POSICIONAMENTO DE UM PÊNDULO INVERTIDO USANDO ALGORITMOS GENÉTICOS

José Homero Feitosa Cavalcanti[a], Pablo Javier Alsina[b], Edilson Ferneda[c]

[a] UFPB / CCT / Departamento de Engenharia Elétrica, homero@dee.ufpb.br
[b] UFRN / CT / Departamento de Engenharia Elétrica / LECA, pablo@leca.ufrn.br
[c] UFPB / CCT / Departamento de Sistemas e Computação, edilson@dsc.ufpb.br
Av. Aprígio Veloso, 882 — 58.109-970 Campina Grande, PB

Resumo: Neste trabalho propõe-se um Sistema de Controle Inteligente (SCI), usando Redes Neurais Artificiais, Lógica Nebulosa e Algoritmos Genéticos. São apresentados resultados experimentais obtidos do posicionamento de um pêndulo invertido utilizando o SCI proposto. Algoritmos Genéticos são usados para gerar diferentes parâmetros do controlador neural e posicionar, com sucesso, o pêndulo invertido com peso desconhecido na posição de equilíbrio instável.

Palavras-chave: Controle inteligente, Lógica Nebulosa, Algoritmos Genéticos.

Abstract: In this work, an Intelligent Control System (ICS) based on Neural Networks, Fuzzy Logic and Genetics Algorithms is proposed. Experimental results obtained from the positioning of an inverted pendulum with the proposed ICS are presented. Genetic Algorithms are utilized for generating the different parameters of a neural controller, in order to successfully positioning the inverted pendulum of unknown weight at its unstable equilibrium point.

Keywords: Intelligent Control, Fuzzy Logic, Genetics Algorithms

1 INTRODUÇÃO

Na década de 50, K. S. Fu introduziu o termo "Controle Inteligente" (Shoureshi, 1991) ou "Sistema de Controle Inteligente" (SCI). SCI é um sistema que possui a habilidade de sentir o seu ambiente, processar as informações para reduzir as incertezas nos parâmetros do processo, planejar, gerar e executar ações de controle. Um SCI deve ser capaz de distribuir as tarefas de decisão entre um conjunto de executores de tarefas, utilizando intensivamente os computadores disponíveis para inferir o estado atual do sistema e detectar mudanças no seu estado interno e na sua circunvizinhança. Um SCI se diferencia de sistemas convencionais por sua habilidade de tomar decisão e por sua capacidade de aprendizagem. Isto permite ao SCI inferir sobre a dinâmica do processo de uma forma adaptativa e preditiva. Os SCI são necessários em sistemas *(a)* com variações na sua circunvizinhança, *(b)* onde ocorrem mudanças nos modelos de referência, *(c)* que utilizam diferentes critérios de desempenho, e *(d)* sistemas que podem ser sujeitos a falhas de componentes. Um SCI pode ser implementado com técnicas usuais de controle (controladores lineares do tipo PID e controladores adaptativos (Äström, 1989) acoplado a técnicas de inteligência artificial (neurônios artificiais, conjuntos nebulosos ou sistemas especialistas) e utilizando microcomputadores para o controle digital e a gerência do sistema. Alguns Sistemas de Controle Inteligente utilizando Redes Neurais artificiais Multi Camadas (RNMC), Regras Nebulosas (Zadeh, 1988) e Algoritmos Genéticos, estão sendo empregados no controle adaptativo de sistemas não-lineares (Cavalcanti, Lima e Deep, 1994; Cavalcanti, 1995, Cavalcanti, 1996).

A Computação evolucionária possui três ramos principais: Algoritmo Genético (AG), Estratégia Evolucionária (EE) e Programação Evolucionária (PE) (Fogel e Fogel, 1994). Recentemente, Cavalcanti e Ferneda (1995) propuseram três diferentes estratégias de controle para um SCI e apresentaram resultados experimentais obtidos do posicionamento de um pêndulo invertido na sua posição de máxima instabilidade. Os autores usaram um sistema híbrido composto de um controlador neural direto e Lógica Nebulosa. Posteriormente, foi proposto o uso das técnicas de computação evolucionária para chaveamento, em tempo real, entre as diferentes estratégias do SCI (Cavalcanti e Sales, 1996). O novo controlador foi denominado de SCI Evolucionário (SCIE).

Neste trabalho, descreve-se um novo SCIE para um pêndulo invertido, baseado em Controlador Neural, Lógica Nebulosa e Algoritmo Genético. Na Seção 2, o pêndulo invertido é descrito. Na Seção 3 apresenta-se o controlador neural. As regras para controle do pêndulo invertido são descritas na Seção 4. O algoritmo de controle por meio da RNMC para posicionamento do pêndulo invertido é apresentado na Seção 5. Na Seção 6, é apresentado o esquema de chaveamento entre as estratégias de treinamento associadas aos diferentes estados do pêndulo. O SCIE proposto é descrito na Seção 7. Na Seção 8, detalha-se o esquema proposto para adaptação de parâmetros, baseado em AG. Na Seção 9, alguns resultados experimentais são apresentados, mostrando o posicionamento com sucesso do pêndulo invertido para cargas desconhecidas. Na Seção 10, a partir da análise dos resultados experimentais obtidos, são apresentadas as conclusões finais e sugestões para trabalhos futuros.

Artigo submetido em 10/03/97
1a. Revisão em 22/01/98; 2a. Revisão em 14/08/98;
Aceito sob recomendação da Eda. Consa. Profa.Dra Sandra Aparecida Sandri

2 O PÊNDULO INVERTIDO

Na Fig.1 é apresentado esquematicamente um pêndulo simples. O comportamento dinâmico do mesmo é descrito matematicamente pela equação diferencial não linear:

$$Tl(t) = ML^2 \frac{d^2\theta(t)}{dt^2} + PLsen(\theta(t)) \quad (1)$$

onde,

- t: Tempo na forma contínua
- Tl(t): Torque de carga.
- P: Peso do pêndulo.
- M: Massa do pêndulo
- L: Comprimento do pêndulo.
- θ(t): Ângulo entre o pêndulo e a vertical.

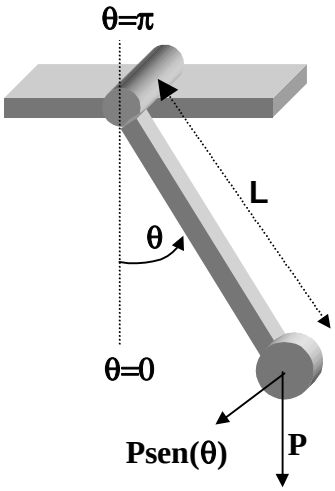

Fig.1 - Pêndulo Simples

Na Fig.2 é mostrado o esquema da montagem de um braço rígido, com um peso (na extremidade), acoplado ao eixo do motor de corrente contínua (na base). Nesta figura são mostradas 4 posições especiais do braço do pêndulo, representadas pelos ângulos entre o braço do pêndulo e a normal. Um pêndulo é dito ser um pêndulo invertido se é permitido o seu posicionamento em $\theta = \pi$ rad (pêndulo nos quadrantes superiores). O protótipo experimental utilizado neste trabalho consiste de um pêndulo com comprimento L = 0.01 m e massa M = 0.05 Kg que, para determinados valores do ângulo θ, apresenta torque Tl maior que o atrito viscoso do eixo do motor.

3 O CONTROLADOR NEURAL

Na Fig.3 mostra-se o controlador neural usado no SCIE (Narenda e Parthasarathy, 1990). Assume-se que $\theta_r(t+1)$ representa o valor de referência ou o valor desejado para a posição $\theta(t+1)$ do eixo do motor, onde $\theta_r(t+1)$ e $\theta(t+1)$ são representados em P.U. com valor de base π rad. A tensão de armadura U(t) é representada em P.U. com valor de base 12V. A RNMC da Fig.3 possui 4 neurônios do tipo linear (L) na entrada, seus valores de entrada sáo representados por X_i, com $X_1 = \theta_r(t)$; $X_2 = \theta(t)$; $X_3 = \theta(t-1)$; $X_4 = U(t)$, 4 neurônios do tipo sigmóide (S) (Eq.2) na camada oculta e um neurônio do tipo tangente hiperbólico (T) (Eq.3) na camada de saída. Nos neurônios, Θ representa a intensidade, β a declividade, e δ o deslocamento da sigmóide. A curva sigmóide com Θ = 1, β = 1 e δ = 0 é ilustrada na Fig. 4.

$$S(X_i,\Theta,\beta,\delta) = \frac{\Theta}{[1+\exp(-\beta(\Sigma X_i * Win_i + \delta))]} \quad (2)$$

$$T(X_i,\beta,\delta) = 2*S(.)/\Theta - 1 = \frac{[1-\exp(-\beta(\Sigma Y_i * Wout_i + \delta))]}{[1+\exp(-\beta(\Sigma Y_i * Wout_i + \delta))]} \quad (3)$$

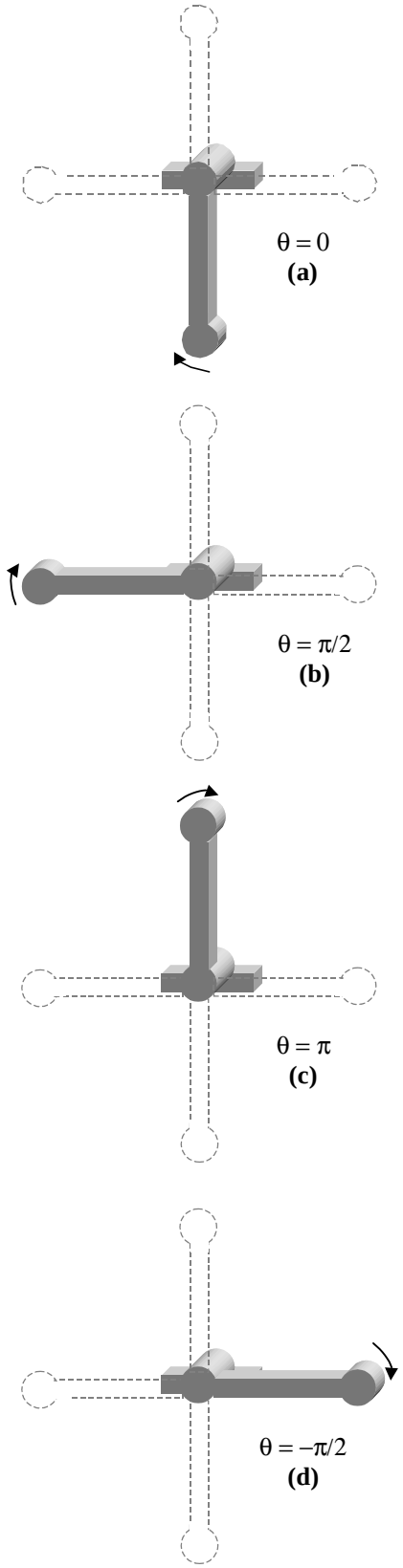

Fig.2 - Posicionamento do Eixo do Motor CC

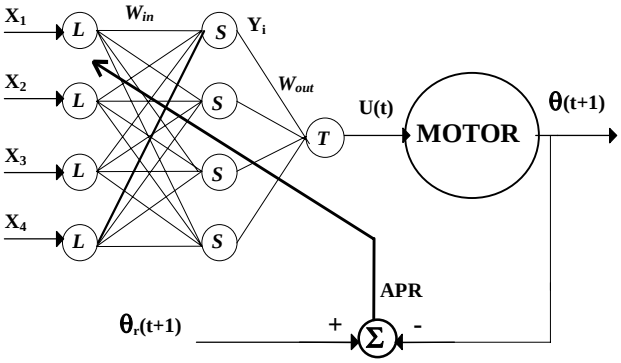

Fig.3 - Controlador neural direto de posição

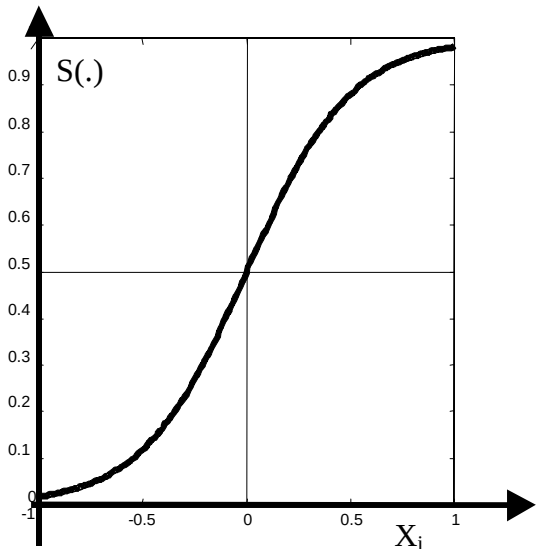

Fig.4 - A sigmóide

Define-se o erro no posicionamento do eixo do motor como $E(t+1) = \theta_r(t+1) - \theta(t+1)$. O índice de desempenho usado para adaptação da variável de controle U(t) da RNMC do SCIE é mostrado na Eq.4.

$$I(U(t)) = \tfrac{1}{2}(E(t+1))^2 = \tfrac{1}{2}[\theta_r(t+1)-\theta(t+1)]^2 \quad (4)$$

O treinamento da RNMC é feito com o Algoritmo de Propagação Retroativa – APR (Humelhart, Hilton e Willams, 1986) e o valor U(t+1) é encontrado usando a regra delta (Eq.5) e o índice de desempenho da Eq.4. Define-se μ como o fator de treinamento da RNMC.

$$U(t+1) = U(t)+(-\mu\nabla(I(U))=U(t)-\mu\partial(I(U(t))/\partial U \quad (5)$$

Desenvolvendo a derivada do índice de desempenho como mostrado na Eq.6, obtém-se a derivada parcial do índice de desempenho em relação à entrada como uma função do jacobiano do motor que é representado como $\partial\theta(t+1)/\partial U(t)$.

$$\partial(I(U(t))/\partial U(t) = -E(t+1)\,\partial\theta(t+1)/\partial U(t) \quad (6)$$

A Eq.7 é usada para determinar U(t) se for conhecido o jacobiano do motor.

$$U(t+1) = U(t)+\mu\,E(t+1)\partial\theta(t+1)/\partial U(t) \quad (7)$$

4 REGRAS LINGUÍSTICAS DE CONTROLE

Na Fig.5a é mostrado um círculo, com quatro quadrantes, representando o torque de carga Tl(t) gerado pelo pêndulo para diferentes ângulos θ(t). O torque $T_m(t)$ gerado pelo motor para estabilizar o pêndulo deve ter a mesma intensidade de $T_l(t)$ mas com sinal contrário.

No pêndulo invertido, para cada quadrante do círculo de torque da Fig.5a, existem diferentes valores de U(t) capazes de estabilizar o pêndulo no quadrante. Cavalcanti e Ferneda (1995) desenvolveram as regras linguísticas mostradas abaixo para o treinamento seguido de controle do SCIE. Nestas regras, θm representa a função de pertinência dos quadrantes de θ(t) (ver Fig.5b), ΔU(t) representa o incremento de U(t) necessário para sustentar o braço do pêndulo nos quadrantes Q_i. Além disso, definiu-se a variável nebulosa θ_p (ver fig.5c) que assume valores linguísticos θ_p = PG (Positivo Grande) quando, em P.U., θ(t) > 1 (ou θ(t) > π), e θ_p = NG (Negativo Grande) quando, em P.U., θ(t) < -1 (ou θ(t) < -π).

1) **if** $\theta_m == Q_1$ **or** $\theta_m == Q_2$ **then** U(t) < 0;
2) **if** $\theta_m == Q_3$ **or** $\theta_m == Q_4$ **then** U(t) > 0;
3) **if** $\theta_m == Q_1$ **or** $\theta_m == Q_2$ **then** ΔU(t) < 0;
4) **if** $\theta_m == Q_3$ **or** $\theta_m == Q_4$ **then** ΔU(t) > 0;
6) **if** θ_p == PG **then** θ(t) = θ(t)-2π;
7) **if** θ_p == NG **then** θ(t) = θ(t)+2π;

5 O ALGORITMO DO CONTROLADOR NEURAL

Cavalcanti, Lima e Deep (1994) desenvolveram o conceito de estado passivo para permitir o treinamento em tempo real da RNMC. Basicamente, eles supuseram que o sistema representado pelo conjunto motor mais controlador neural (sistema global) tem pelo menos um estado de equilíbrio, e sem perda de generalidade, a origem pode ser considerada um desses estados. Representando-se o sistema motor na forma discreta por x = f(x,u), eles definiram:

Definição 1 - Um ponto $x_e \in \Re^n$ é um ponto de equilíbrio de x = f(x,u) se existir uma entrada u_e tal que: $x_e = f(x_e,u_e)$.

Assim, particularmente, a RNMC da Fig.3 pode ser treinada, usando-se a tensão de excitação da armadura u e o estado x_e, para fornecer na sua saída o valor u_e correspondente ao estado

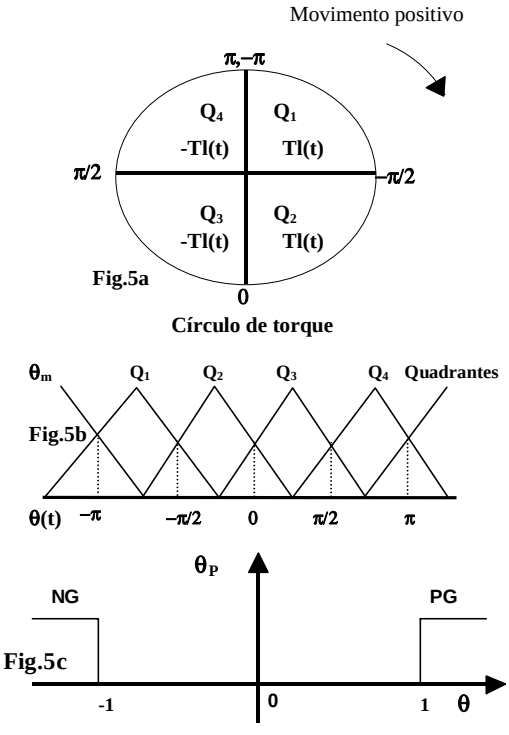

Fig.5 - Posicionamento do pêndulo.

de equilíbrio. Baseado nessa estratégia, Cavalcanti, Lima e Deep (1994) definiram o Estado Passivo do sistema global como:

Definição 2 - Um sistema dinâmico está no estado passivo quando estiver no ponto de equilíbrio com $x_e = f(x_e,u_e)$ e o controlador neural garantir a excitação de controle u_e.

Além disso, Cavalcanti, Lima e Deep (1994), usando a regra delta generalizada e a Eq.2, provaram que para o sistema motor cc, se o ângulo de referência ($\theta_r(t)$) for aproximadamente igual ao ângulo atual do eixo do motor ($\theta(t)$), as Eq.8 e Eq.9 podem ser usadas para treinamento em tempo real da RNMC. Eles definiram χ como o fator de adaptação do controlador neural.

$$U(t+1) = U(t)+\Delta U(t) \quad (8)$$
$$\Delta U(t) = \chi.E(t+1) \quad (9)$$

No Algoritmo de Controle mostrado abaixo, a ETAPA 1 é responsável pela inicialização dos parâmetros e pesos da RNMC. Na ETAPA 2 é feito o treinamento da rede neural através do Algoritmo de Propagação Retroativa de Erro (APR) e o controle em tempo real do sistema motor cc. Em ambas é necessário que $\theta_r(t) \cong \theta(t)$.

Algoritmo de Controle

ETAPA 1

1) Geram-se aleatoriamente os pesos W_{in} e W_{out} entre os limites 0.25 e -0.25.
2) Faz-se: $\Theta = \beta = 1$ e $\delta = 0$ (todos os neurônios).
3) Atribui-se: $\mu = 0.1$ (fator de treinamento) e $\chi = 0.5$ (fator de adaptação do controlador).

ETAPA 2

4) Obtém-se $\theta r(t)$.
5) Utiliza-se $\theta_r(t)$ capaz de calcular o novo U(t) do ponto passivo da RNMC (Algoritmo híbrido).
6) Calcula: $\Delta U(t) = \chi.E(t+1)$ e $U(t+1) = U(t)+\Delta U(t)$
7) Treina a RNMC com o APR.
8) Repetem-se as etapas 5, 6 e 7 até que: $|\theta(t)-\theta_r(t)| < \varepsilon$, para um ε suficientemente pequeno.

Na Fig.6a são mostrados resultados experimentais obtidos para o movimento do pêndulo da sua posição de repouso ($\theta = 0$ rad) até uma posição referência de 30^o ($\theta = \pi/6$ rad). Inicialmente, a RNMC da Fig.3 foi treinada com o estado passivo zero ($X_1 = X_2 = X_3 = X_4 = 0.$). Na Fig.6a pode-se observar que o ângulo referência $\theta r(t)$ varia de 30^o a -30^o. A velocidade do pêndulo é representada por $\Omega(t)$ em P.U. com valor de base π rad/s. Na Fig.6a, o crescimento linear de U(t) é explicado pelo ajuste $\Delta U(t) = \chi.E(t+1)$ na saída do controlador neural adaptativo. Quando o braço do pêndulo se aproxima da posição de referência, o valor de U(t) fica aproximadamente constante. Na Fig.6b são representadas as posições ocupadas pelo pêndulo durante o seu movimento de $\theta(t) = 0$ (posição a), passando por $\theta(t) = \pi/6$ (posição b) até atingir $\theta(t) = \pi/6$ rad (posição c).

O valor $E(t) = |\theta_r(t)-\theta(t)|$ é suficientemente pequeno para garantir o estado passivo da RNMC. O movimento cíclico do braço do pêndulo, da posição $-\theta_r(t)$ a $\theta_r(t)$ (e vice-versa), é denominado de oscilação do pêndulo. O número de oscilações é NO e a sua variável nebulosa associada é representada por NO_f (ver Fig.7), NO_f assume os seguintes valores linguísticos: Z (próximo de zero), M (médio) e G (grande).

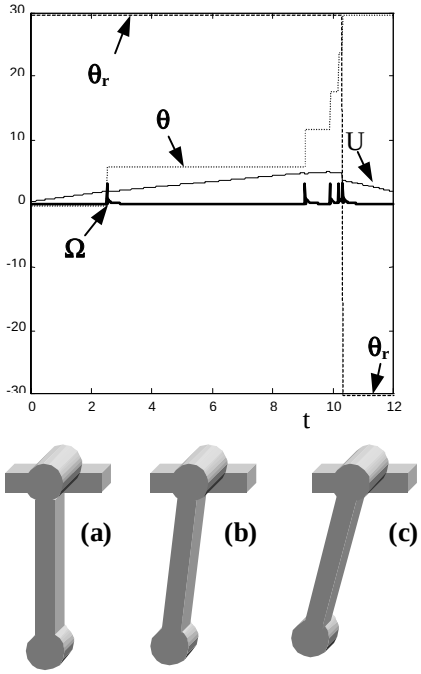

Fig. 6 - Treinamento da RNMC: (a) Resultados Experimentais; (b) Posições do pêndulo

6 O CÍRCULO DAS ESTRATÉGIAS

Cavalcanti e Ferneda (1995) sugeriram a utilização de um conjunto de duas estratégias para representar os tipos de controle do pêndulo pelo SCI. Neste trabalho, devido à frequente falha no posicionamento do pêndulo invertido, desenvolveu-se para o SCIE o círculo das estratégias mostradas no diagrama de estados da Fig.8. Nesse círculo são mostradas três estratégias: TREIN; OSCIL, e ATRAC. Essas estratégias são ativadas de acordo com o estado do sistema pêndulo invertido (posição e velocidade do pêndulo). As variáveis binárias que representam as transições e as estratégias são INICIO, OSCIL e ATRAC.

Inicialmente INICIO=1 e a regra linguística número 8 é executada.

8) **if** INICIO **then** TREIN = 1

A estratégia TREIN é representada pelas regras 9 e 10 mostradas abaixo. ϕ é o valor do ângulo referência. Os símbolos == e $\cong$ são utilizados para representar a comparação entre os valores das variáveis, respectivamente a igualdade e a igualdade aproximada.

9) **if** TREIN **and** $\theta(t) == \phi$
then $\theta_r(t) = -\phi$ **and** NO = NO+1

10) **if** TREIN **and** $\theta(t) == -\phi$
then $\theta_r(t) = \phi$ **and** NO = NO+1

A variável NO (número de oscilações), mostrada na Fig.7, é utilizada para indicar o fim da fase de treinamento. Após esta fase, a estratégia OSCIL é acionada (regra 11).

11) **if** TREIN **and** $NO_f == M$
then TREIN = 0 **and** OSCIL = 1

A regra linguística 12 é usada na estratégia OSCIL. ϖ representa o incremento do ângulo referência.

12) **if** OSCIL **and** $\theta(t) == \theta_r(t)$
then $\theta_r(t) == -\theta_r(t)-\varpi*SGN(\theta_r(t))$ **and** NO = NO+1

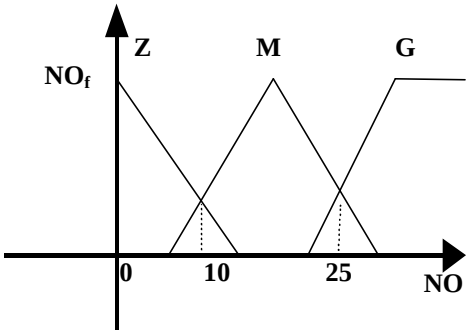

Fig. 7 Funções de pertinência de NO.

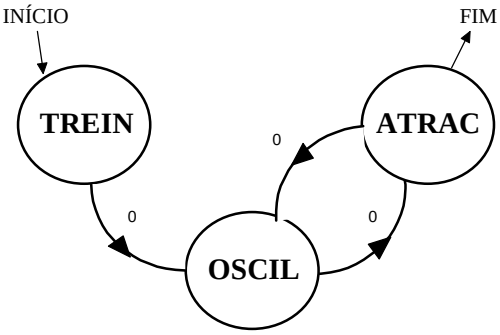

Fig.8 O círculo das estratégias

A estratégia de atração (ATRAC) é acionada sob as condições mostradas na regra linguística 13. Isto é, quando o braço do pêndulo se aproximar do objetivo.

13) **if** OSCIL **and** $|\theta(t)| \cong \pi$
then ATRAC = 1 **and** OSCIL = 0 **and** $|\theta_r(t)| = \pi$
and NO = 0.

Durante a estratégia ATRAC, a variável NO é incrementada quando o pêndulo atravessa, com velocidade não nula, o ponto $\theta(t)=\theta_r(t)$ (ver regra 14).

14) **if** ATRAC **and** $\theta(t) == \theta_r(t)$ **and** $(\theta(t)-\theta(t-1)) \neq 0$
then NO = NO+1

A estratégia OSCIL é reativada (regras 15 e 16) quando o pêndulo ultrapassa um número máximo de oscilações permitidas (50 oscilações) ou o pêndulo se desloca para um dos quadrantes inferiores (Q_2 ou Q_3).

15) **if** ATRAC **and** $NO_f == G$
then OSCIL = 1 **and** ATRAC = 0 **and** $\theta_r(t) = 0$ **and**
NO = 0

16) **if** ATRAC **and** $(\theta_m == Q_2$ **or** $\theta_m == Q_3)$
then OSCIL = 1 **and** ATRAC = 0 **and** $\theta_r(t) = 0$

A estratégia OSCIL se encarrega de posicionar o pêndulo em $\theta(t) = 0$.

As regras linguísticas 6 e 7 são utilizadas para normalizar o ângulo do pêndulo entre $-\pi$ e π.

7 O SISTEMA DE CONTROLE INTELIGENTE EVOLUCIONÁRIO

A Fig.9 mostra o esquema geral do SCIE. Neste esquema o bloco PLANTA representa o sistema motor CC e o pêndulo invertido. A RNMC do controlador Neural de Posição (bloco Controle Neural) é treinada com o Algoritmo de Propagação Retroativa. Os valores das posições da planta , das posições de referência e do fator de treinamento da rede neural artificial são transformados para a forma nebulosa (bloco Nebulização) para serem usadas pelo núcleo do Sistema Inteligente para que, juntamente com o Algoritmo Genético, gerarem os fatores de adaptação (χ_i) da RNMC e as posições de referência ($\theta_r(t)$) inerentes a cada uma das estratégias.

Algoritmos Genéticos são utilizados para a escolha dos fatores de adaptação χ_T; χ_O; χ_A do controlador neural (valores χ da Eq.9) associados às estratégias TREIN, OSCIL e ATRAC, respectivamente. O número de oscilações é utilizado pelo AG como função "fitness" (ver seção 8) para cálculo desses fatores de adaptação do controlador neural. Para o fator de adaptação χ foi associado a variável nebulosa χm, cujos valores usados na fase de "desnebulização" são mostrados na Tabela 1. A variável nebulosa χ_m é caracterizada pelos termos lingüísticos: Z (próximo de zero), Pq (pequeno), M (médio) e G (grande).

Para o peso acoplado ao pêndulo foi associado a variável nebulosa P_m (ver Fig.10), que é caracterizada pelos termos lingüísticos: Z (próximo de zero), Pq (pequeno), M (médio) e G (grande). A Tabela 2 mostra os valores de P usados na fase de "desnebulização".

Na Fig.11 são apresentados os resultados experimentais obtidos durante o posicionamento do pêndulo invertido usando o SCIE. A abscissa representa o tempo em segundos e a ordenada representa os ângulos referência (θ_r) e atual (θ) do pêndulo. Esses resultados foram obtidos com: 1) Fator de treinamento da RNMC μ = 0.3; 2) Número de oscilações da estratégia de treinamento NO = 10; 3) Peso P_m = Pq no eixo do motor cc.

Pode-se observar na Fig.11 que, inicialmente, a estratégia TREIN é acionada (regra 8). Na estratégia TREIN, usando o Algoritmo de Controle e as regras 9 e 10, a RNMC é treinada com valores de referência $\theta_r(t) = \pm 18^o$ e $\chi_T = 0.9$ (χ_m = G) (intervalo $0 \leq t \leq 23$s da Fig.11). A seguir, a estratégia OSCIL é acionada (regra 11). A estratégia OSCIL utiliza a regra 12 e $\chi_O = 0.3$ (χm = Pq) (intervalo $23s \leq t \leq 57s$ da Fig.11). Aproximadamente em t = 57s, o pêndulo estará próximo do objetivo e, usando a regra 13, a estratégia ATRAC é acionada. A estratégia ATRAC usa $\chi_A = 0.3$ (χ_A = Pq). A estratégia ATRAC consegue posicionar o pêndulo na posição invertida (em t = 70s).

A regra 14 incrementa o número de oscilações ocorridas durante a fase ATRAC. Observou-se que nem sempre se conseguia com sucesso o posicionamento do pêndulo na sua posição invertida usando a estratégia ATRAC. Nestes casos, o pêndulo permanecia oscilando em torno da posição de referência. Quando o número de oscilações ultrapassa um valor especificado, as regras 15 e 16 são usadas para reativar a estratégia OSCIL e levar o sistema ao ponto passivo, reiniciando todo o processo novamente. Para indicar a falha no posicionamento do pêndulo, adotou-se o valor 50 para o número de oscilações máximo admissível. (NO = 50).

Na Tabela 3 são mostrados os valores dos fatores de adaptação do controlador neural, junto com o número de oscilações (NO = 34) necessárias para se conseguir o posicionamento do pêndulo invertido para o experimento ilustrado na Fig.11.

8 ADAPTAÇÃO POR ALGORITMO GENÉTICO

Nesta Seção descreve-se a implementação do Algoritmo Genético do SCIE. Observou-se que nem sempre se conseguia com sucesso o posicionamento do pêndulo na sua posição invertida usando os valores da Tabela 3, para diferentes cargas. Por este motivo, procurou-se adaptar os ganhos $\{\chi_T, \chi_O, \chi_A\}$,

de maneira a otimizar o desempenho do sistema. Para isto, escolheu-se um método de otimização evolucionário: o Algoritmo Genético. Algoritmos genéticos, como originalmente formulado por Holland (1992), foi inicialmente usado para modelar a evolução biológica. Posteriormente, demonstrou-se que o mesmo poderia ser utilizado para resolver problemas de otimização e que resultados de otimização global poderiam ser alcançados com o mesmo. Numerosas versões modificadas do algoritmo original foram propostas, melhorando seu desempenho. Na sua forma padrão, os Algoritmos Genéticos são métodos estocásticos de otimização para problemas da forma mostrada na Eq.10.

Maximizar f(s), sujeito a : $s \in \Omega = \{0, 1\}^n$ (10)

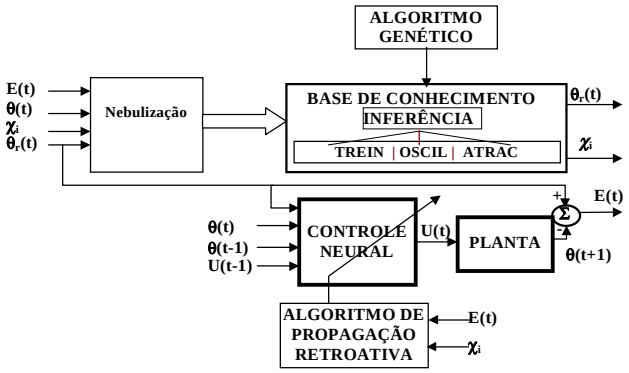

Fig.9 O SCIE

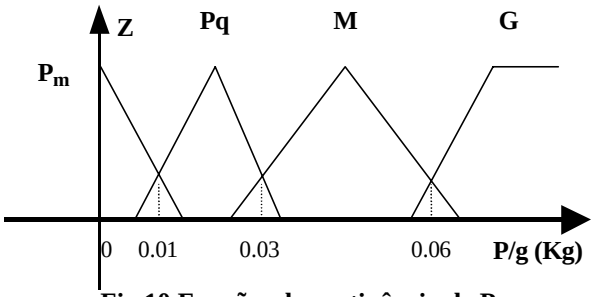

Fig.10 Funções de pertinência de P_m

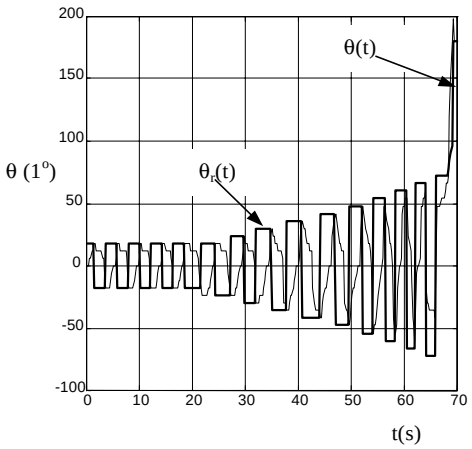

Fig.11 Resultados experimentais, P_m = Pq

Tabela 1

χ_m	Z	Pq	M	G
χ	0.01	0.3	0.6	0.9

Tabela 2

P_m	Z	Pq	M	G
P (Kg)	0.01	0.02	0.04	0.08

Tabela 3

χT	χ_O	χ_A	NO
G	Pq	Pq	34

onde, f : $\Omega \rightarrow$ R é a função "fitness" e os vetores s, binários n–dimensionais em Ω, são chamados "strings" ou cromossomos. Os vetores s, que representam diferentes soluções do problema, funcionam de modo a emular e modelar cromossomos biológicos e a função f(.) é uma medida da "aptidão" dos mesmos para solucionar o problema em questão. A característica mais marcante dos Algoritmos Genéticos é que os mesmos mantém uma coleção de amostras do espaço de busca Ω, em lugar de um simples ponto do mesmo. Esta coleção s(i) = $\{s_1, ..., s_m\}$ é chamada de "População" de cromossomos, a qual é modificada a cada "Geração" i (iteração do Algoritmo Genético) através de operadores genéticos: Seleção, Cruzamento e Mutação. A população inicial é gerada aleatoriamente. Populações grandes não implicam necessariamente melhores resultados. A Seleção é implementada através do método da Roleta. Neste método, os m cromossomos da nova geração são selecionados a partir da geração anterior, de acordo com uma probabilidade p_s, proporcional à função fitness relativa de cada cromossomo (Eq.11).

$$p_S(s_i) = \frac{f(s_i)}{\sum_{j=1}^{m} f(s_j)} \qquad (11)$$

Os strings sobreviventes ao processo de seleção são submetidos a cruzamento e mutação.

Na Fig.12 mostra-se o Algoritmo Genético utilizado para encontrar os melhores valores dos fatores de adaptação do controlador neural do SCIE. Este algoritmo, proposto para solucionar o problema de posicionamento do pêndulo invertido, difere em alguns aspectos do Algoritmo Genético padrão. A sua implementação e resultados experimentais serão descritos a seguir.

Início. Usou-se a RNMC já treinada anteriormente no posicionamento do pêndulo com peso P_m = Pq (mostrado na seção 7). A seguir, os valores de χ_T, χ_O, χ_A, foram codificados em 2 dígitos binários, de tal maneira que os bits 00, 01, 10 e 11 representassem os valores nebulosos Z, Pq, M e G, respectivamente. Escolheu-se a minimização do número de oscilações do pêndulo, necessário para alcançar o posicionamento invertido do pêndulo, como função objetivo ("fitness").

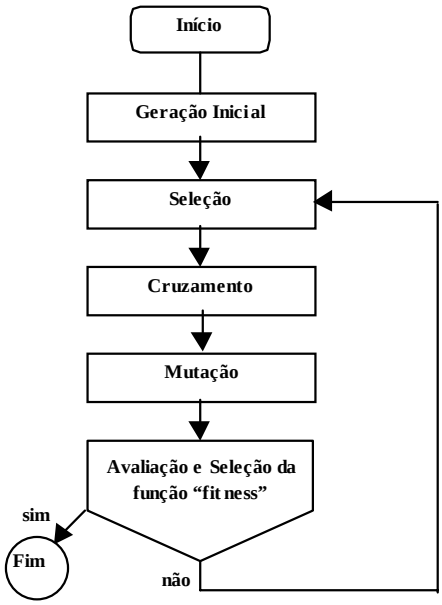

Fig.12 Algoritmo Genético

Geração Inicial. Criou-se uma tabela (Tabela 4) para armazenar a população dos cromossomos e o número de oscilações do pêndulo. Na primeira coluna da tabela foi armazenado o índice da linha. Suas colunas centrais (colunas 2, 3 e 4) foram utilizadas para armazenar os valores linguísticos dos fatores de adaptação $\{\chi_T; \chi_O; \chi_A\}$ do controlador neural. Os dez conjuntos de três pares de bits (cromossomos) que codificam os valores linguísticos dos fatores de adaptação da Geração Inicial, foram gerados aleatoriamente. Para cada conjunto de valores $\{\chi_T; \chi_O; \chi_A\}$ foi realizada experimentalmente a tentativa de posicionar o pêndulo. Foi estabelecido um número de oscilações máximo permissível NO = 50. Quando NO ultrapassou este limite, considerou-se que o posicionamento fracassou e a regra 16 foi executada, de modo a fazer nova tentativa. A última coluna mostra os números de oscilações NO obtidas experimentalmente para cada tentativa de posicionamento.

Seleção. A seleção se caracteriza pela ordenação da Tabela 4. O resultado desta ordenação é mostrado na Tabela 5.

Neste processo, diferente do método da roleta do Algoritmo Genético padrão, a importância de um cromossomo na solução do problema é expressa pela sua posição relativa na tabela. Os cromossomos menos eficientes são mantidos no fundo da tabela, visando gerar uma maior variedade de cromossomos "filhos" através de cruzamento e reprodução. Desta maneira, os cromossomos "superiores" representam as melhores soluções atuais encontradas dentro de uma região localizada do espaço de busca. Os cromossomos "inferiores" permitem gerar soluções alternativas em outras regiões do espaço de busca, as quais podem, eventualmente, levar a soluções melhores, caso os cromossomos "superiores" se encontrem numa região de mínimo local.

Cruzamento. A seguir, foi realizado o processo cruzamento entre pares de cromossomos escolhidos aleatoriamente na população dos cromossomos (Tabela 5). Um lugar de cruzamento (no cromossomo) é escolhido através de um número inteiro randômico k para cada par selecionado. Os dois bits imediatamente após o bit k dos cromossomos de um certo par são intercambiados de acordo com uma probabilidade de cruzamento $p_c = 0.6$.

Na Tabela 6 mostra-se este processo para os cromossomos 1 e 7, onde os cromossomos "filhos", resultantes do processo de cruzamento, são denotados como *1 e *7. Diferentemente do Algoritmo Genético padrão, utilizou-se cruzamento de dois pontos.

Mutação. Diferente dos Algoritmo Genético padrão, o operador de mutação foi aplicado a cromossomos selecionados randomicamente (com uma probabilidade de 80% de sofrerem mutação). Apenas um bit, escolhido aleatoriamente dentro de cada cromossomo selecionado, foi submetido a mutação. Como a população de cromossomos é formada por 10 indivíduos e cada cromossomo compõe-se de 6 bits, cada bit da população tem uma probabilidade de 0.8*0.1/6 de sofrer mutação, ou $p_m = 0.0133$. A Tabela 7 mostra um exemplo de mutação para o cromossomo 3 da Tabela 5. A mutação é feita no bit em negrito no campo sombreado. Os valores da nova linha (marcada como *3) representam o cromossomo resultante.

Os cromossomos "filhos" resultantes dos processos de cruzamento e mutação foram utilizados para posicionar o pêndulo e a função fitness (número de oscilações NO) foi computada para cada um deles (ver 5ª coluna das tabelas 6 e 7). Esses cromossomos, junto com os "pais", competiram entre si para permanecer na população. A Tabela 8 mostra a população dos cromossomos da 2ª iteração do Algoritmo Genético. As linhas 2 a 9 da tabela original (Tabela 5) são deslocadas para as linhas 3 a 10. O resultado do cruzamento marcado (*1), após a competição, é escolhido para a próxima geração e colocado na linha 2.

Tabela.4

i	χ_T	χ_O	χ_A	NO
1	11	10	10	35
2	11	11	01	36
3	10	11	01	33
4	10	11	11	28
5	10	10	10	40
6	11	10	01	35
7	10	10	01	35
8	10	00	11	50
9	10	00	10	50
10	11	10	11	50

Tabela 5

i	χ_T	χ_O	χ_A	NO
1	10	11	11	28
2	10	11	01	33
3	11	10	10	35
4	11	10	01	35
5	10	10	01	35
6	11	11	01	36
7	10	10	10	40
8	10	00	11	50
9	10	00	10	50
10	11	10	11	50

Tabela 6

i	χ_T	χ_O	χ_A	NO
1	10	**11**	**11**	28
7	10	10	**10**	40
*1	10	**11**	**10**	***32***
*7	10	10	**11**	***50***

Tabela 7

i	χ_T	χ_O	χ_A	NO
3	11	10	**1**0	35
*3	11	10	**0**0	***50***

Tabela 8

i	χ_T	χ_O	χ_A	NO
1	10	11	11	28
2	10	11	10	32
3	10	11	01	33
4	11	10	10	35
5	11	10	01	35
6	10	10	01	35
7	11	11	01	36
8	10	10	10	40
9	10	00	11	50
10	10	00	10	50

9 RESULTADOS EXPERIMENTAIS DO SCIE

Aplicou-se o SCIE proposto para o posicionamento do pêndulo invertido com cargas variáveis (P_m = Z, Pq, M ou G). Aplicando o Algoritmo Genético repetidas vezes, verificou-se que, na maioria delas, a função fitness NO convergiu em poucas iterações (menos de dez gerações) para valores aceitáveis (25 < NO < 30), conseguindo sucesso no posicionamento do pêndulo na posição invertida. Por exemplo, utilizando-se um peso P_m = G, tentou-se posicionar o pêndulo invertido como mostrado na Fig.13, onde se mostra os resultados experimentais obtidos para duas primeiras gerações do Algoritmo Genético. Na primeira geração, utilizando os valores dos fatores de adaptação da primeira linha da tabela de cromossomos, o SCIE tenta posicionar o pêndulo até atingir cinqüenta oscilações, momento em que se assume fracasso no posicionamento. Na segunda tentativa, utilizando o cromossomo da segunda geração, observa-se na figura que o SCIE conseguiu posicionar o pêndulo na posição invertida.

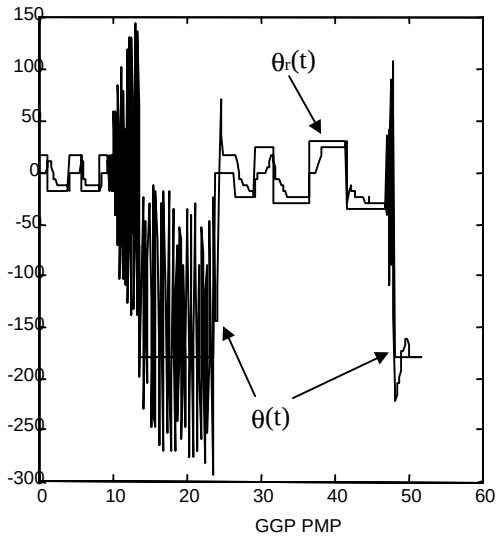

Fig. 13 - Resultados experimentais.

10 CONCLUSÃO

Foi projetado e implementado experimentalmente um Sistema de Controle Inteligente Evolucionário, baseado em Controlador Neural, Lógica Nebulosa e Algoritmos Genético, para posicionamento automático de um pêndulo invertido independente da sua carga. A linguagem de implementação utilizada foi C++ 4.5 da Borland. A escolha dos fatores de adaptação do controlador neural foi realizada por meio do Algoritmo Genético, e verificou-se experimentalmente que o SCIE é capaz de garantir o posicionamento do pêndulo invertido na maioria das vezes, mesmo falhando em algumas tentativas. Foram apresentados exemplos dos operadores para mutações e para cruzamentos nos cromossomos do SCIE. Atualmente, estuda-se a aplicação da arquitetura de SCIE proposta para o controle de um sistema de robôs cooperantes (dois robôs de dois elos movimentando uma única carga).

REFERÊNCIAS BIBLIOGRÁFICAS

Äström, K. J.; Wittenmark, B. (1989). *Adaptive Control*, Addison-Wesley, Reading, USA.

Cavalcanti, J.H.F. (1995). Adaptation and Learning Factor for Neural Controllers, *Proceedings of the IECON'95*, Orlando, USA, pp.1406-1410.

Cavalcanti, J.H.F. (1996). Controle Inteligente de um motor c.c. usando algoritmos genéticos, *Anais do INDUSCON'96*, São Paulo, SP.

Cavalcanti, J.H.F., Sales Jr, E.F. (1996). Movimento do Robô-Trapezista Usando Computação Evolucionária, *Anais do XI Congresso Brasileiro de Automática - CBA'96*, São Paulo, SP, pp.567-572.

Cavalcanti, J.H.F; Ferneda, E. (1995). Intelligent Control of an Inverted Pendulum - Training and Evolution, *Anais do SEMISH'95*, Canela, RS, pp.627-637.

Cavalcanti, J.H.F; Lima, A.M.N.; Deep, G.S. (1994). On-line Training of Adaptative Neural Network controllers, *Proceedings of the IEEE Industrial Electronics Society IECON'94*, Bologna, Italy.

Fogel, D.B.; Fogel, L.J. (1994). Evolutionary Computation, *Transactions on Neural Networks*, Vol.5, nº1, Special Issue on Evolutionary Computation, pp.1-147.

Holland, J.H. (1992). *Adaptation in Natural and Artificial Systems*, The MIT Press.

Humelhart D.E.; Hinton, G.E.; Williams R.J. (1986). Learning Internal Representations by Error Propagation, In: Humelhart, D.E.; McClelland, J.L. (Eds.) *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, Vol. 1, pp.318-362, The MIT Press.

Narenda, K.S.; Parthasarathy, K. (1990). Identification and Control of Dynamical Systems Using Neural Networks, *IEEE Transactions On Neural Networks*, Vol.1 Nº 1, pp.4-27.

Shoureshi, R. (1991). The Mystique of Intelligent Control, *IEEE Control Systems Magazine*, nº 1, pp.33-33.

Zadeh L.A. (1988). Fuzzy Logic, *IEEE Computer Mag.* April 1988, pp83-93. Publicado também In: Anderson, J.A.; Rosenfeld, E. (Ed.) *Neurocomputing Foundations of Researchs*, pp.177-194, The MIT Press, Cambridge, USA.