

Otimização das Funções de Pertinência de um Controlador Nebuloso utilizando Algoritmos Genéticos

Anderson Vinícius de
Medeiros
vinicius@ufrnet.br

André Laurindo
Maitelli
maitelli@leca.ufrn.br

Oscar Gabriel Filho
oscargf@ufrnet.br

LECA – CT – UFRN
59072-970 Natal/RN Brasil

Resumo

Pesquisas recentes têm mostrado que sistemas inteligentes híbridos (unindo técnicas como computação evolucionária e redes neurais) fornecem métodos eficientes para aplicações práticas. Ao se compensar as deficiências de uma técnica com os benefícios de outra, criam-se estruturas de enorme potencial [6,7]. Então, dada a dificuldade na definição de alguns parâmetros dos controladores nebulosos, elaborou-se um procedimento utilizando algoritmos genéticos para otimizar o conjunto das funções de pertinência de um sistema nebuloso (*fuzzy system*), testado na simulação do controle da pressão arterial de um paciente. Tal metodologia atribui um aspecto robusto e elegante à função requerida, comparativamente a outros métodos aplicáveis a problemas dessa natureza.

palavras-chave: Controle Nebuloso, Algoritmos Genéticos, Controle de Pressão Arterial, Otimização.

1. Introdução

Sistemas de controle baseados no conceito de lógica nebulosa (*fuzzy logic*), desenvolvido por Lofti Zadeh em meados de 1960, têm sido utilizados com sucesso em diversas áreas, tais como: eletrodomésticos, indústria automobilística, sistemas de auxílio à tomada de decisão e controle industrial [6]. Nestas aplicações, as técnicas tradicionais podem ter dificuldade e/ou impossibilidade de atingir níveis desejados de eficiência, frente às características intrínsecas ao sistema, devido à necessidade de se agregar conhecimento especializado de operadores humanos, muitas vezes expresso sob formas imprecisas, vagas, ou até mesmo em situações em que o sistema possui dinâmica pouco conhecida.

No entanto, na medida em que a complexidade do processo aumenta, torna-se difícil estabelecer a configuração ideal de alguns dos componentes de um sistema nebuloso. Pode-se, então, aplicar ferramentas cuja função primordial é fornecer, após uma fase dita *aprendizagem*, a solução procurada [7]. Para o caso

em estudo, no qual é simulado o controle da pressão arterial de um paciente, aborda-se a otimização das funções de pertinência para o sistema *fuzzy*. A ferramenta escolhida são os algoritmos genéticos, métodos dinâmicos de busca baseados em mecanismos de evolução e seleção naturais, em metáfora à biologia genética. Neste ponto é possível fazer uso da habilidade de incorporar *a priori* restrições sabidamente necessárias para melhorar o desempenho do controlador nebuloso [6].

Os resultados positivos atingidos ratificam a viabilidade da fusão dessas abordagens em um sistema único inteligente.

2. Lógica e Controle *Fuzzy*

Em oposição à lógica clássica de dois valores, a lógica nebulosa é multivalores, ou seja, atribui a uma afirmação não o valor ‘verdadeiro’ ou o ‘falso’, e sim um grau de veracidade dentro de um intervalo numérico. Além disso, é possível um tratamento das implicações lógicas seguindo regras naturais de raciocínio, analisando condições e estipulando conseqüências [1,3,4]. Seus fundamentos advêm da teoria dos conjuntos nebulosos (*fuzzy sets*), que permite a manipulação de valores incertos, expressões verbais abstratas (e.g. pequeno, próximo, muito rápido, etc.).

No entanto, embora seja possível modelar fenômenos através da lógica *fuzzy*, é requerido um processamento computacional sobre esse modelo. É na etapa de “fuzzificação” ou “nebulização” que tais informações são convertidas em números *fuzzy* para então ocorrer a formulação e execução de uma estratégia de controle. As grandezas provenientes do domínio do mundo real, captadas por sensores, dispositivos computadorizados ou mesmo provenientes de outros segmentos do processo de controle, sofrem essa conversão por meio da definição de um conjunto de variáveis nebulosas (funções lingüísticas de pertinência) que descrevem a entidade no domínio de abrangência. Normalmente utilizam-se mnemônicos para descrever essas

variáveis, por exemplo: variação da pressão, erro, temperatura. Em princípio, as funções de pertinência podem assumir qualquer formato, porém comumente estas são definidas como triangulares (como exibido na figura 1) ou trapezoidais.

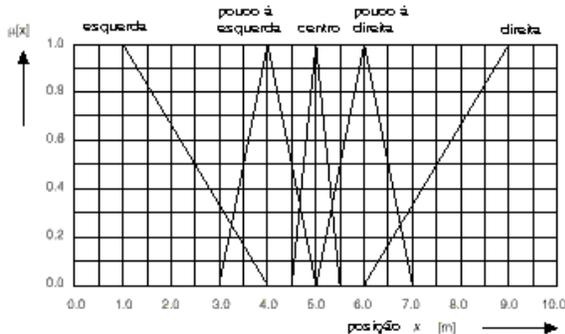


Figura 1: Exemplo de funções de pertinência [3]

A definição satisfatória da quantidade e grau de superposição entre as funções de pertinência é fundamental quando da aplicação de um controlador nebuloso [1]. Ela influencia diretamente na fase seguinte do processo: a inferência ou lógica de tomada de decisão.

Trata-se de um conjunto de regras (base de conhecimento) do tipo *se ... então ...* que descrevem a dependência entre as variáveis lingüísticas de entrada e a de saída. Essa relação geralmente é determinada heurísticamente.

A inferência consiste de dois passos: Agregação, quando se avalia a parte *se ...* de cada regra, através do operador “AND nebuloso”, o qual fornece o menor (*min*) dos dois valores de pertinência (μ) comparados, indicando quão adequadamente cada regra descreve o estado atual do sistema; e a etapa de composição, ponderando as diferentes conclusões das regras ativas sob o operador “OR nebuloso”, que prove o maior dos parâmetros (*max*) [1, 3, 4].

Por outro lado, após a inferência da ação a ser tomada, necessita-se de uma tradução do valor lingüístico para a variável numérica de saída, que pode representar funções como ajustar a posição de um botão ou acionar uma válvula. Este passo é conhecido como “defuzzificação” ou desnebulização, como pode acontecer de surgirem saídas distintas num mesmo momento, com diferentes graus de pertinência, deve-se encontrar o melhor compromisso entre essas ações contrastantes. Um método que atinge essa meta é o “centro da área”, porém este requer muito esforço computacional, pois surgem em sua definição integrais que deve ser calculadas [3].

Uma segunda abordagem (adotada neste trabalho), consiste em calcular a média ponderada entre os valores de cada termo da variável lingüística

de saída, segundo a equação (1), onde u_s é a saída de controle final e os $u_{i,s}$ são as saídas intermediárias [3].

$$u_s = \frac{\sum_{i=1}^m m_i u_{i,s}}{\sum_{i=1}^m m_i} \quad (1)$$

3. Algoritmos Genéticos (AG's)

Baseados nas teorias da Evolução Darwiniana (Charles R. Darwin, 1809-1882) e da Seleção Natural (Gregor Mendel, 1822-1884), têm por objetivo encontrar o indivíduo ótimo de uma população geneticamente refinada. Esse modelo faz uma analogia computacional com os critérios probabilísticos da seleção e evolução naturais, por meio de simulações de gerações que se sucedem, formando assim uma população final mais adequada, ou adaptada, ao sistema em questão [5].

Para tanto, primeiramente é feita uma representação cromossômica dos indivíduos da população. Os cromossomos são compostos por genes, dígitos alfanuméricos, que serão, tal como na biologia, alterados quando da reprodução. Cada indivíduo é uma possível solução do problema. No caso estudado, elabora-se uma representação das funções de pertinência por seus pontos notáveis (como são funções triangulares, armazena-se a informação do vértice e das duas semi-bases; ver figura 2), com a simplificação do uso de simetria em relação ao eixo das ordenadas.

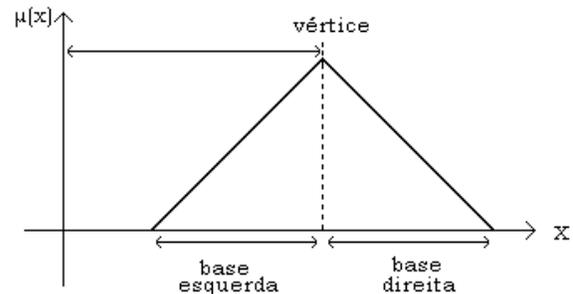


Figura 2 : Parâmetros otimizados

O algoritmo inicia seu desenvolvimento criando aleatoriamente a primeira população. A partir daí, é calculada a adaptabilidade relativa de cada indivíduo ao sistema. Frente a esse cálculo, são selecionados (também por sorteio, mas obedecendo critérios probabilísticos) os elementos que pertencerão a geração seguinte. Para completar a população, efetua-se a reprodução dos pais selecionados, implementando-se, então, operadores genéticos, como Recombinação (*Crossover*), Mutação e Inversão.

Cada operador genético tem uma taxa de ocorrência apropriada, em metáfora à biologia. O processo se repete até que uma condição de parada especificada seja atendida. Essa condição pode se basear na estagnação da evolução da adaptabilidade média da população, na repetibilidade do melhor indivíduo, ou mesmo num limite de gerações pré-fixado [5].

As vantagens no uso dos AG's para a otimização podem ser vistas na não necessidade de testar efetivamente cada combinação de pontos de um domínio em busca do melhor resultado (busca exaustiva). Dado o aspecto dinâmico da evolução, simultâneos pontos do espaço são verificados, explorando a existência de um máximo global para a solução do problema. Além disso trata-se cada indivíduo sob o ponto de vista de sua aptidão para ser considerado solução do problema, e não através de derivadas ou de informações correlatas [5,6]. Conjectura-se então que a adaptabilidade relativa de um indivíduo é parametrizada pelo erro quadrático em uma simulação.

4. Sistema de Controle de Pressão Arterial

Para verificar a eficiência do conjunto de funções de pertinência encontradas durante o desenvolvimento do algoritmo genético, efetuou-se a simulação, via *software*, do controle *fuzzy* da pressão arterial de um paciente no período pós-operatório. O sistema deveria realizar a supervisão da variação da pressão, sob certas condições iniciais, estabilizando em um valor determinado.

Como configuração de partida, supõe-se que a pressão arterial é 150 mmHg. Deseja-se alterar esse valor para 100 mmHg, através da infusão de um medicamento vaso-dilatador (Nitroprussiato de Sódio), o qual atua no organismo de modo a baixar a pressão. Porém, a quantidade de infusão deve obedecer certas restrições fisiológicas, neste caso, o domínio da aplicação varia entre 0 e 180 ml/h. Assim, quando a pressão aumenta (ou permanece num nível elevado), é efetuada uma infusão de acordo com o erro entre o valor atual da pressão e o valor de referência. Caso contrário, como não existe uma infusão negativa, suspende-se a infusão e aguarda-se o aumento natural da pressão.

O modelo matemático do comportamento da pressão arterial média de um paciente sob a infusão do nitroprussiato de sódio é dada por, [2],

$$PAM = P_0 - \Delta P(t) + P_d(t) + v(t) \quad (2)$$

onde PAM é a pressão arterial média, P_0 é a pressão arterial inicial e ΔP é a variação da pressão devido ação do nitroprussiato de sódio, P_d é a reação do organismo à droga e $v(t)$ é o ruído estocástico.

O modelo determinístico contínuo que descreve a relação entre a variação da pressão arterial e a taxa de infusão do medicamento é:

$$\Delta P(s) = \frac{Ke^{-T_i s} (1 + ae^{-T_c s})}{1 + ts} I(s) \quad (3)$$

onde ΔP é a variação da pressão, I é a taxa de infusão, K é a sensibilidade ao fármaco, a é a constante de recirculação, T_i é o atraso de transporte inicial, T_c é o tempo de atraso da recirculação e t é uma constante de tempo do sistema.

O modelo matemático discretizado correspondente a esse processo é dado por:

$$\Delta p(t) = \frac{q^{-d} (b_0 + b_m q^{-m})}{1 - a_1 q^{-1}} I(t) \quad (4)$$

ou seja,

$$y(k) - a_1 y(k-1) = b_0 u(k-d) + b_m u(k-m-d) \quad (5)$$

onde os parâmetros b_0 , b_m , a_1 , d e m são obtidos através da versão discretizada da equação (3) e $y(k)$ corresponde à variação da pressão. O ruído estocástico foi desprezado neste trabalho.

Considerando um intervalo de amostragem de 15 segundos, a_1 , b_0 , d , b_m , e m têm seus valores mostrados na tabela 1, [2]. A duas variáveis foi dada maior atenção, d e m , pois representam os retardos da ação do controlador. Assim, realizou-se simulações variando esses parâmetros dentro do intervalo de 2 até 5, [2].

| Variável | Valor Nominal |
|----------|---------------|
| A_1 | 0.741 |
| B_0 | 0.187 |
| D | 3 |
| B_m | 0.075 |
| M | 3 |

Tabela 1 : Parâmetros da Evolução da Pressão Arterial

5. Resultados

Em princípio, foi implementado um método puramente iterativo na busca da melhor definição das funções de pertinência, avaliando um conjunto restrito de combinações. Sua representação computacional consistia de um vetor com os vértices das funções (triangulares) e um parâmetro que indicava a largura de suas bases. Diversificava-se essa largura, dentro de limites coerentes, mudando em

cada passo o grau de superposição das funções. De forma análoga procedeu-se com os vértices das funções de pertinência. Mantendo-se fixa a largura da base das funções, estes deixaram de ser constantes. Através de um fator multiplicativo realizava-se toda a simulação, desde a situação em que os vértices estavam bem próximos até aquela na qual estavam distantes. O próximo passo foi variar simultaneamente a abertura da base das funções de pertinência e a posição de seus vértices. Ao final do processo, apenas identificava-se qual configuração apresentou o menor erro.

Haja visto essa não seria a melhor solução para o problema de otimização das funções *fuzzy*, procurou-se outro enfoque para sua determinação: os algoritmos genéticos, [6, 7].

Assim, criou-se uma população inicial de indivíduos que continham a definição de todo o conjunto de funções de pertinência. Cada indivíduo da população representava uma configuração possível das funções de pertinência, como exposto anteriormente. Além disso, estabeleceu-se a exigência de que não existisse região, dentro dos limites estudados, sem alguma função definida, i.e. obrigatoriamente haveria superposição de funções. Porém, num dado ponto, não mais de duas funções poderiam estar ativas. Essas restrições exibem a possibilidade de o sistema incorporar conhecimento prévio do caminho em busca de uma solução, [6].

As duas variáveis linguísticas otimizadas no algoritmo genético foram o *erro* e a *variação do erro*. Cinco funções de pertinência descreviam as possíveis qualificações dessas variáveis, segundo os termos linguísticos: *muito negativo*, *negativo*, *zero*, *positivo* e *muito positivo*. Como simplificação, adotou-se a equivalência entre as formas das funções das duas variáveis, porém as escalas entre elas eram diferentes.

A variável de controle, ou de saída, correspondendo à infusão aplicada, tinha por funções de pertinência pulsos unitários (*singletons*, com sete ações distintas), [3]. A tabela 2 exibe o conjunto das regras de inferência finalmente adotado, após uma avaliação e reestruturação do criado inicialmente.

| | | Erro | | | | |
|------------------|----|------|-----|-----|-----|-----|
| | | MN | N | Z | P | MP |
| Variação do Erro | MP | Z | Z | PP | MDP | IM |
| | P | Z | Z | MPP | PP | P |
| | Z | Z | Z | MPP | PP | MDP |
| | N | Z | MPP | MPP | PP | MDP |
| | MN | Z | MPP | PP | PP | MDP |

Tabela 2 : Regras de inferência

Na tabela, Z = zero, MPP = muito pouco positivo, PP = pouco positivo, MDP = médio

positivo, P = positivo, MP = muito positivo, IM = infusão máxima, N = negativo e MN = muito negativo.

Com todos os parâmetros definidos, acionou-se o processo de otimização usando o AG. A cada geração, realizava-se uma simulação, com a mesma configuração inicial e durante o mesmo período de tempo (neste caso, um número fixo de iterações), com todos os indivíduos da população, associando a cada um deles um rótulo numérico: a adaptabilidade relativa (como visto, baseada no erro quadrático). Deste modo, quanto menor fosse o erro durante todo um desenvolvimento do processo, maior seria a adaptabilidade do indivíduo em questão e, devido ao modo como era feita a seleção, maior era a probabilidade desse elemento permanecer na população futura.

Assim, foi feita uma seleção dos indivíduos mais aptos, por meio do método estocástico *roulette wheel*, ou roleta, [5]. Consiste em dividir um intervalo em regiões proporcionais à adaptabilidade relativa, realizando-se o sorteio de um número nesse intervalo e selecionando-se o indivíduo correspondente. Como normalmente ocorrem repetições, e apenas uma amostra de cada indivíduo pode pertencer à população seguinte, o restante da população é gerado através da “reprodução” entre os selecionados. Adotou-se o método de recombinação (*crossover*): dados dois pais, gera-se um filho cujo material genético é uma combinação de seqüências das cadeias de seus ancestrais [5, 7].

Assim a população evoluía até convergir para uma solução (figura 3). O critério de convergência foi a estagnação da evolução: verificava-se a diferença entre as adaptabilidades de duas populações consecutivas. Se essa diferença for menor que um nível estabelecido (10^{-4}), incrementava-se um contador de estagnação. Quando essa variável alcançasse o valor 20, o processo era finalizado. Então, selecionava-se o melhor indivíduo da última população como configuração ótima das funções de pertinência.



Figura 3 : Evolução da população (23 gerações)

Após sucessivas execuções do programa otimizador implementado, observou-se acentuada semelhança entre as funções obtidas, mesmo frente ao aspecto de aleatoriedade (parcial, pois é combinado

com critérios probabilísticos) do sistema evolutivo. Além disso, exceto em casos espúrios, todas as soluções apresentavam um valor de adaptabilidade relativa do melhor indivíduo praticamente igual. Outra característica observada foi a configuração da função mediana (termo lingüístico *zero*) cobrindo uma grande área do domínio, significando que, na região de estabilização do sistema, mais acurada deve ser a ação do controlador (menos influência de outras regras). Uma solução encontrada para a variável *erro* é mostrada na figura 4.

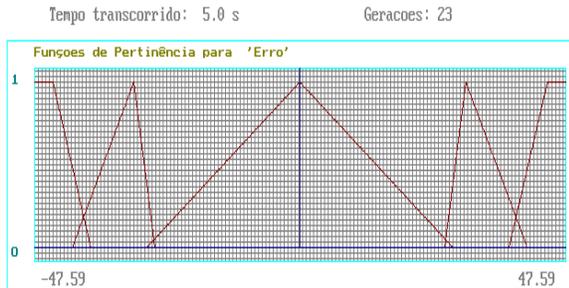


Figura 4 : Funções de pertinência para o erro

A figura 5 mostra o resultado da simulação do controlador fuzzy com o melhor conjunto de funções de pertinência, para um paciente com $m=d=3$. Ocorre uma rápida subida do valor requerido, um pequeno sobre-sinal, da ordem de 1,5%. Para a simulação foram feitas 100 iterações, observando-se a estabilização de $y(k)$ antes da vigésima quinta iteração.



Figura 5 : Desenvolvimento da variação da pressão

Tão bom resultado deve levar em conta o aspecto de que como trata-se de uma implementação totalmente via *software*, não surgiram erros devido a perturbações externas, como ruído na captação dos dados.

O sinal de controle, representando a infusão de nitroprussiato de sódio, apresentou um comportamento próximo ao esperado, qual seja, um valor elevado no início do processo e posterior estabilização em sincronia com a variável analisada, $y(k)$. O gráfico do valor da infusão por iteração é mostrado na figura 6.

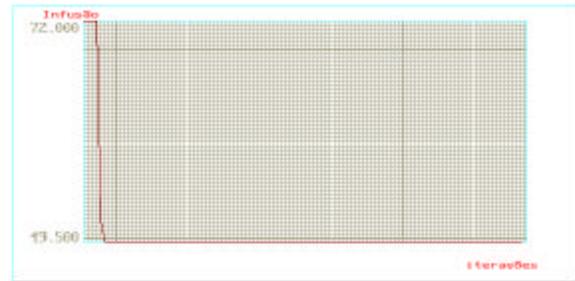


Figura 6 : Infusão x tempo

Efetuada alteração dos parâmetros de atraso da ação do controlador (constantes D e M), observou-se pequena modificação dos resultados. Ao se diminuir o atraso, obteve-se uma evolução mais rápida, porém com um sobre-sinal da ordem de 4,3%. Ao se aumentar o atraso, o sistema evoluiu mais lentamente, gerando um sobre-sinal alto (entre 8% e 10%). Em ambos os casos, após a estabilização o erro foi também muito pequeno.

Tendo em vista que as constantes de atraso (D e M) caracterizam os atrasos na reação do paciente à infusão, foi feito o seguinte experimento: com as funções de pertinência encontradas para a situação $D = M = 3$, verificou-se o comportamento de um indivíduo para o qual $D = M = 5$. Isso equivale a “treinar” o sistema para um paciente e analisar se ele é aplicável a outro. Nota-se, no entanto, que os resultados são menos eficientes. Conforme as figuras 7 e 8, a aplicação de uma solução ótima num paciente diferente não produz resultados idênticos. Mesmo assim, o controlador fuzzy sofre uma perda de desempenho menor que a sofrida por um controlador PID, quando mudam as características do paciente.

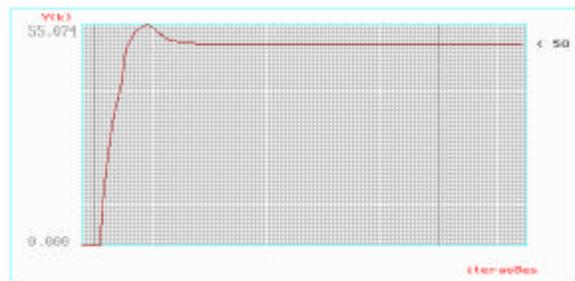


Figura 7 : Otimização com $M = D = 5$

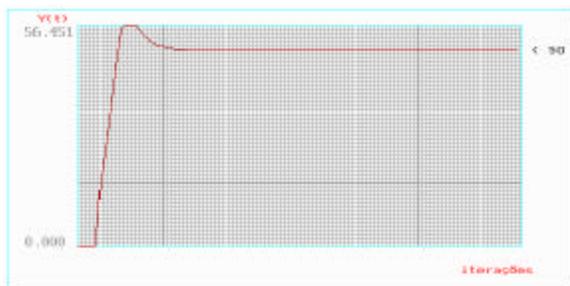


Figura 8 : Otimização com $M = D = 3$ e simulação num paciente para o qual $M = D = 5$

Finalmente, verifica-se que para esse sistema pequenas alterações na base de regras de inferência, ou nas funções de pertinência do sinal de controle acarretam variações no comportamento da variável analisada (variação da pressão). A tabela 3 mostra a definição da variável de controle que apresentou melhores resultados.

| Índice da função | Termo lingüístico | Infusão (ml / h) |
|------------------|-------------------|------------------|
| 0 | Z | 0 |
| 1 | MPP | 49,5 |
| 2 | PP | 51,5 |
| 3 | MDP | 72 |
| 4 | P | 100 |
| 5 | MP | 140 |
| 6 | IM | 180 |

Tabela 3 : Variável lingüística sinal de controle (infusão)

Em comparação com o primeiro método adotado, busca exaustiva fornecendo a mesma solução, o tempo para encontrar a configuração ideal foi sensivelmente menor.

Os resultados com o controlador *fuzzy* são melhores que os apresentados por controladores clássicos do tipo PID e controladores adaptativos, principalmente quando temos variação nos atrasos de transporte, [2].

6. Conclusões

A crescente adoção de sistemas baseados em lógica nebulosa na solução de problemas de engenharia confirma a eficiência desse enfoque, [4]. Pesquisas em todo o mundo, no entanto, surgem objetivando explorar ainda mais esse campo de potencial eminente, de forma a facilitar/possibilitar, e otimizar, a definição dos componentes de um controlador nebuloso. Métodos que integram mais de um tipo de sistema inteligente, unindo os méritos de um para compensar deficiências isoladas, são ferramentas poderosas para superar esses desafios [6,7]. Verifica-se que a utilização de algoritmos

genéticos realmente é uma técnica eficiente para determinar parâmetros de um *fuzzy logic controller*, fornecendo resultados extremamente satisfatórios.

Mesmo considerando o ambiente ideal de simulação do controlador, estima-se que abordagens análogas podem ser usadas para aplicações práticas em outras áreas da ciência. De fato, em 1991, já se lançava no Japão os primeiros produtos utilizando a combinação de sistemas nebulosos e redes neurais artificiais (*neurofuzzy systems*) direcionados aos consumidores domésticos, e em 1994 surgiam produtos domésticos coreanos aplicando algoritmos genéticos [7].

As perspectivas para trabalhos futuros relacionam-se à otimização de todo o sistema de controle nebuloso, desde a etapa alcançada, abrangendo as funções de pertinência das variáveis internas do controlador, até a configuração de todo o conjunto das regras de inferência e respectivos sinais de controle, de modo a solucionar problemas apenas parcialmente conhecidos, ou mesmo aqueles para os quais não existe nenhuma base heurística disponível.

Referências

- [1] Shaw, Ian S. e Simões, M. G., *Controle e Modelagem Fuzzy*. FAPESP, Editora Edgard Blücher LTDA, 1999.
- [2] Pajunem, Grazyna A., Steinmetz, Michael e Shankar, Ravi, *Model Reference Adaptive Control with Constraints for Postoperative Blood Pressure Management*. IEEE Transactions on Biomedical Engineering, Vol. 37, n° 7, pp. 679-682, 1990.
- [3] Fuzzy Logic for G Toolkit, *Reference Manual*. National Instruments, 1997.
- [4] Tsoukalas, L. H. e Uhrig, R. E., *Fuzzy and Neural Approaches in Engineering*. A Wiley-Interscience Publication, 1997.
- [5] Gabriel F., Oscar, *Inteligência Artificial*. Em revisão.
- [6] Jain, L. C. e Jain, R. K., *Advances in Fuzzy Systems – Applications and Theory*. Vol. 11 - *Hybrid Intelligent Engineering Systems*. World Scientific, 1997.
- [7] Ruan, D., *Intelligent Hybrid Systems – Fuzzy Logic, Neural Networks and Genetic Algorithms*, cap. 1 e 11, Kluwer Academic Publishers, 1997.
- [8] Maitelli, A. L. & Yoneyama, T. Suboptimal dual adaptive control for blood pressure management. *IEEE Transactions on Biomedical Control*. Vol. 44 n° 6, pp. 486-492, 1997.