

Indução de Regras Nebulosas via Sistemas Classificadores para o Controle da Arquitetura em Redes Neurais Construtivas para Navegação Autônoma

Rodrigo Calvo¹, Myriam R. Delgado¹, Heitor S. Lopes¹, Maurício Figueiredo²

¹ CPGEI, CEFET-PR, Av. 7 de Setembro, 3165, CEP: 80230-901 Curitiba, Brasil
rcalvo@cpgei.cefetpr.br, myriam@dainf.cefetpr.br, hslopes@cefetpr.br

² Departamento de Informática / Universidade Estadual de Maringá (UEM-PR)
mauricio@din.uem.br

Abstract

This work describes an evolutionary system to control the growth of constructive neural networks in autonomous navigation systems. We propose a classifier system to generate rules of a Takagi-Sugeno fuzzy system, which controls the architecture of neural networks that changes during learning epochs. The classifier system has a learning mechanism, defined according to the mobile robot performance that modifies the linguistic terms. Experiments show the efficiency of the classifier system in analyzing the networks state and decide to insert a new processing element.

1. Introdução

Redes neurais construtivas são sistemas que, a partir de uma arquitetura mínima, procuram encontrar, de forma automática, uma arquitetura ótima para um determinado problema [3, 9]. Há grandes expectativas para a utilização de métodos construtivos para o controle da expansão da arquitetura de redes neurais. O sistema apresentado neste trabalho adota redes neurais construtivas para projetar um sistema autônomo, capaz de controlar o ajuste de direção do robô.

Métodos baseados em computação evolutiva têm sido propostos como uma alternativa eficiente e robusta para o projeto de redes neurais artificiais. Há uma grande diversidade de abordagens na aplicação de computação evolutiva no projeto de redes neurais. Dentre os trabalhos que empregam o processo de definição automática da arquitetura das redes neurais através de métodos evolutivos, podem ser citados, por exemplo [2, 7, 8].

Sistemas Nebulosos [5] e Sistemas Classificadores [6] são paradigmas da Inteligência Computacional. Este artigo propõe um sistema nebuloso para supervisionar a expansão das redes neurais construtivas que controlam o robô. A dificuldade em determinar as regras do sistema nebuloso e o perfil dos termos linguísticos levou à concepção de um sistema classificador para a geração da

base de regras. O sistema classificador aprende de acordo com o desempenho do robô durante a navegação, ajustando o perfil das funções de pertinência das variáveis que constituem o sistema nebuloso.

2. Sistema de navegação autônoma

O sistema de navegação autônoma, baseado no modelo proposto em [1], consiste de três módulos principais: Desvio de Obstáculos (DO), Busca a Alvo (BA) e Distância ao Alvo (DA), conforme mostra a Figura 1. Estes módulos são conectados a um neurônio de saída, sendo que os módulos BA e DO geram comportamentos instintivos, desprovidos de aprendizagem. O Módulo de Coordenação tem a função de ponderar os comportamentos instintivos gerados pelos módulos inatos BA e DO, após o processo de aprendizagem.

As redes do módulo de coordenação são conectadas a diferentes tipos de sensores: distância de obstáculos, direção de alvos e distância de alvos (Figura 1). A arquitetura das redes neurais (Figura 2) consiste de duas camadas de neurônios nebulosos [5]. A primeira camada é construtiva, ou seja, nos instantes (t_c - instante de colisão; t_a - instante de captura) em que o ciclo de aprendizagem é acionado, alguns neurônios são inseridos na primeira camada (para maiores detalhes da arquitetura, ver [1]).

A todo instante t_c um neurônio é inserido na rede DO, assim como a todo instante t_a um neurônio é inserido nas redes BA e DA. Dessa forma, as redes estão sujeitas a crescer a todo evento de colisão ou captura. Para tratar o crescimento indevido das redes, este trabalho propõe uma abordagem para monitorar suas arquiteturas, que são baseadas na concepção de um sistema nebuloso que decide pela inserção ou não de neurônios nas redes.

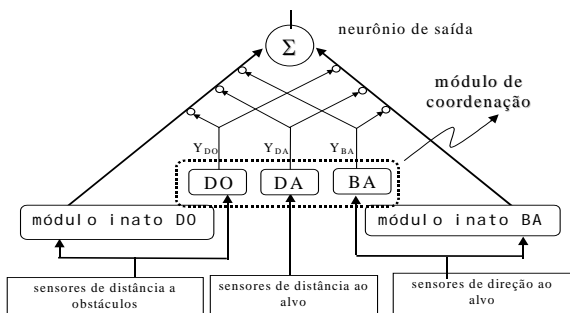


Figura 1. Sistema de navegação autônoma.

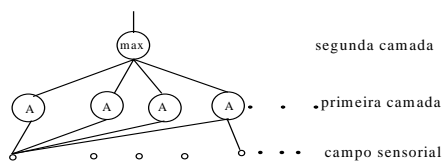


Figura 2. Arquitetura da rede de coordenação.

Neste problema, a elaboração das regras nebulosas torna-se uma tarefa complexa pelo fato de que não é de conhecimento do projetista o efeito das variáveis (ou estado das redes) na arquitetura das redes ao longo das iterações. Deste modo, é utilizado um sistema classificador para a geração das regras e dos perfis das funções de pertinência.

3. Proposta da arquitetura construtiva

A arquitetura construtiva do modelo de coordenação neural deverá ser controlada por um sistema híbrido composto por uma base de regras nebulosas obtida por um sistema classificador.

3.1. O Sistema nebuloso

Algumas variáveis que descrevem o estado (e arquitetura) das redes neurais foram definidas como variáveis de entrada do sistema nebuloso, são elas:

- quantidade de neurônios da rede neural (*Qtde*);
- taxa de crescimento da rede neural (*Taxa*);
- distância (semelhança) entre o padrão (neurônio) de entrada e os padrões (neurônios) armazenados (*Dist*);
- frequência de respostas baixas dos padrões armazenados (*Freq*).

O universo de cada variável foi particionado em três termos lingüísticos, que são associados a funções de pertinência, cujos parâmetros são ajustados ao longo do processo evolutivo. A agregação dos antecedentes das regras do sistema é feita pelo operador MIN. O modelo do sistema nebuloso utilizado é Takagi-Sugeno de ordem zero cujo conseqüente é uma constante (limitada aqui ao intervalo [0..1]).

3.2. Sistema classificador

A teoria de Sistemas Classificadores (SC) é uma das abordagens da computação evolutiva inspirada na evolução dos seres vivos, sendo originalmente introduzida por Holland [6]. Um SC é um mecanismo para criação e atualização evolutiva de conhecimento representado por regras, denominadas classificadores. Diferentemente dos sistemas especialistas, um SC é um mecanismo genérico de aprendizagem que pode ser aplicado em diversos ambientes.

Este trabalho utiliza um sistema classificador para definir regras nebulosas para controle da arquitetura das redes neurais do módulo de coordenação, descrito na seção 2. Tal sistema classificador é constituído por quatro componentes principais: população de regras, atribuição de créditos, mecanismo de geração e módulo de competição (Figura 3).

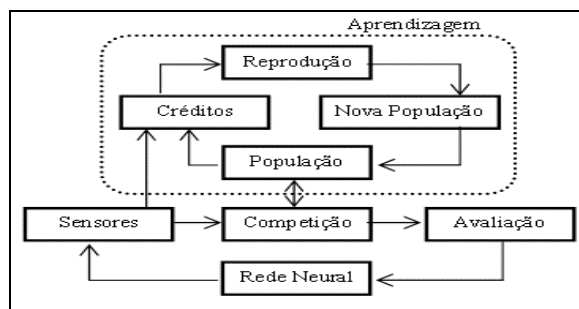


Figura 3. Arquitetura do Sistema Classificador.

A população de regras representa o conhecimento do sistema e evolui durante a navegação do robô. O módulo de competição recebe as informações do estado da rede, que são avaliadas, detectadas pelos sensores (sensores “virtuais” ou dispositivos que obtêm o estado das redes), e compara-os com as regras, definindo qual delas deve atuar no ambiente, podendo realizar mudanças na arquitetura da(s) rede(s). Este processo se repete a cada colisão do robô e captura de alvo, formando um ciclo que somente é interrompido quando se faz necessária uma evolução da população (aprendizagem). Um evento em particular dispara a aprendizagem: o decaimento de desempenho entre a captura de dois alvos consecutivos. Nestes casos as regras são avaliadas e alteradas por meio de operadores evolutivos, produzindo uma nova população.

3.2.1 População de regras

Cada indivíduo (cromossomo) da população representa uma regra nebulosa que compõe a base de regras do sistema classificador. Entretanto, o cromossomo é dividido em três partes, como mostra a Figura 4.

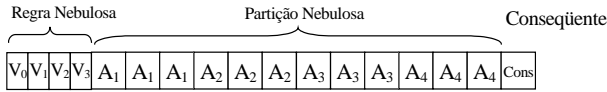


Figura 4. Estrutura do cromossomo.

A parte correspondente à regra nebulosa é composta por quatro alelos, equivalente ao número de variáveis de entrada do sistema nebuloso. Cada alelo representa uma variável de entrada na seguinte ordem: *Qtde*, *Dist*, *Taxa*, *Freq*. A informação contida em cada um dos alelos indica qual termo lingüístico da respectiva variável está presente na regra representada pelo cromossomo. A associação entre os valores assumidos pelos alelos e os termos lingüísticos é definida como: Pequena (0), Média (1) e Alta (2). Considerando somente a parte do cromossomo que corresponde aos termos lingüísticos, a Figura 5 mostra alguns exemplos.

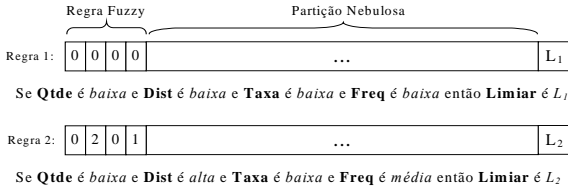


Figura 5. Exemplos de regras fuzzy.

Na partição nebulosa, cada variável é codificada em três alelos (D_{ij} , L_{ij} , R_{ij}) onde i indica o termo lingüístico da variável j , $0 \leq i \leq 2$ e $0 \leq j \leq 3$. Esta codificação representa o perfil da função de pertinência do termo lingüístico indicado na primeira parte do cromossomo. Os alelos assumem valores reais não negativos que são relativos a uma posição de referência no universo. Considerando-se o suporte da função como os pontos no eixo x para os quais a função assume valores maiores que zero, tem-se que D_{ij} é a distância da função de pertinência i para a função de pertinência $i - 1$ (exceto para o caso $i = 1$ em que a distância é medida em relação ao limite inferior do universo); L_{ij} é a distância entre o valor de máximo da função de pertinência i e o primeiro ponto do suporte; R_{ij} é a distância entre o valor de máximo da função de pertinência i e o último valor do suporte. A Figura 6 mostra a codificação da função de pertinência dos termos lingüísticos que compõem um antecedente.

As regras são inicialmente construídas de maneira que ocorram todas as combinações possíveis entre os antecedentes das regras. Sendo o sistema nebuloso constituído por quatro variáveis de entrada e cada uma delas formada por três termos lingüísticos, o número de regras geradas é $3^4 = 81$. Os conseqüentes das regras são gerados aleatoriamente dentro do intervalo $[0..1]$.

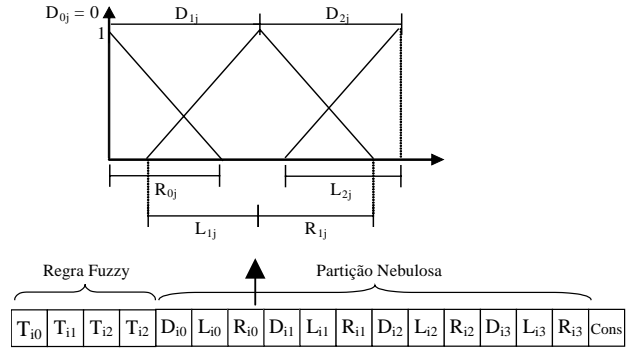


Figura 6. Cromossomo e a codificação da função de pertinência representada na partição nebulosa.

3.2.2 Módulo de competição

A cada instante t_c (para a rede DO) e t_a (para as redes BA e DA), os sensores detectam o estado da(s) rede(s) neural(is). As informações capturadas são enviadas para o módulo de competição. Nesta etapa, todas as regras competem para definir a vencedora, ou seja, aquela cuja parte antecedente é mais semelhante aos estímulos. A semelhança de cada regra r , $S(r)$, é dada pela Equação 1.

$$S(r) = \min_{x=1}^N (\mu_i(x)) \quad (1)$$

onde x é a variável de entrada para o sistema nebuloso, N é o número de entradas para o sistema, $\mu_i(\cdot)$ é o valor da pertinência da variável x para o i -ésimo antecedente da regra r .

3.2.3 Módulo de avaliação

O conseqüente da regra vencedora é enviado para o módulo de avaliação, para que seja comparado com o LIMIAR da rede(s) em questão. Se o conseqüente da regra for maior que o inverso do valor da resposta da(s) rede(s) no exato instante t_c ou t_a , então um novo padrão (de eminente colisão, ou de captura) deve ser inserido na(s) rede(s) alterando sua(s) arquitetura(s), caso contrário a(s) arquitetura(s) permanece(m) inalterada(s).

Independente da inserção de um novo padrão na(s) rede(s), o LIMIAR da(s) mesma(s) é(são) ajustado(s) em função do conseqüente da regra vencedora, como mostra a Equação 2:

$$LIMIAR_{Rede} = LIMIAR_{Rede} + \xi(Cons - LIMIAR_{Rede}) \quad (2)$$

em que *Rede* refere-se às redes neurais, *Cons* é o valor do conseqüente da regra vencedora e ξ é uma constante dentro do intervalo $[0..1]$, em particular $\xi = 0,5$.

3.2.4 Aprendizagem

O processo de aprendizagem, responsável pela evolução da população de regras, ocorre nos instantes t_a (exceto para a primeira ocorrência de captura). O processo de aprendizagem tem início sempre que o desempenho até a captura de um determinado alvo seja inferior ao desempenho até a captura do alvo anterior.

3.2.5 Módulo de atribuição de créditos

Neste módulo, as regras recebem créditos de acordo com o seu desempenho no módulo de competição. O valor $S(r)$ que cada regra adquiriu através da Equação 1 é enviado para o módulo de atribuição de créditos para que seja acumulado com os créditos da regra, adquiridos em competições anteriores. Na primeira ocorrência do módulo de competição após um processo de aprendizagem, o crédito de cada regra é nulo. A atribuição de créditos da regra r é mostrada pela Equação 3. Além de atribuir créditos para as regras no instante t_c ou t_a corrente, este módulo contempla também as regras vencedoras nas duas últimas ocorrências da execução do módulo de competição. As atribuições de créditos para as regras vencedoras da última e penúltima competição são mostradas nas Equações 4 e 5, respectivamente.

$$C_r(t+1) = C_r(t) + S(r) \quad (3)$$

$$C_r(t+1) = C_r(t) + \zeta C_r(t) \quad (4)$$

$$C_r(t+1) = C_r(t) + \delta C_r(t) \quad (5)$$

em que $C_r(t)$ é o crédito da regra r no instante t , onde t é uma generalização para os instantes t_c e t_a , ζ e δ são constantes dentro do intervalo $[0..1]$, tal que $\delta \ll \zeta$, em particular $\zeta = 0,1$ e $\delta = 0,01$.

3.2.6 Módulo de reprodução

Neste módulo é iniciado o processo da geração de uma nova população. Para tanto, é utilizado o método de seleção por torneio com $k=2$ [4]. Os operadores evolutivos tradicionais são aplicados aos indivíduos selecionados.

O *crossover* toma um par destes indivíduos, realiza a recombinação em um ponto e gera dois filhos. Este operador é ativado de acordo com uma taxa de probabilidade, definida em 65%.

Após o processo de *crossover*, cada indivíduo está sujeito a sofrer mutação em cada um de seus alelos com a taxa de probabilidade equivalente a 8%. Este operador altera o valor dos alelos aleatoriamente.

Tanto o *crossover* quanto a mutação respeitam a propriedade de completude das variáveis do sistema

nebuloso, para que não haja qualquer ponto, dentro do universo de cada uma delas, que não seja coberto por nenhum de seus termos linguísticos. Para garantir a validade desta propriedade, a cada ocorrência dos operadores genéticos a cobertura das variáveis é avaliada e, se necessário, as deficiências são reparadas.

A Figura 7 mostra um exemplo da atuação de *crossover*, onde é mostrado apenas o efeito do operador na variável afetada. O segundo e terceiro termos linguísticos da quarta variável (*Freq*) sofrem a atuação do operador.

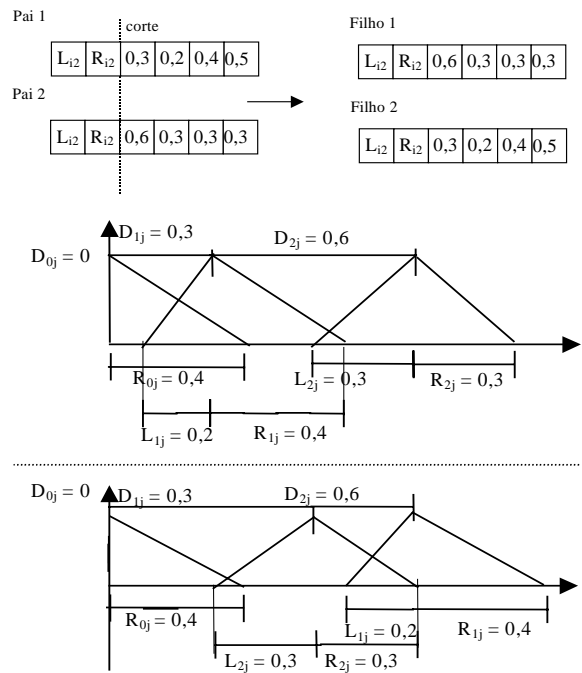
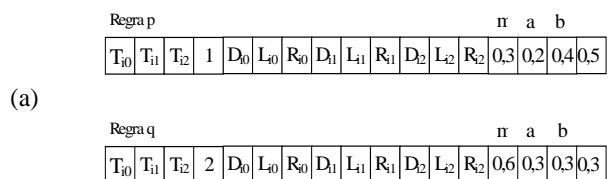


Figura 7. Efeito do operado de *crossover* em uma variável do sistema nebuloso.

A Figura 8 mostra um exemplo da atuação do operador de mutação. Em 8(a) estão os cromossomos que sofrerão mutação. Para simplificar, somente os três últimos alelos correspondentes ao último antecedente da regra são considerados. O exemplo mostra qual o intervalo de valores que os alelos (correspondentes a estes pontos) podem assumir para que não violem a restrição de completude. Em 8(b) é ilustrado o intervalo de valores que cada alelo indicado em 8(a) pode assumir, garantindo a completude do universo.



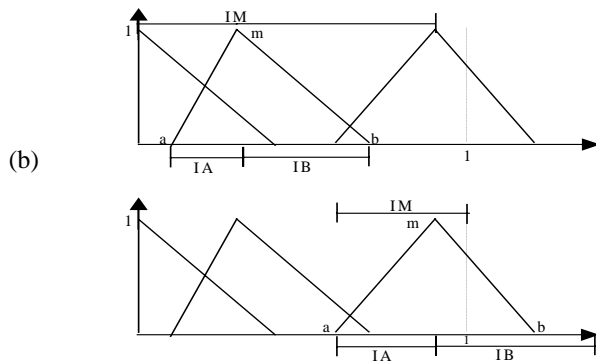


Figura 8. Atuação do operador de mutação.

5. Resultados

Esta seção mostra resultados de simulação do trabalho proposto. A Figura 9 apresenta os ambientes definidos para a avaliação. Os retângulos preenchidos representam os obstáculos, os círculos são os alvos e o triângulo o robô. As posições dos alvos são definidas aleatoriamente. Durante a simulação, somente um alvo está presente no ambiente. Após a captura, o alvo (capturado) é eliminado e um novo alvo é inserido no ambiente. A ordem de captura dos alvos é definida pela ordem em que estes são inseridos no ambiente.

Os experimentos são realizados nos três ambientes da Figura 9 na ordem: Ambiente 1, Ambiente 2 e Ambiente 3. O limiar de cada rede é inicializado em zero, facilitando, portanto, a inserção de neurônios no início da simulação. À medida que ocorrem colisões e capturas a tendência para o limiar é crescer e estabilizar permitindo a inserção de neurônios nas redes com uma certa tolerância.

O primeiro experimento foi realizado sem a atuação do sistema classificador. A todo instante de colisão e de captura, um novo neurônio é inserido na rede DO e nas redes BA e DA, respectivamente, como pode ser observado na Tabela 1. Foram necessárias 15689 iterações para a simulação dos três ambientes.

Tabela 1. Número de neurônios das redes neurais.

Ambientes	Colisões	Capturas	DO	BA	DA
1	13	15	13	15	15
2	19	25	19	25	25
3	19	40	19	40	25

O gráfico da Figura 10(a) mostra o número de neurônios da rede DO em relação ao número de colisões que é linearmente proporcional. Ao final da simulação o desempenho do robô pareceu estável e o conhecimento adquirido em um ambiente foi suficiente para a

navegação nos ambientes posteriores, como mostra o gráfico da Figura 10(b).

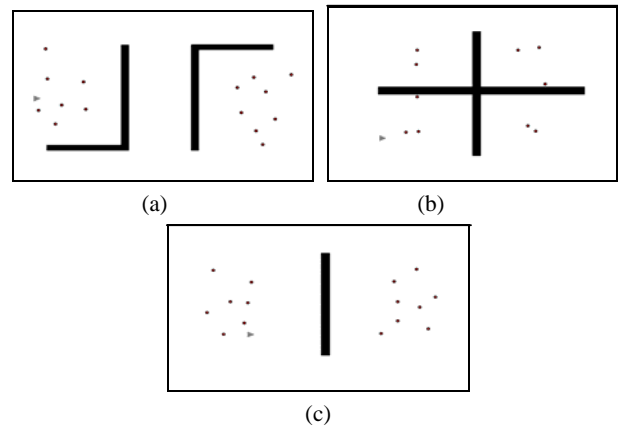


Figura 9. Ambientes utilizados para as simulações: (a) Ambiente 1, (b) Ambiente 2 e (c) Ambiente 3.

O segundo experimento foi realizado com a atuação do sistema classificador. A população inicial foi gerada aleatoriamente (81 indivíduos) e as taxas de *crossover* e mutação foram de 65% e 8%, respectivamente.

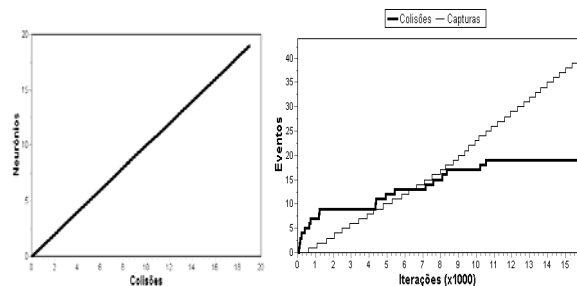


Figura 10. (a) total de neurônios da Rede x o número de colusões e (b) eventos x o número de iterações.

Observa-se na Tabela 2 que o número de neurônios se mostrou reduzido em relação ao número de colisões e de captura. A simulação para os três ambientes precisou de 18552 iterações e o número de ocorrências do processo de aprendizagem para o sistema classificador, ou seja, o número de evoluções foi de 9. O limiar para as redes DO, BA e DA se estabilizaram em 0,59, 0,609 e 0,609, respectivamente.

Tabela 2. Número de neurônios das redes neurais.

Ambientes	Colisões	Capturas	DO	BA	DA
1	9	15	8	5	5
2	10	25	8	5	5
3	12	40	8	5	5

O gráfico da Figura 11(a) mostra o número de neurônios da rede DO em relação ao número de colisões.

Nota-se que, após a oitava colisão, nenhum neurônio foi inserido na rede DO, por decisão das regras do sistema nebuloso.

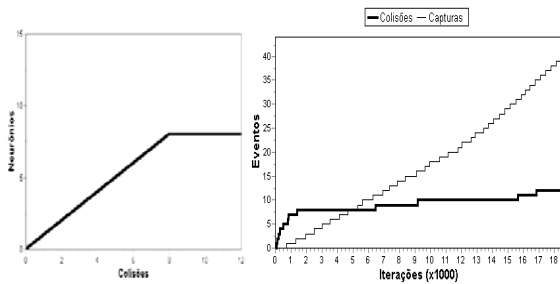


Figura 11. (a) número de neurônios da rede DO x número de colisões, (b) eventos x o número de iterações.

Da mesma forma que o experimento 1, o desempenho do robô se tornou estável e o sistema foi capaz de generalizar o conhecimento adquirido para a atuação em outros ambientes, neste caso, Ambiente 2 e Ambiente 3. Isto pode ser observado pelo gráfico da Figura 11(b), em que a captura se tornou um evento praticamente constante.

A Figura 12 mostra o perfil das variáveis do sistema nebuloso após os processos de evolução.

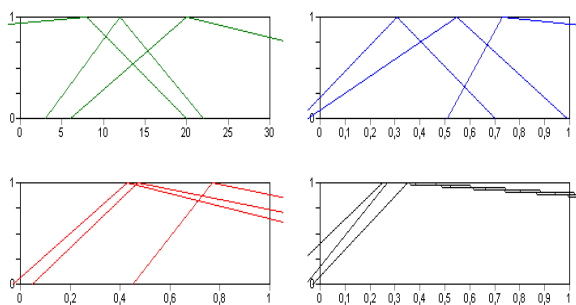


Figura 12. Variáveis do Sistema Nebuloso (da esquerda para direita e de cima para baixo): *Qtd*, *Dist*, *Taxa*, *Freq*.

6. Conclusão e trabalhos futuros

Este trabalho descreveu um sistema evolutivo para o controle do crescimento de redes construtivas em sistemas de navegação autônoma. O sistema de navegação, composto por dois módulos baseados em redes neurais (desvio de obstáculo e busca ao alvo) coordena, a partir de sua aprendizagem, estes dois comportamentos inatos que são conflitantes. Um sistema classificador foi proposto para gerar regras de um modelo nebuloso do tipo Takagi-Sugeno que controla a arquitetura das redes neurais que é alterada a cada instante de aprendizagem do sistema neural. Os resultados de simulação mostraram que a introdução do sistema classificador permitiu controlar o crescimento das redes sem que isto representasse uma queda no desempenho do robô. A característica de

generalização do conhecimento adquirido em um ambiente particular também foi mantida, uma vez que o robô foi capaz de navegar em diferentes ambientes não-treinados com um bom desempenho.

Como trabalho futuro pode ser citada a ampliação do sistema nebuloso no sentido de permitir também a eliminação de neurônios na rede. Outro ponto a ser estudado é a limitação de sobreposição das funções de pertinência na partição do universo com o intuito de melhorar a interpretabilidade do sistema.

6. Referências

- [1] R. Calvo, M. Figueiredo, "Reinforcement learning for hierarchical and modular neural network in autonomous robot navigation," in: *Proceedings of 2003 International Joint Conference on Neural Networks*, IEEE Press, 2003, vol. 1, pp. 1340-1345.
- [2] Z. Chen, J. Xiao, J. Cheng, "PASS: A program for automatic structure search," in: *Proceedings of the International Conference on Neural Networks*, vol. 1, 1997, pp. 308-311.
- [3] E. Eduardo, *Inteligência Computacional no Projeto Automático de Redes Neurais Híbridas e Redes Neurofuzzy Heterogêneas*, Dissertação de Mestrado, UNICAMP, Campinas, 2000.
- [4] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, 1989.
- [5] F. Gomide, W. Pedrycz, *An Introduction to Fuzzy Sets: Analysis and Design*, The MIT Press, Cambridge, 1998.
- [6] J.H. Holland, *Adaptation in natural and artificial systems*, The University of Michigan Press, Ann Arbor, 1975.
- [7] D.E. Moriarty, R. Miikkulainen, "Hierarchical evolution of neural networks," in: *Proceedings of the Conference on Evolutionary Computation*, 1998, pp. 428-433.
- [8] Q. Zhao, "A co-evolutionary algorithm for neural network learning," in: *Proceedings of the International Conference on Neural Networks*, vol. 1, 1997, p. 432-437.
- [9] Fung, W. e Liu, Y.; "Adaptive Categorization of ART Networks in Robot Behavior Learning Using Game-Theoretic Formulation", *Neural Networks*, Vol. 16, no 10, pp. 1403 a 1420, dezembro, 2003.