

6. Controle por Aprendizado Neuro-Fuzzy

6.1. Introdução

Neste capítulo é apresentado o controle por aprendizado utilizando um sistema híbrido Neuro-Fuzzy, para o cálculo e atualização dos pontos de reversão da servo-válvula do sistema servo-hidráulico. Além disso, são apresentadas as comparações entre os modelos de controle por aprendizado desenvolvido. Todos os programas de simulação foram desenvolvidos no software MATLAB®.

6.2. Esquema do controle por aprendizado NF

Neste modelo de controle por aprendizado, a informação desconhecida é estimada ou aprendida pelo sistema Neuro-Fuzzy e fornecida ao controlador “bang-bang”. Assim, à medida que o sistema Neuro-Fuzzy armazena mais informação, o controlador é atualizado melhorando o desempenho do sistema.

Na Figura 6.1 apresenta-se o diagrama de blocos que ilustra o controle por aprendizado Neuro-Fuzzy. Neste modelo de controle, a informação é representada pela mesma variável U_{II} definida anteriormente, e que nesse caso é a saída do sistema Neuro-Fuzzy. A informação que gera o valor de U_{II} é armazenada nos pesos da estrutura do sistema Neuro-Fuzzy. A variável U_{II} , utilizada para mudar a ação de controle sobre a servo-válvula, é atualizada após cada ciclo de operação através do ajuste dos pesos da estrutura Neuro-Fuzzy, utilizando um algoritmo de aprendizado baseado nos erros medidos.

O objetivo do sistema Neuro-Fuzzy é fornecer o valor de U_{II} ao controlador, determinando assim o ponto de reversão no qual a servo-válvula vai reverter seu sentido, de maneira que a máquina (o sistema servo-hidráulico) seja mantida trabalhando em seus limites de operação como nos controladores descritos no capítulo anterior.

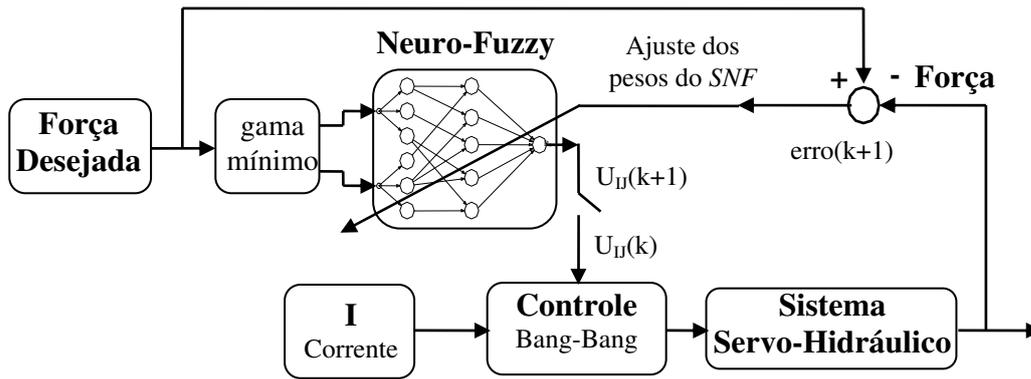


Figura 6.1. Diagrama de blocos do controle por aprendizado Neuro-Fuzzy.

Na Figura 6.1, o valor de $U_{II}(k)$ é atualizado pela chave com o valor de $U_{II}(k+1)$, e o erro normalizado $erro(k+1)$ é obtido em função da força desejada e da força real medida.

As entradas do sistema Neuro-Fuzzy são *gama* (dobro da amplitude) e *mínimo* da grandeza controlada, a saída do sistema é a variável U_{II} . O sistema Neuro-Fuzzy é constituído por duas camadas escondidas: a camada Fuzzy e a camada de regras. Os pesos do sistema Neuro-Fuzzy ω_j entre a camada de regras e a camada de saída são atualizados utilizando o algoritmo de aprendizado *Backpropagation*, baseado no $erro(k+1)$ a cada iteração. Na Figura 6.2 apresenta-se a estrutura do sistema Neuro-Fuzzy proposto.

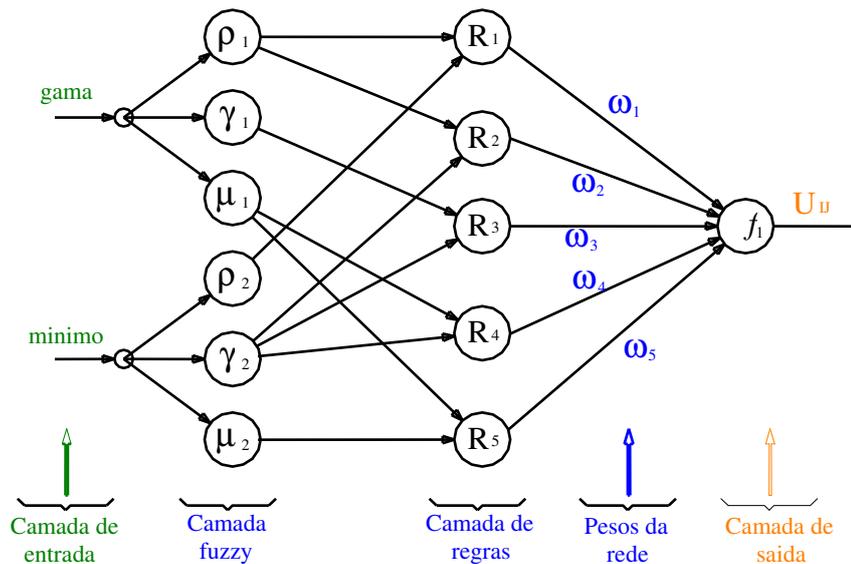


Figura 6.2. Estrutura do Sistema Neuro-Fuzzy.

6.3. Modelagem do Controle por Aprendizado Neuro-Fuzzy

O controle por aprendizado baseado em sistemas híbridos Neuro-Fuzzy (*SNF*) geralmente é constituído pela combinação de sistemas Fuzzy e Redes Neurais (*ANN*). Ele combina as vantagens das *ANN* tais como a habilidade de aprendizagem, otimização, e conexão em estrutura, com as vantagens dos sistemas Fuzzy, que usa raciocínio semelhante ao humano, com facilidade de incorporar informações de especialistas.

A modelagem deste sistema é determinada pelas configurações das características de seus “genitores”, a parte Fuzzy e Neural.

6.3.1. Modelagem Fuzzy do *SNF*

A modelagem das características da parte Fuzzy é determinada pela configuração dos parâmetros das seguintes 4 categorias:

- **Modelo Fuzzy:** O modelo de inferência Fuzzy (formato das regras) implementado neste *SNF* foi o modelo Takagi-Sugeno, com conjuntos Fuzzy de saída do tipo “*singleton*” para cada uma das regras. As regras fuzzy são da seguinte forma:

$$\text{Regra } j: \text{ Se "gama" é } \rho_1 \text{ e "mínimo" é } \mu_2 \text{ Então } U_{jj} = \omega_j \quad (6.1)$$

- **Formato das Funções de Pertinência:** As funções de pertinência na camada Fuzzy do *SNF* são geralmente funções simétricas, tais como funções triangulares, trapezoidais, e sigmóidais. Nesta dissertação, escolhem-se 8 funções de pertinência do tipo triangular para cada uma das variáveis de entrada, pela simplicidade para sua implementação experimental. Na Figura 6.3, ilustra-se a função de pertinência triangular, cuja equação é expressa por:

$$\mu(x_i) = 1 - \frac{2|x_i - c_i|}{b_i} \quad (6.2)$$

onde, x_i é o valor da variável de entrada, c_i é o centro do triângulo da função de pertinência, e b_i é a largura da base do triângulo.

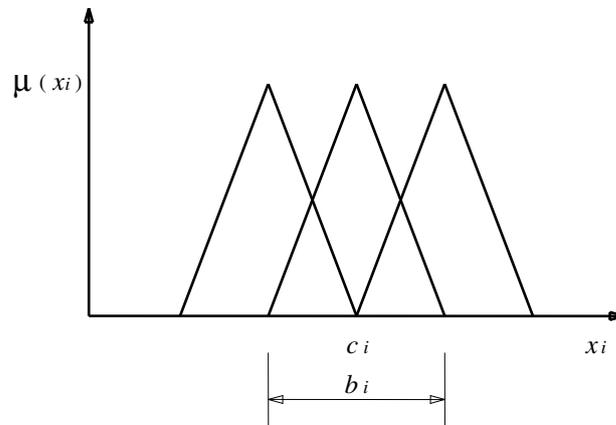


Figura 6.3. Função de pertinência triangular do SNF.

- **Particionamento do espaço de E/S:** O particionamento do espaço das variáveis de entrada e saída é do tipo fuzzy grid, o qual mapeia internamente regiões fuzzy relacionados através de suas regras. Na Figura 6.4 apresenta-se o particionamento fuzzy grid.

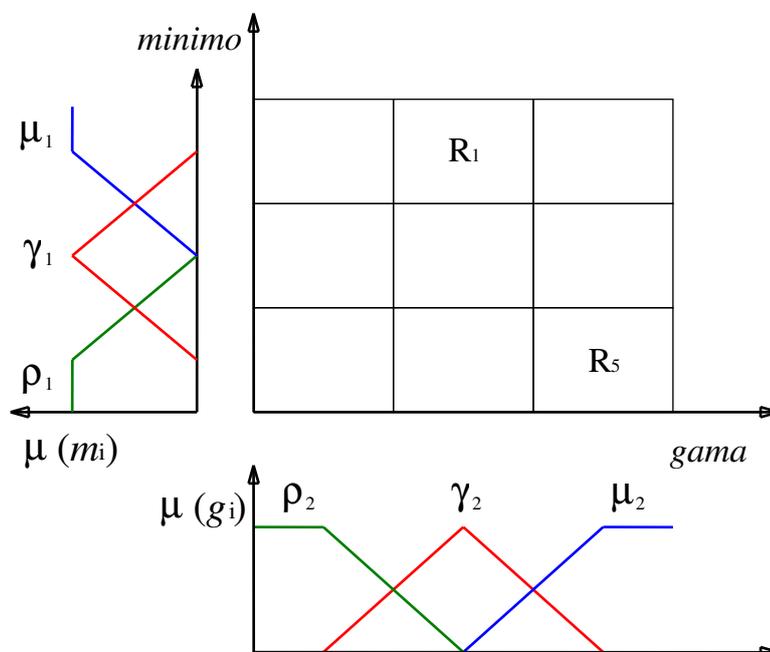


Figura 6.4. Particionamento Fuzzy Grid, onde ρ_1, γ_1, μ_1 e ρ_2, γ_2, μ_2 , são os graus de pertinência dos conjuntos Fuzzy das variáveis “*minimo*” e “*gama*” respectivamente.

- **Método de defuzzificação:** O conseqüente das regras é uma função do tipo “*singlenton*”, e a saída do sistema, U_{II} , é obtida da média ponderada dos graus de disparo de cada regra. A saída U_{II} é calculada por:

$$U_{II} = f(P(\mu_i, \mu_i), \omega) \quad (6.3)$$

onde $P(\mu_i, \mu_i)$ é o nível de disparo correspondente a cada regra, e ω é o peso da estrutura (saída “*singlenton*”).

6.3.2. Modelagem da parte neural

A modelagem das características da parte neural está relacionada com a capacidade de aprendizado do *SNF*, e determinada pela configuração dos parâmetros das seguintes sub-classes:

- **Tipo de aprendizado:** O aprendizado utilizado é do tipo *online*, isto devido a que ao longo da operação da máquina o sistema tem que ter a capacidade de mudar os pontos de reversão, de maneira que os pesos do *SNF* têm que ser atualizados a cada iteração.
- **Identificação da estrutura.** O número de regras é determinado pela combinação dos conjuntos fuzzy das variáveis de entrada.
- **Identificação dos parâmetros.** Neste modelo de *SNF* só apresentam aprendizado nos parâmetros do conseqüente de cada regra, baseado na medida do erro normalizado.

6.4. Controle por Aprendizado Neuro-Fuzzy

6.4.1. Cálculo do valor de U_{IJ}

O valor da variável adimensional U_{IJ} é obtido como o resultado da avaliação do *SNF* para cada combinação de *mínimo* e *gama*. Assim, para um carregamento com valor mínimo igual a min_i , e gama igual a $gama_j$, tem-se U_{IJ} como saída do *SNF*.

O carregamento gerado pelo sistema servo-hidráulico está na faixa de $[-100,100]$ kN. Assim, os valores da variável de entrada, *mínimo* e *gama* esta na faixa de $[-100,100]$ e $[-200,200]$ respectivamente. Os valores de gama são positivos quando o sistema esta indo de um vale para um pico e negativo quando vai de um pico para um vale. Estas variáveis de entradas do *SNF* (*mínimo* e *gama*) são normalizadas na faixa de $[-1,1]$ utilizando a Equação 6.4.

$$x_n = \frac{2.(x - \min)}{Max - \min} - 1 \quad (6.4)$$

Onde, x_n é o valor normalizado da variável x , \min e Max são os valores de mínimo e Máximo da variável x .

A normalização das variáveis de entrada, *mínimo* e *gama* e feito substituindo na Equação 6.4 e obtêm-se.

$$min_n = \frac{min}{100} \quad (6.5)$$

$$gama_n = \frac{gama}{200} \quad (6.6)$$

Depois da normalização das variáveis de entrada na camada de entrada, vide a Fig. 6.5, na camada fuzzy calcula-se o grau de pertinência com que as entradas satisfazem aos conjuntos fuzzy associado a cada entrada.

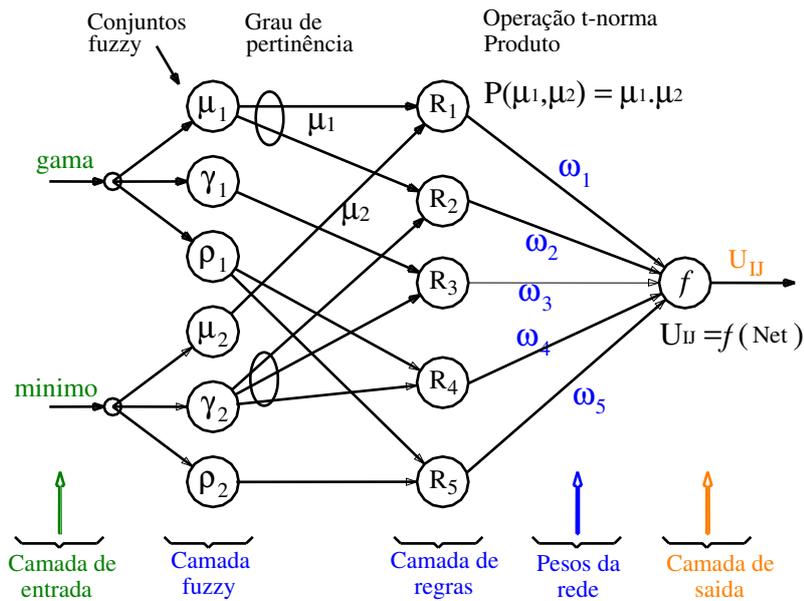


Figura 6.5. Cálculo de U_{II} e descrição das camadas do SNF.

Na camada de regras, calcula-se o nível de disparo correspondente a cada uma das regras, executando a operação *t-norm* (produto).

$$P_k(\mu_i, \mu_j) = \mu_i \cdot \mu_j \quad (6.7)$$

Na camada de saída calcula-se o valor de U_{II} em função do $P_k(\mu_i, \mu_j)$ e ω_k .

$$U_{II} = f(\text{Net})$$

$$\text{Net} = \frac{\sum_{k=1}^M P_k(\mu_i, \mu_j) \cdot \omega_k}{\sum_{k=1}^M P_k(\mu_i, \mu_j)} \quad (6.8)$$

Nesta dissertação, considero-se f como função de ativação linear pela facilidade na implementação experimental, mas pode-se aplicar outras funções do tipo sigmoidais (logsig ou tansig).

$$U_{II} = \frac{\sum_{k=1}^M P_k(\mu_i, \mu_j) \cdot \omega_k}{\sum_{k=1}^M P_k(\mu_i, \mu_j)} \quad (6.9)$$

Onde, $P_k(\mu_i, \mu_j)$ é o resultado da operação *t-norm* na camada de regras e ω_k é o peso de conexão da regra k e o neurônio de saída.

As equações apresentadas acima determinam o valor de U_{II} para qualquer tipo de carregamento. Uma vez calculado o valor de U_{II} , os pontos de reversão da servo-válvula são calculados, pela seguinte equação:

$$Ponto_Reversão = \begin{cases} \min + U_{II} \cdot gama & (na\ subida) \\ (\min + gama) - U_{II} \cdot gama & (na\ descida) \end{cases} \quad (6.10)$$

6.4.2. Lei de aprendizado do SNF

O aprendizado do SNF é feito pelas atualizações dos pesos ω_k em um instante seguinte com o valor atual. Todos os valores de ω_k são inicializados com 0,5 e depois atualizados em função do erro normalizado, a taxa de aprendizado e o nível de disparo de cada um das regras. O erro normalizado é o erro obtido entre o pico (ou vale) desejado x_d e o pico (ou vale) atingido x , dividido pela *gama* obtida entre o pico (ou vale) desejado e o valor de pico (ou vale) atingido na reversão anterior x' , como foi definido na Equação 5.26.

De maneira semelhante ao controle por aprendizado acelerado, se x e x_d forem picos, x' terá sido um vale. Assim, no caso que $x < x_d$ e o *erro* > 0 , tem-se *undershoot*. E no caso que *erro* < 0 , tem-se *overshoot*. Entretanto, se x e x_d forem vales, x' terá sido um pico. Assim, no caso que $x > x_d$, e *erro* > 0 , tem-se *undershoot*. E no caso que *erro* < 0 , tem-se *overshoot*.

Conclui-se assim que erros positivos estão associados ao *undershoot*, e negativos ao *overshoot*, tanto na subida (vale – pico) como na descida (pico-vale).

Pelo geral, o valor de *erro* esta na faixa de [-1,1] e o algoritmo de atualização dos pesos ω_k do *SNF* é dado pela seguinte lei de aprendizado.

$$\omega_k(t+1) = \omega_k(t) + \Delta\omega_k(t) \quad (6.11)$$

$$\Delta\omega_k(t) = \eta \cdot \text{erro} \cdot \frac{P_k(\mu_i, \mu_j)}{\max[P_k(\mu_i, \mu_j)]} \quad (6.12)$$

Onde, $\omega_k(t)$ peso de conexão correspondente á regra k , η é a taxa de aprendizado, erro é o erro normalizado e $P_k(\mu_i, \mu_j)$ é o nível de disparo correspondente á regra k .

6.4.3. Algoritmo de controle por aprendizado Neuro-Fuzzy

O fluxograma do algoritmo de controle por aprendizado Neuro-Fuzzy apresenta os mesmos passos seguidos pelo fluxograma do controle por aprendizado acelerado, com a diferença que o valor de U_{II} é determinado como o valor de saída da avaliação do *SNF*, e a lei de aprendizado atualiza os pesos ω_k do *SNF*, a qual incluem uma taxa de aprendizado η e um termo de momentum Ω que permite acelerar o processo de aprendizado.

Na Figura 6.6 é apresentado o fluxograma do algoritmo de controle por aprendizado Neuro-Fuzzy.

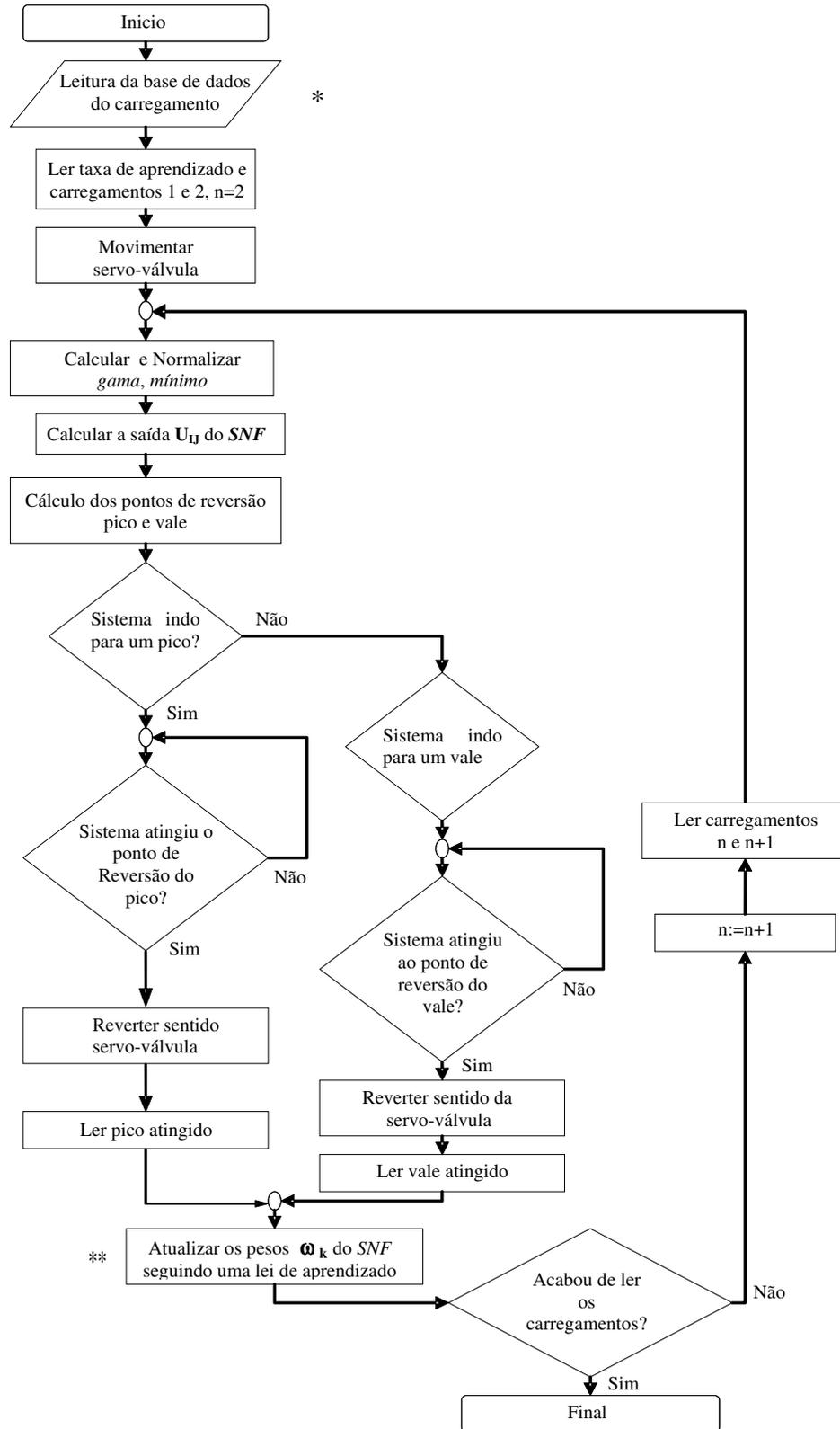


Figura 6.6. Algoritmo de controle por aprendizado Neuro-Fuzzy.

* Os carregamentos são picos e vales requeridos para o ensaio de fadiga

** Para atualizar os pesos ω_k é calculado o erro do pico e/ou vale e usada uma lei de aprendizado.

6.4.4. Resultado das simulações do controle por aprendizado *SNF*

As simulações do controle por aprendizado Neuro-Fuzzy proposto foram feitos no software de Matlab©. Para as simulações escolheram-se carregamentos de amplitude constantes e variáveis, obtendo melhores resultados que os obtidos pelas leis de aprendizado, vide Figuras 6.7 e 6.8.

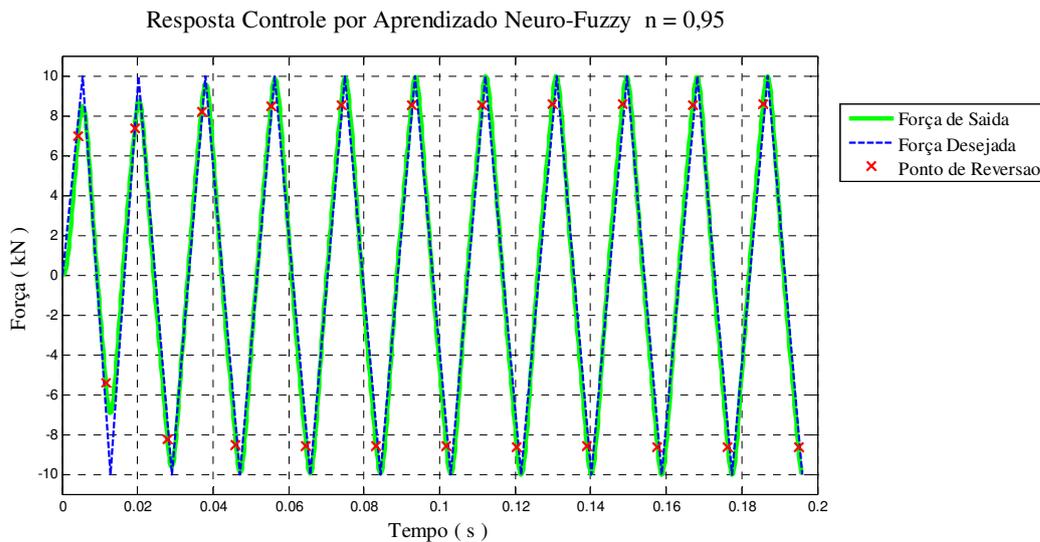


Figura 6.7. Resposta do controle por aprendizado Neuro-Fuzzy para um carregamento de amplitude constante de ± 10 kN .

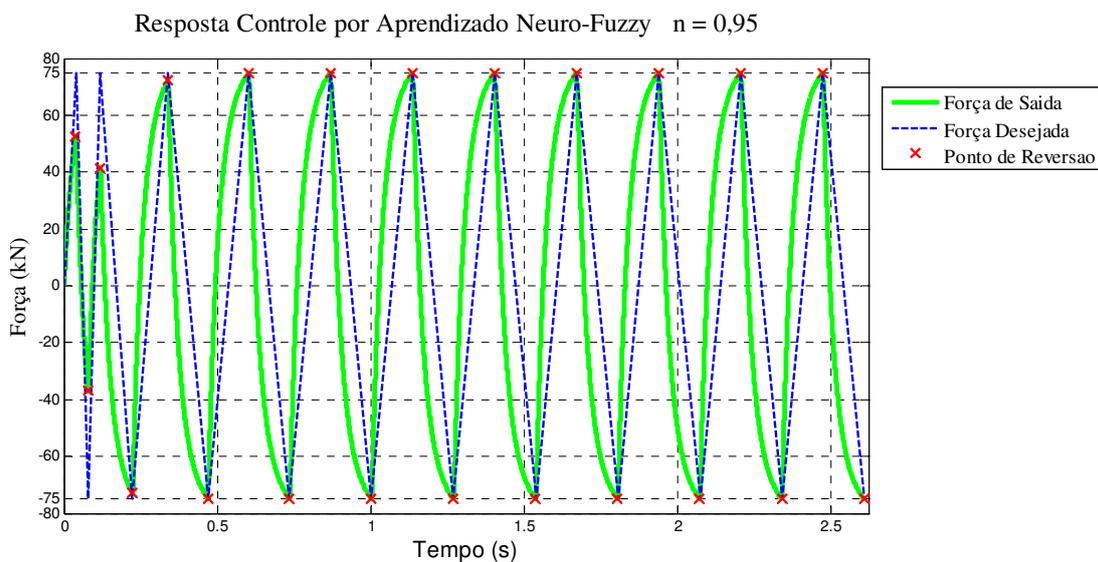


Figura 6.8. Resposta do controle por aprendizado Neuro-Fuzzy para um carregamento de amplitude constante de ± 75 kN.

Nas Figuras 6.7 e 6.8, mostram-se as simulações para carregamentos de

amplitude constante de ± 10 kN e ± 75 kN, respectivamente, com uma taxa de aprendizado $\eta = 0.95$. Os pontos de reversão de pico/vale são modificados à medida que são apresentados novos carregamentos, até convergir em um valor ótimo de reversão. Assim, no futuro para carregamentos da mesma amplitude, o controlador pode responder satisfatoriamente sem a necessidade de reaprendizado.

Na Figura 6.9 é apresentado o desempenho do controle por aprendizado Neuro-Fuzzy para carregamentos de amplitude variável. O algoritmo de aprendizado atualiza os pesos da estrutura Neuro-Fuzzy para cada combinação de *mínimo* e *gama*. Assim, os valores dos pontos de reversão são atualizados após cada ciclo até alcançar pontos de reversão ótimos, como mostrado na figura.

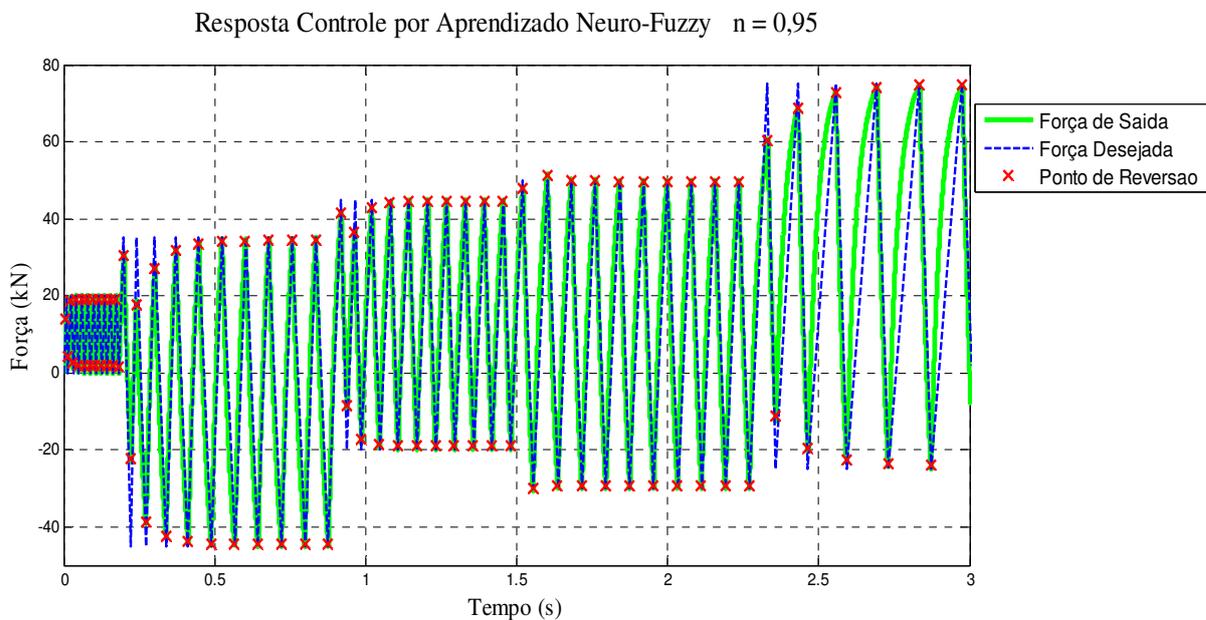


Figura 6.9. Resposta do controle por aprendizado Neuro-Fuzzy para diferentes amplitudes de carregamento.

Toda a informação obtida ao longo da operação do controlador é armazenada nos pesos da estrutura do sistema Neuro-Fuzzy. Portanto, no futuro o sistema não tem a necessidade de reaprendizado para carregamentos que já tiverem sido apresentados anteriormente.

Na figura 6.7-6.8 e 6.9 mostra-se que a força de saída não ultrapassa à força desejada, isto devido á que nos ensaios de fadiga o overshoot (ultrapassa a força

desejada) é indesejado. Entretanto, o undershoot (não ultrapassa a força desejada) é tolerável.

A Figura 6.10 apresenta a resposta do controle por aprendizado Neuro-Fuzzy para um carregamento que já tinham sido apresentados 5 vezes anteriormente. Pode-se notar que o sistema apresenta um ótimo desempenho desde o primeiro ciclo.

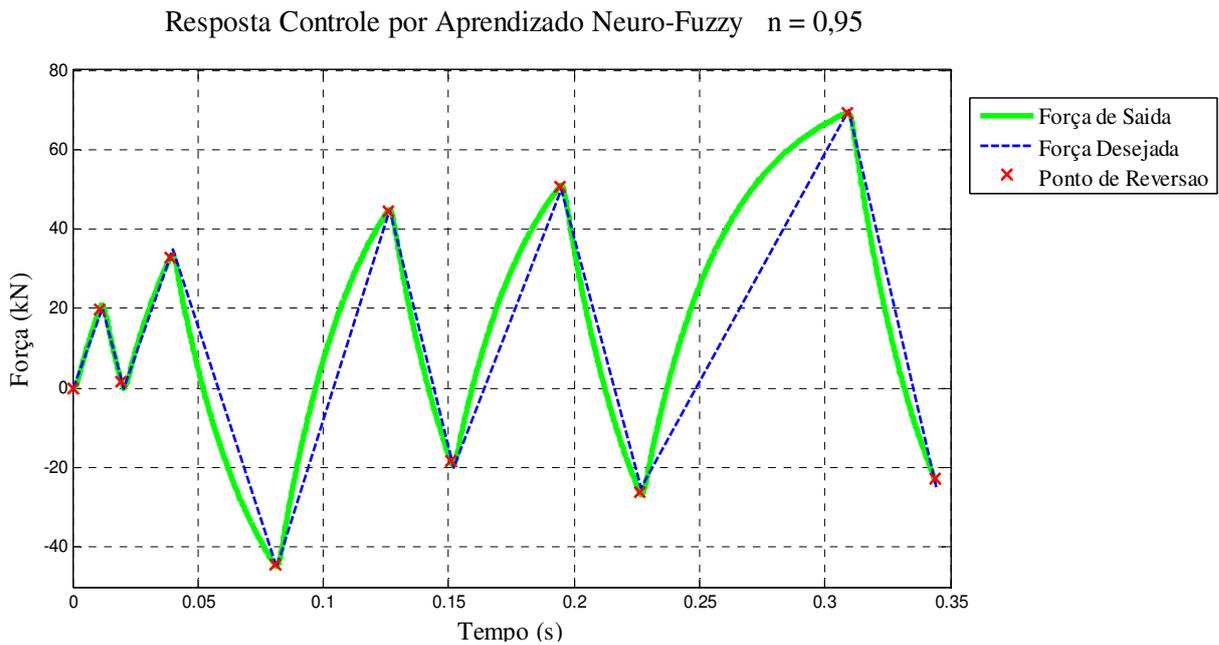


Figura 6.10. Resposta do controle por aprendizado Neuro-Fuzzy para carregamentos já apresentados anteriormente.

Na Figura 6.11, as curvas representam a convergência do erro normalizado ao longo do processo de aprendizagem. A linha “vermelha” representa a convergência do erro para o controle por aprendizado de Alva, enquanto a linha “azul” representa a convergência do aprendizado acelerado, e a linha “verde” do aprendizado Neuro-Fuzzy. Pode-se notar que o aprendizado Neuro-Fuzzy tem rápida convergência, com uma maior velocidade de aprendizado em relação aos outros dois modelos de controle.

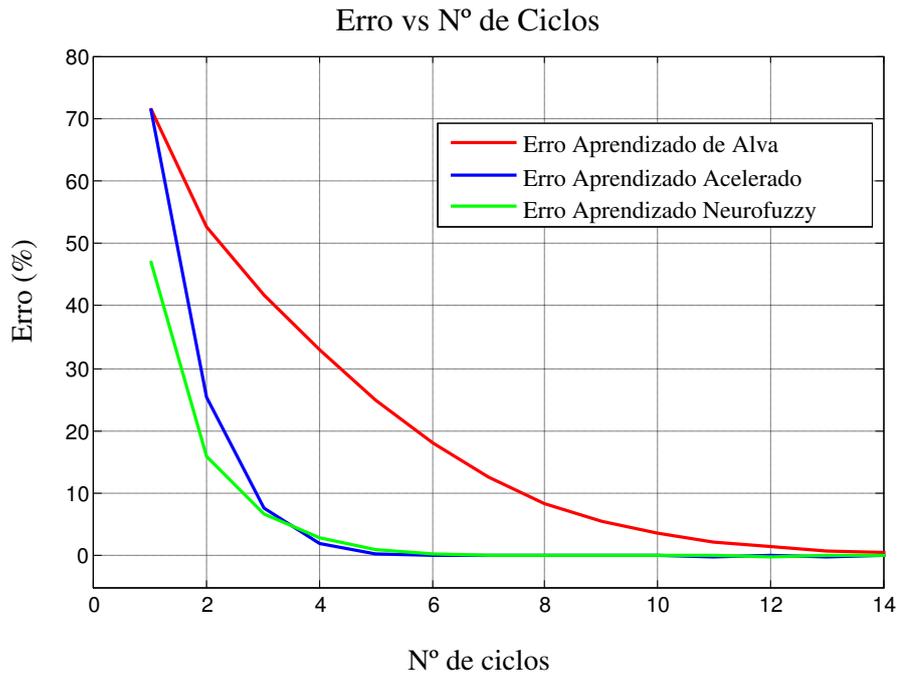


Figura 6.11. Desempenho dos Modelos de controle por aprendizado para carregamento constante ± 20 kN.

Todas as simulações apresentadas nas figuras anteriores foram executadas para uma taxa de aprendizado $\eta = 0.95$. Este parâmetro influi no processo de aprendizagem, fazendo com que o aprendizado seja mais rápido ou lento. Na figura 6.12 é apresentada a convergência do erro ao atingir picos ou vales para os três modelos de controle por aprendizado para $\eta = 0.1$.

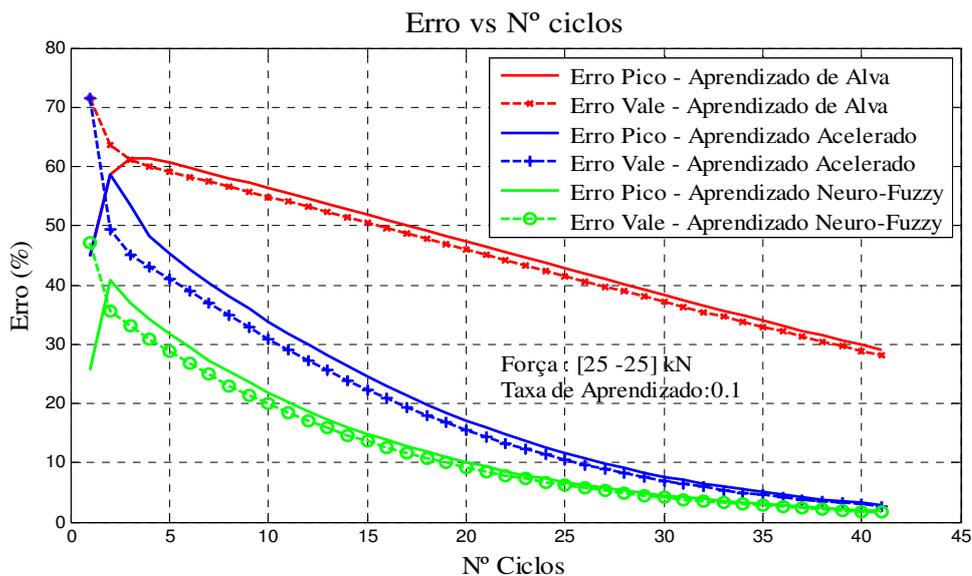


Figura 6.12. Desempenho do controle por aprendizado para $\eta = 0.1$ e carregamento de amplitude constante de ± 25 kN.

Pode-se observar que o erro do aprendizado Neuro-Fuzzy (linha verde) tem uma convergência mais rápida em relação aos outros tipos de controle. No entanto, o valor $\eta = 0.1$ faz com que ela tenha um aprendizado lento, e convergindo após 80 ciclos.

À medida que a taxa de aprendizado aumenta, a velocidade de convergência do erro aumenta, logo a velocidade de aprendizagem do controle é maior.

Na Figura 6.13 é apresentada a convergência do erro pico/vale dos três modelos de controle por aprendizado, para uma taxa de aprendizado $\eta = 0.95$. Pode-se observar que o erro do controle por aprendizado Neuro-Fuzzy (linha verde) converge para zero após cerca de 8 ciclos. O controle por aprendizado acelerado converge em 10 ciclos e o controle por aprendizado de Alva após 30 ciclos.

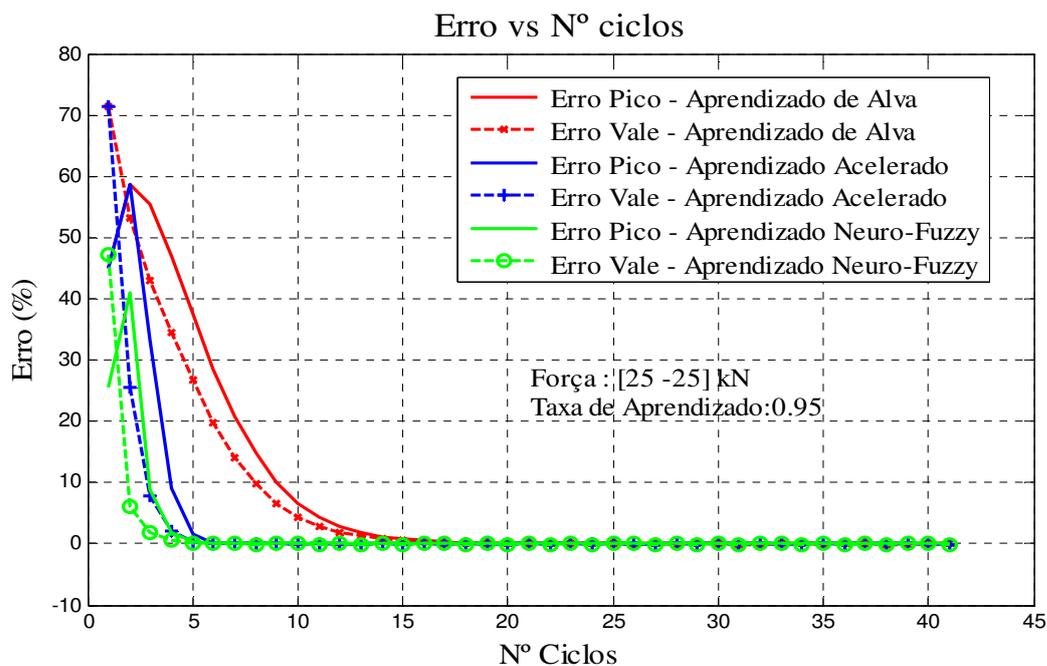


Figura 6.13. Desempenho do controle por aprendizado para $\eta = 0.95$ e carregamento de amplitude constante de ± 25 kN.

O aumento da taxa de aprendizado permite que a velocidade de aprendizagem seja maior. No entanto, uma taxa de aprendizagem muito alta apresenta oscilações no controle e o problema de *overshoot*, que é prejudicial em ensaios de fadiga por introduzir efeitos de sobrecarga.

A Figura 6.14 apresenta o comportamento do erro ao longo do processo de aprendizagem para $\eta = 1.5$. O controle por aprendizado Neuro-Fuzzy (linha verde) apresenta uma rápida convergência, mas apresenta o problema de oscilações, indesejado neste tipo de aplicação, além de um número de ciclos para convergência é maior que no caso $\eta = 0.95$.

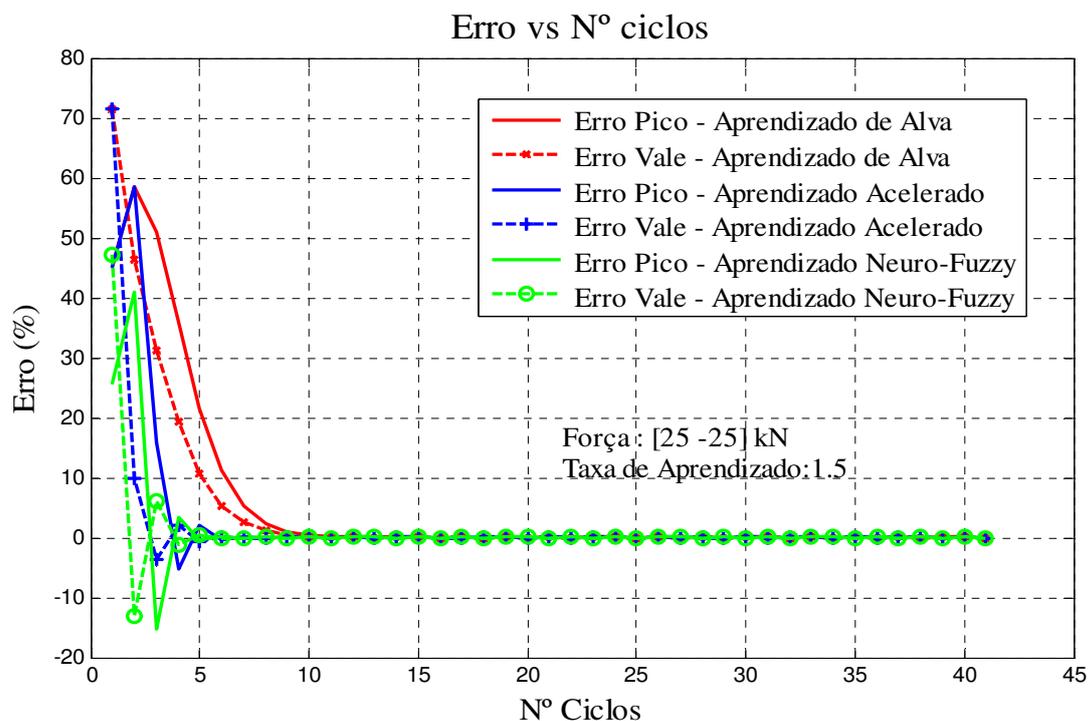


Figura 6.14. Desempenho do controle por aprendizado para $\eta = 1.5$ e carregamento constante de ± 25 kN.

A figura 6.15 apresenta o número de ciclos até que o controle consiga atingir um erro menor que um valor admissível, neste caso definido em 2% (98% de exatidão), em função da taxa de aprendizado, para um carregamento de ± 25 kN. O controle por aprendizado Neuro-Fuzzy (linha verde) tem uma aprendizagem muito mais rápida em relação ao controle por aprendizado desenvolvido por Alva. No entanto, tem um desempenho muito semelhante ao do aprendizado acelerado. Também se mostra que para taxas de aprendizado na faixa de $[0.1, 1]$ o número de ciclos até a convergência diminui à medida que aumenta a taxa de aprendizado. E para valores $\eta > 1$, o número de ciclos para convergência aumenta com a taxa de aprendizado.

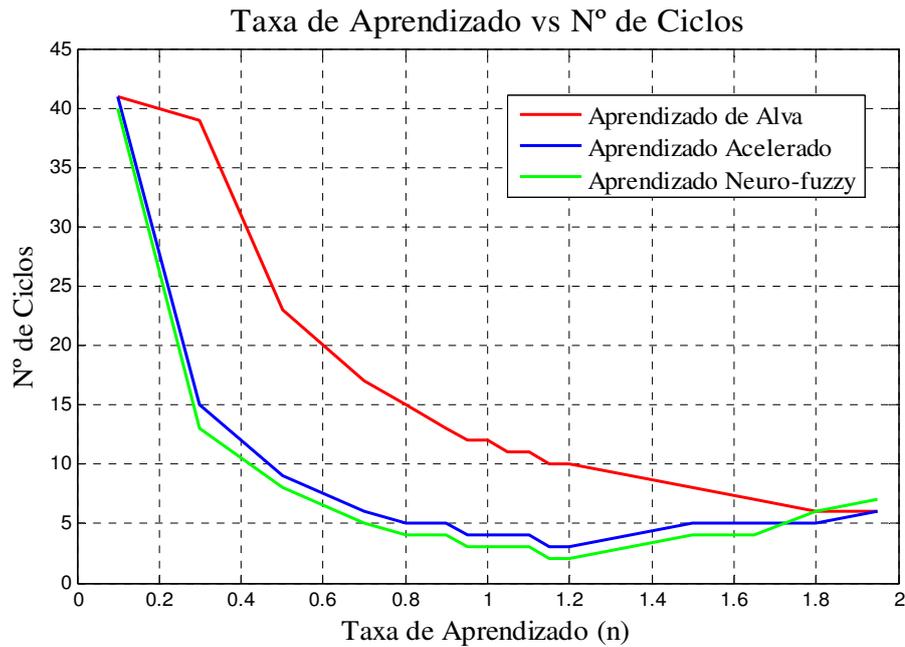


Figura 6.15. Número de ciclos de convergência em função da taxa de aprendizado, carregamento de ± 25 kN.

A Figura 6.16 apresenta o desempenho do erro em cada pico ou vale em função do número de ciclos para um carregamento de ± 80 kN e uma taxa de aprendizado $\eta = 0.1$. O comportamento é semelhante ao obtido para um carregamento de ± 25 kN.

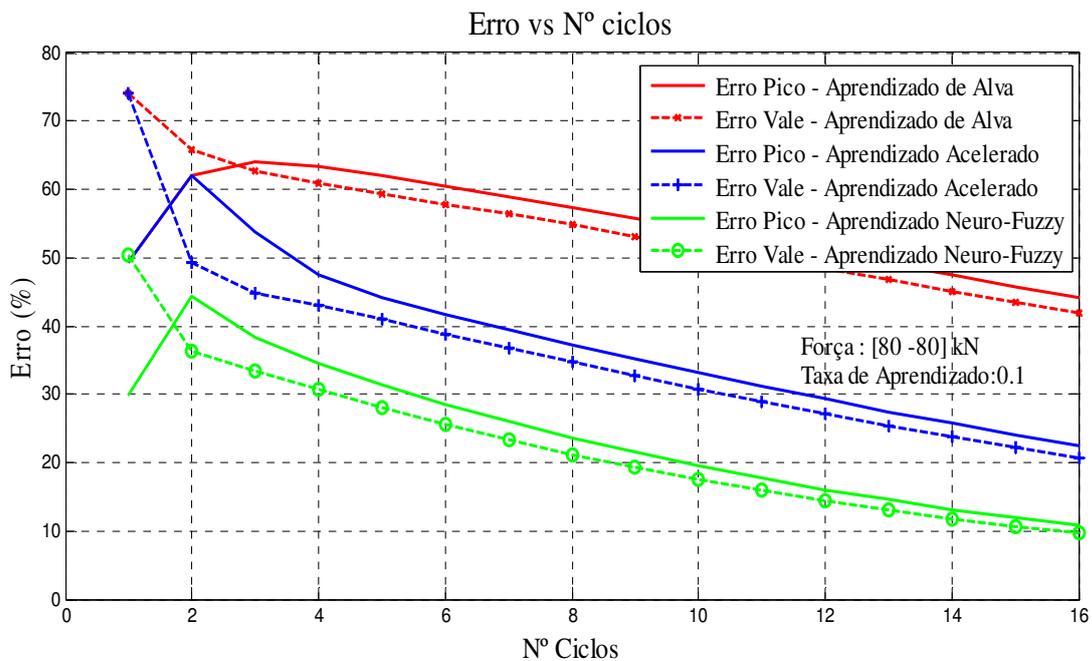


Figura 6.16. Desempenho do controle por aprendizado para $\eta = 0.1$ e carregamento de amplitude constante de ± 80 kN.

Na Figura 6.17 é apresentado o desempenho do erro para os mesmos níveis de carregamento e com uma taxa de aprendizado $\eta = 0.85$. Neste caso, para um carregamento maior, e para uma taxa de aprendizado menor em relação ao caso da Figura 6.13, o controle Neuro-Fuzzy tem um melhor desempenho, convergindo após apenas 6 ciclos.

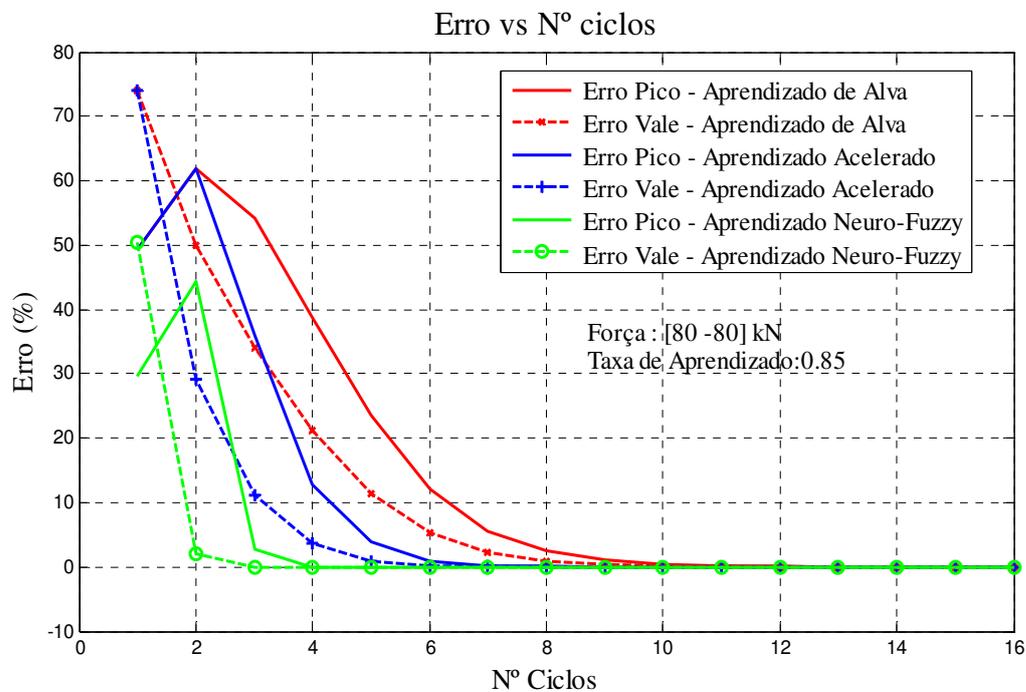


Figura 6.17. Desempenho do controle por aprendizado para $\eta = 0.85$ e carregamento de amplitude constante de ± 80 kN.

A Figura 6.18 mostra que o controle por aprendizado Neuro-Fuzzy apresenta oscilações para uma taxa de aprendizado de $\eta = 0.95$ para a amplitude de ± 80 kN, o que mostra que o desempenho do aprendizado também depende dos níveis de carregamento desejado.

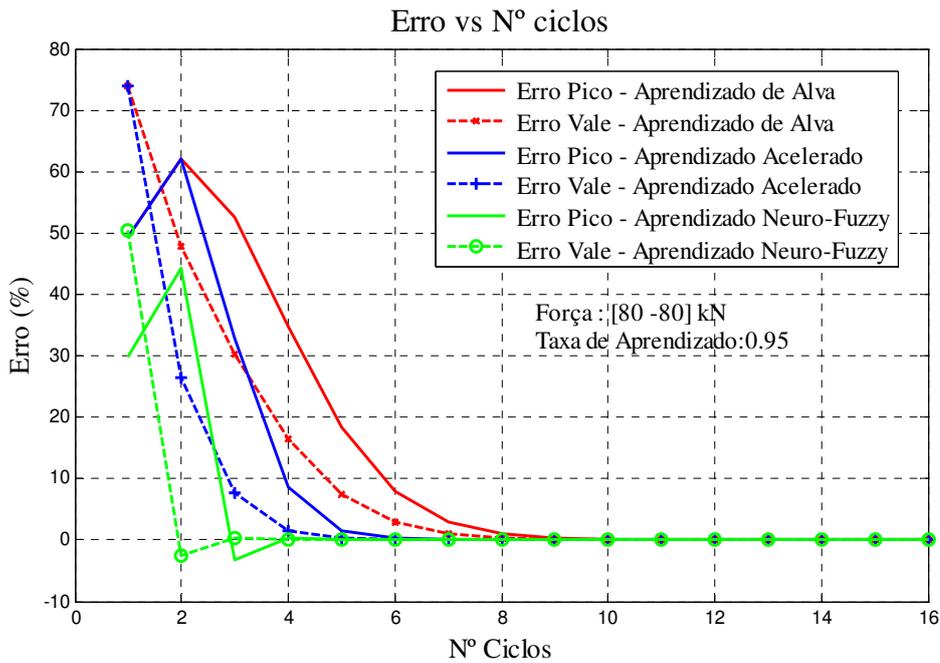


Figura 6.18. Desempenho do controle por aprendizado para $\eta = 0.95$ e carregamento de amplitude constante de ± 80 kN.

A figura 6.19 apresenta o número de ciclos até que o controle consiga atingir o erro de 2% (98% de exatidão), em função da taxa de aprendizado, para um carregamento de ± 80 kN.

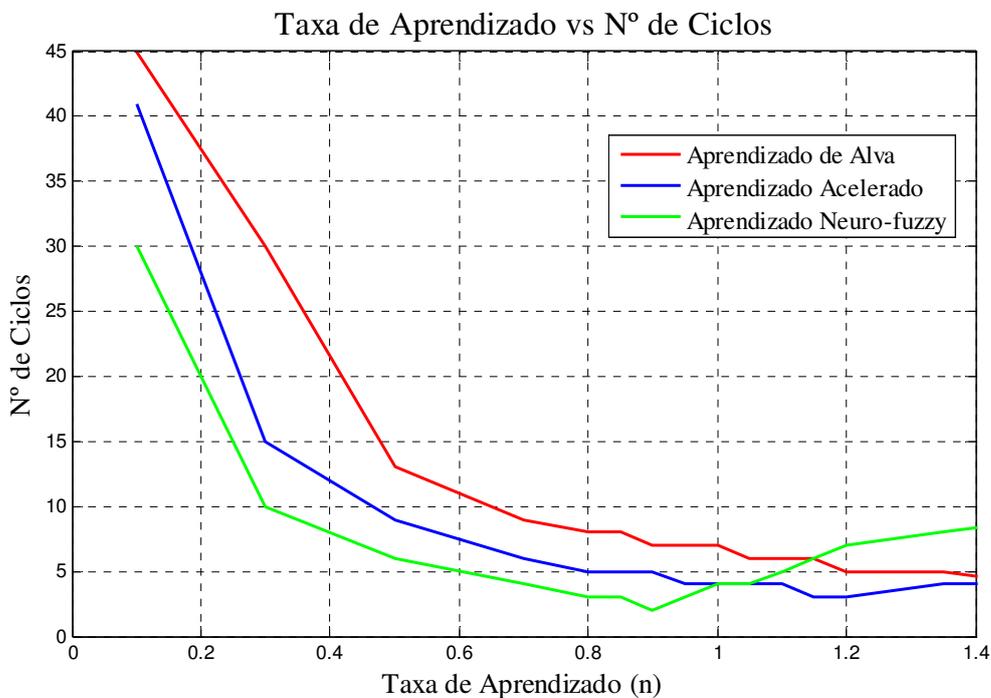


Figura 6.19. Número de ciclos de convergência em função da taxa de aprendizado, carregamento de ± 80 kN.

A tendência do desempenho é similar ao mostrado na Figura 6.15, com a diferença que no controle por aprendizado Neuro-Fuzzy (linha verde), para valores $\eta > 0.9$, o número de ciclos aumenta com a taxa de aprendizado.

Assim, o valor ótimo da taxa de aprendizado com o qual o erro de controle convergiu com o menor número de ciclos depende do nível de carregamento desejado. Isto é apresentado na Figura 6.20, onde se pode observar que o valor ótimo da taxa de aprendizado η para o sistema Neuro-Fuzzy para um carregamento de ± 25 kN (linha vermelha) é perto de 1.2, para o carregamento de ± 55 kN (linha azul) é 1.1, e para o carregamento ± 80 kN (linha verde) é 0.9.

Como a escolha de um valor da taxa de aprendizado maior que o ótimo apresenta oscilações e um maior número de ciclos para convergência, é conveniente escolher o menor η ótimo, que provavelmente estará associado à maior amplitude da história de carregamentos. Isto devido a que as oscilações apresentam casos de overshoot que são indesejados nos ensaios de fadiga.

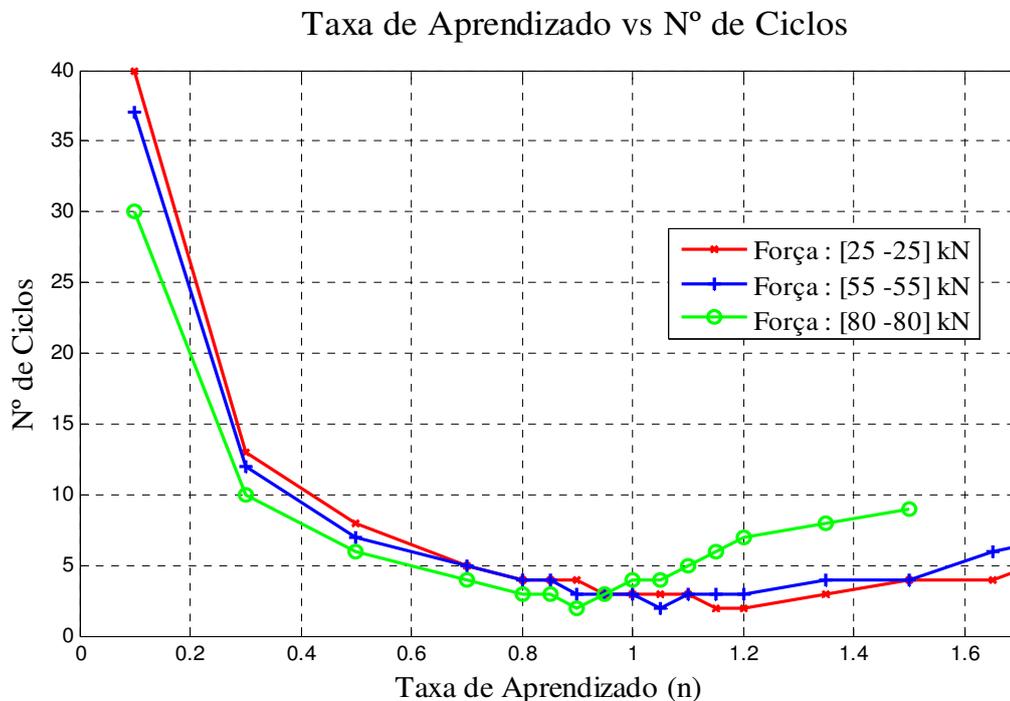


Figura 6.20. Número de ciclos de convergência em função da taxa de aprendizado, para diferentes carregamentos.

Uma sugestão ainda melhor é adotar η variável em uma mesma história, escolhendo seu valor em função da amplitude de cada evento do carregamento.

O controle por aprendizado Neuro-Fuzzy foi testado com uma história padrão de amplitude variável denominada SAE_GKN para uma taxa de aprendizado $\eta = 0.95$. Esta história é composta de 22 ciclos, formando um bloco de carregamento. Os erros para cada ciclo são mostrados na Figura 6.21. Nota-se que os erros se estabilizam a partir do terceiro bloco de carregamento.

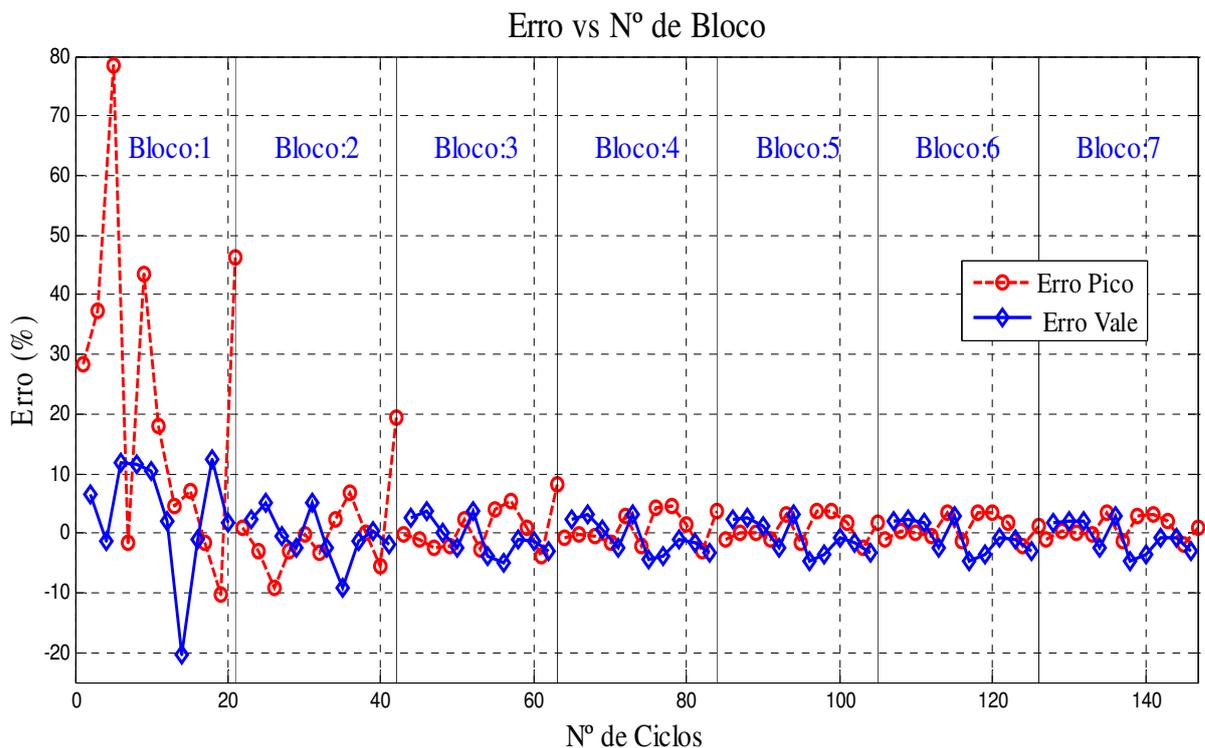


Figura 6.21. Convergência do erro a cada bloco para o carregamento SAE-GKN com picos máximos de 80 kN.

Uma sugestão para melhorar a convergência de um ensaio com carregamento variável é filtrar os carregamentos com amplitudes pequenas devido a que elas podem atrapalhar a convergência no processo de aprendizado.

No primeiro bloco, o erro obtido é relativamente grande, como mostrado na Figura 6.22.

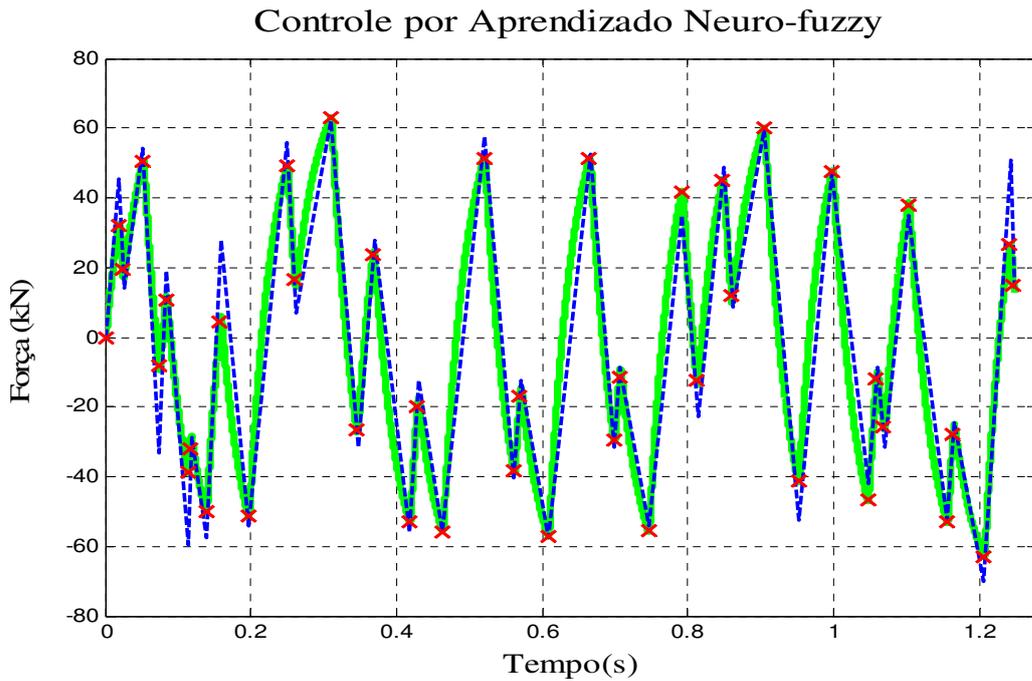


Figura 6.22. Resposta do controle Neuro-Fuzzy no primeiro bloco da história SAE-GKN.

A resposta do controle no quinto bloco é mostrada na Figura 6.23, apresentado erros muito menores que os do primeiro bloco.

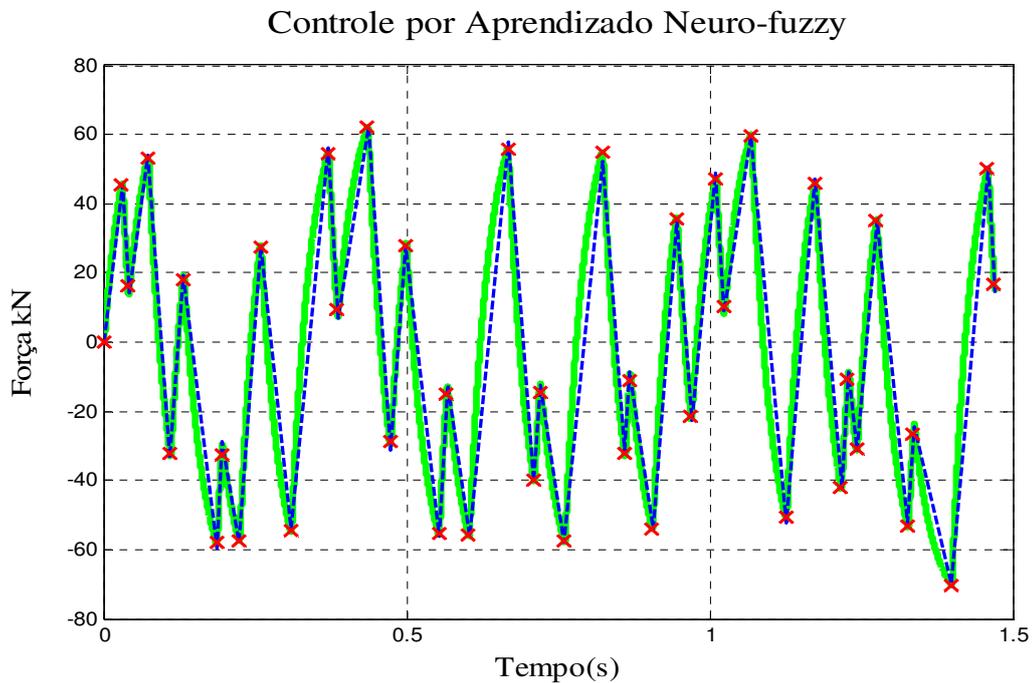


Figura 6.23. Resposta do controle Neuro-Fuzzy depois da convergência, no quinto bloco da história SAE-GKN.

No próximo capítulo, as técnicas de controle apresentadas são aplicadas ao sistema experimental.