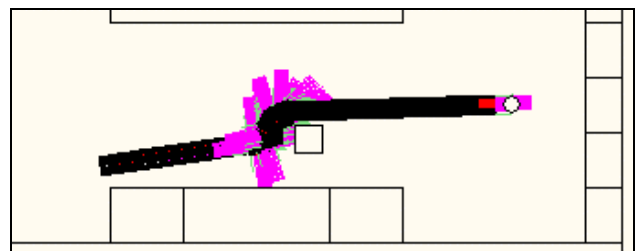


FIGURA 1. Deslocamento do robô em um ambiente desestruturado. O sistema classificador não tem informação prévia sobre o ambiente a não ser as fornecidas pelos sensores durante o deslocamento. A linha contínua indica a trajetória do robô. O círculo indica o ponto final de trajetória.

A cada deslocamento do robô, os procedimentos para a formação de novas regras e execução dos movimentos devem seguir os seguintes procedimentos: formação da mensagem dos sensores, seleção de regras, identificação das que melhor adaptam-se ao movimento, geração do

movimento, interação com o meio ambiente. Para cada ciclo, preservamos 40% das melhores regras, dentre as de maior nível de energia, e então se aplica “crossover” (taxa de 5%) e mutação (taxa de 2%).

Na figura 1 mostramos a área de trabalho, o robô, sua trajetória e, as áreas de amostragem dos sensores.

#### IV. CONCLUSÃO

Os testes realizados mostram que as abordagens evolutivas utilizadas (Algoritmos Genéticos e Sistemas Classificadores) fornecem boa resposta ao problema proposto. Nas simulações o robô conseguiu desviar dos obstáculos e realizar a trajetória especificada. É claro que, no início do processo evolutivo, o robô mostrou-se muito sensível às condições iniciais das regras e ao conjunto de regras selecionadas. Ele apresentou movimentos aleatórios e/ou circulares, não convergindo para a direção preferencial de movimento e não desviando de obstáculos, até que um conjunto de regras mínimo fosse gerado e selecionado. Essa evolução pode ser observada no gráfico da figura 2 onde inicialmente o número de regras selecionadas é inferior a 1%. Com a evolução das regras, observa-se uma maior seleção das mesmas, em torno de 2%.

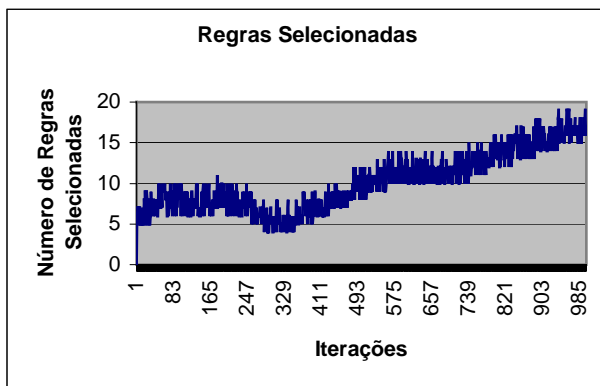


FIGURA 2. Evolução do número total de regras selecionadas durante o movimento de um total de 1.000 regras especificadas.

Nas simulações utilizamos partes do conjunto de regras previamente avaliadas em testes anteriores. A incorporação de novas regras produziu, inicialmente, algum “ruído” no comportamento dinâmico do robô, mas a convergência foi muito mais rápida.

Para trabalhos futuros, faz-se necessária a ampliação do escopo de atuação das regras permitindo a fusão de mais informações provenientes dos sensores e estados internos do sistema, atuando como uma memória de curto e longo

prazo, permitindo o reconhecimento de padrões de movimentos e a repetição destes padrões. Desta forma, as regras também podem ser agrupadas formando cadeias, cuja execução produza comportamentos contextualizados ao robô.

#### BIBLIOGRAFIA

- [1] **Banzhaf, W. ; Nordin, P. ; Keller, R.E. ; Francone, F.D. (1998)** . Genetic Programming An Introduction : On the Automatic Evolution of Computer Programs and Its Applications. Morgan Kaufmann Publishers.
- [2] **Booker, L. B. (1985)**. Improving the Performance of Genetic Algorithms in Classifier Systems. Proceedings of an International Conference on Genetic Algorithms and their Applications, pp. 80-92.
- [3] **Booker, L. B.; Goldberg, D. E. ; Holland, J. H. (1989)**. Classifier Systems and Genetic Algorithms. Artificial Intelligence, vol. 40, pp. 235-282.
- [4] **Castillo, Oscar; Trujillo, Leonardo; Melin, Patricia** Multiple objective Genetic Algorithms for Autonomous Mobile Robot Path Planning Optimization. Applied Computational Intelligence - Proceedings of the 6th International FLINS Conference, Applied Computational Intelligence - Proceedings of the 6th International FLINS Conference, 2004, p 444-449
- [5] **Fong T., Nourbakhsh I., and Dautenhahn K. (2003)**. A survey of socially interactive robots. Robotics and Autonomous Systems, pp. 42:143-166.
- [6] **Goldberg, D. E. (1989)**. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Inc.
- [7] **Holland, J. H. (1975)**. Adaptation in Natural and Artificial Systems : an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. The MIT Press, Ann Arbor, MI.91.
- [8] **Hu, Yanrong; Yang, Simon X.** A knowledge based genetic algorithm for path planning of a mobile robot. Proceedings - IEEE International Conference on Robotics and Automation, v 2004, n 5, Proceedings - 2004 IEEE International Conference on Robotics and Automation, 2004, p 4350-4355
- [9] **Kovacs, T. ; Lanzi, P. L. (1999)**. A Learning Classifier Systems Bibliograph. Technical Report: CSRP-99-19, University of Birmingham, United Kingdom.

<http://www.cs.bham.ac.uk/~tyk/lcs>

[10] **Koza, J. R. (1996)**. Genetic programming : on the programming of computers by means of natural selection. The MIT Press, Cambridge, Massachusetts.

[11] **Koza, J.,R. ; Bennett III, F.H. ; André, D. ; Keane, M.A. (1999)**. Genetic Programming III: Darwinian Invention and Problem Solving. Morgan Kaufmann Publishers.

[12] **Lubnen N. Moussi ; Gudwin, Ricardo R. ; Von Zuben, Fernando J. ; Marconi, K. Madrid (2001)** Sistemas Classificadores com Redes Neurais (NNCS): Aplicação ao controle de um veículo autônomo simulado computacionalmente. V Simpósio Brasileiro de Automação Inteligente- V SBAI, Vol. 1, pp.1-6, Canela, RS, BRASIL,2001

[13] **Michalewicz, Z. (1996)**. Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag Berlin Heidelberg, New York.

[14] **Michalewicz, Z. (2000)**. Evolutionary Algorithms for Constrained Optimization Problems Tutorial Program, Genetic and Evolutionary Computation Conference - GECCO 2000, Las Vegas, Nevada.

[15] **Moussi, Lubnen N., Ricardo R. Gudwin, Fernando J. Von Zuben, Madri, Marconi K. (2001)**. Sistemas Classificadores com Redes Neurais (NNCS): Aplicação ao controle de um veículo autônomo simulado computacionalmente.V Simpósio Brasileiro de Automação Inteligente - V SBAI, Vol. 1, pp.1-6, Canela, RS, BRASIL.

[16] **Richards, Robert A. R.** Classifier Systems & Genetic Algorithms- Chapter 3 Zeroth-order Shape Optimization Utilizing a Learning Classifier System Ph.D. Dissertation, Mechanical Engineering Department, Stanford University.

[17] **Sedighi, Kamran H.; Ashenayi, Kaveh; Manikas, Theodore W.; Wainwright, Roger L.; Tai, Heng-Ming** Autonomous local path planning for a mobile robot using a genetic algorithm. Proceedings of the 2004 Congress on Evolutionary Computation, CEC2004, v 2, Proceedings of the 2004 Congress on Evolutionary Computation, CEC2004, 2004, p 1338-1345

[18] **Sugihara, Kazuo; Smith, John** Genetic algorithms for adaptive planning of path and trajectory of a mobile robot in 2D terrains. IEICE Transactions on Information and Systems, v E82-D, n 1, 1999, p 309-317

[19] **Tu, Jianping; Yang, Simon X.** Source: Genetic algorithm based path planning for a mobile robot. Proceedings - IEEE International Conference on Robotics and Automation, v 1, 2003, p 1221-1226

[20] **Vargas, P. A. ; Lyra, C. ; Von Zuben, F. J. (2000)**. Sistemas Classificadores para Redução de Perdas em Redes de Distribuição de Energia Elétrica. Anais do XIII Congresso Brasileiro de Automática (CBA 2000), Florianópolis, Setembro, pp. 2186-2191.