

1º Congresso Brasileiro de Redes Neurais

Escola Federal de Engenharia de Itajuba
Itajuba, 24 a 27 de outubro de 1994

CONTROLE AUTOMÁTICO DE SISTEMAS LINEARES EMPREGANDO MODELOS DE REDES NEURONAIS

Roberto Celso Limão de Oliveira

Takashi Yoneyama

Departamento de Engenharia Elétrica
Centro Tecnológico
Universidade Federal do Pará
Caixa Postal 6025, 66.075.900
Belém, PA

Divisão de Engenharia Eletrônica
Instituto Tecnológico de Aeronáutica
CTA - ITA - IEEE
2.225-000
São José dos Campos, SP

RESUMO

Este trabalho descreve a utilização de um modelo de rede neuronal para ser utilizado no controle de um sistema dinâmico. A rede neuronal é treinada como um controlador *feedforward*. Utiliza-se uma técnica simples de aquisição de dados na entrada do modelo neuronal, através de sinais atrasados no tempo. Os pesos representando a intensidade de conexão das sinapses são modificados via algoritmo de aprendizado padrão do tipo *backward error propagation*. Após a fase de treinamento, o controlador neuronal apresenta rápida velocidade de resposta, a qual depende somente do número de níveis de neurônios e do tempo de propagação do sinal através dos mesmos. Os resultados obtidos com o controlador neuronal são comparados com os conseguidos com um controlador PID.

ABSTRACT

This work describes the implementation of a neural network model to be used in the control a dynamical system. The neural network is trained to be used as a feedforward controller. A simple data acquisition technique is applied at the neural network input. A delayers signals bank is used. The weights representing the synaptic connection strenghtes are changed through an backward error propagation learning algorithm. After the training phase, the neural network controller presents a very high response speed with depends only on the number of layers and the time for signal propagation through the neurons. The results obtained with the neural controller are compared with the results of a PID controller.

1. INTRODUÇÃO

Ações de controle são atos naturais aos homens e frequentemente embutido nas máquinas. Existem diferenças significativas entre o controle executado por humanos e o feito por equipamentos. Pessoas fazem o uso de um número grande de informações no planejamento e execução de uma ação de controle quando comparado com controladores industriais. A razão desta diferença não está nas

limitações da disponibilidade de entradas sensoras, mas sim na capacidade dos controladores de processar com sucesso esta quantidade de informações.

Uma outra distinção entre o homem e a máquina é a maneira coletiva de processar informações que o cérebro apresenta, dando ao mesmo a habilidade de responder rapidamente a complexas informações sensoras, sendo que os mais sofisticados algoritmos de controle tem serias limitações de tempo quando

aplicados em tempo real. Uma terceira distinção e a mais importante, é que a ação humana baseia-se no aprendizado, enquanto a operação de controladores artificiais requerem um algoritmo escrito a priori. Devido a estes motivos e com os novos conhecimentos da neurobiologia sobre o funcionamento do cérebro é que surge a idéia de utilizar-se modelos de redes neuronais em sistemas de controle. O intrínseco paralelismo no processamento das informações apresentado por estes modelos, a rápida velocidade de resposta com o qual este processamento é executado e a característica de aprendizado tem despertado o interesse da engenharia de controle.

Um controlador que usa uma arquitetura de rede neuronal exibe três características importantes: utilização de várias informações sensoras, capacidade de processamento coletivo e adaptação [5]. Neste artigo, utilizaremos um modelo de rede neuronal do tipo *perceptron* multicamadas como um controlador *feedforward*. É apresentado um exemplo do seu uso para o controle de um sistema dinâmico linear e os resultados também são comparados com o uso do controlador PID.

2. MODELO NEURONAL TIPO PERCEPTRON

Modelos de redes neuronais tem sido estudados na expectativa de se conseguir a mesma performance que os humanos em algumas tarefas específicas, tais como : reconhecimento de voz e de imagem, controle da atividade motora, problemas de classificação e decisão [6].

Um modelo de rede neuronal é construído a partir de uma estrutura formada por elementos processadores (neurônios), simples e interconectados. Estes elementos processadores operam em paralelo e estão organizados segundo padrões tirados do atual conhecimento biológico [3].

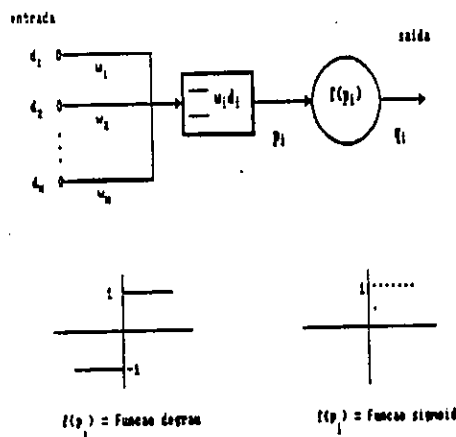


Figura 2.1 - Elemento Processador

Para que o modelo neuronal apresente um bom desempenho, utiliza-se uma densa interconexão dos elementos processadores. Estes são conectados via pesos que são adaptados durante o uso, visando a melhora de um critério de desempenho pré-estabelecido. O modelo neuronal *perceptron* multicamada são redes *feedforward* com um ou mais níveis de elementos processadores entre os elementos processadores de entrada e de saída, chamados de níveis escondidos.

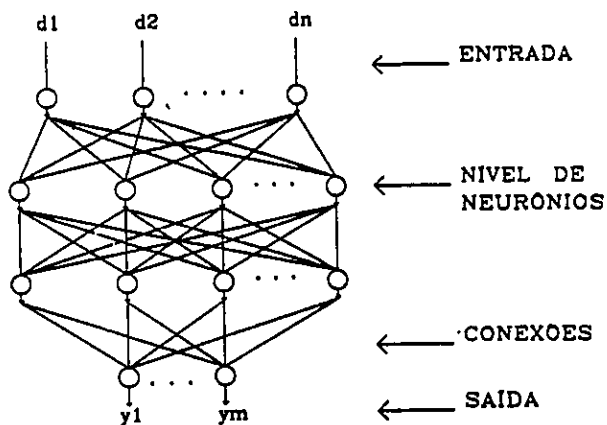


Figura 2.2 - Modelo Neuronal Multicamadas

O procedimento de aprendizado é conseguido por meio do algoritmo *backward error propagation*. Este

iterativo do gradiente que minimize uma função custo quadrática J , que é função do erro entre a saída desejada para o modelo neuronal e saída atual do mesmo [6].

$$J = \frac{1}{2} (u_d - u)^2 \quad (1)$$

3. CONTROLE E APRENDIZADO EMPREGANDO MODELOS DE REDES NEURONAIS

Um sistema onde o modelo neuronal é empregado de modo a funcionar como um controlador *feedforward* é mostrado na figura (3.1), observando-se a saída da planta a ser controlada (y), o padrão de saída desejado (y_d), o laço de realimentação (e), o sinal realimentado (u_f), a saída do modelo neuronal (u_i) e o sinal de controle (u). O sinal realimentado $u_f(t)$ é utilizado como o sinal de erro ($u_d - u$) no algoritmo de aprendizado.

são suficientes para que a rede neuronal encontre, via aprendizado, os valores dos pesos necessários a ação de controle desejada, onde o número de sinais atrasados no tempo é determinado pela ordem da planta. A utilização de um banco de atrasos na entrada do modelo neuronal determina um técnica simples de aquisição de sinais, de modo a fornecer ao controlador neuronal as informações relativas a dinâmica desejada para o processo controlado, facilitando o uso deste tipo de controlador em operações de tempo real.

O controle e o aprendizado são feitos simultaneamente, observando-se que a medida que a rede vai aproximando-se da ação de controle desejada, o sinal realimentado deixa de ter influência no controle ficando apenas o sinal de saída do modelo neuronal a comandar a planta.

4. EXEMPLO APLICATIVO

Utiliza-se como planta a ser

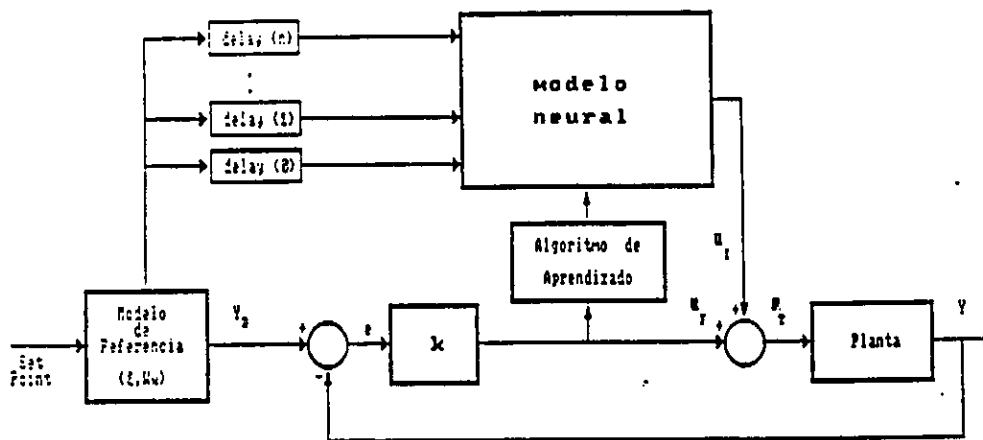


Figura 3.1 - Arquitetura de Controle e Aprendizado

Esta estrutura de controle assume o não conhecimento sobre a dinâmica da planta, exceto pela ordem da mesma. Sendo que na entrada do modelo neuronal temos apenas um conjunto de sinais atrasados no tempo. O uso dos sinais

controlada, um sistema servo-posicionador, sendo que o problema de controle incorpora tanto o caso de regulador como o caso de seguimento de trajetória. Nas simulações é utilizada a

$$G(s) = \frac{600}{s^3 + 50s^2 + 600s} \quad (2)$$

A rede neuronal utilizada é composta de três níveis, sendo que há quatro elementos processadores no nível de entrada, cinco elementos processadores no nível escondido e um elemento processador no nível de saída. O nível de entrada recebe os quatro sinais de entrada $y_d(t)$, $y_d(t-1)$, $y_d(t-2)$ e $y_d(t-3)$ onde observa-se a informação da ordem da planta $n=3$. O último nível tem como saída do único elemento processador o sinal de controle para a planta, sendo que a função ativação deste elemento processador é uma função linear com saturação em valores iguais a ± 10 , indicando limitações físicas no sinal de controle. Nos outros níveis, todas as funções ativações são do tipo sigmoide.

Utiliza-se uma constante de aceleração no algoritmo de aprendizado de 0.025 e pesos iniciais aleatórios com distribuição normal de média zero e variância 0.001. A curva desejada, que segue o modelo de referência de segunda ordem discreto [1], tem como parâmetros o coeficiente de amortecimento igual a 0.8, frequência natural igual a 1.7 rad/seg e período de amostragem de 0.1 segundos, permitindo um tempo de subida igual a 2.4 segundos e o percentual de sobresinal máximo de 0.01%. O controlador de realimentação que tem como entrada $e = y_d - y$ e gera na saída $u_f = k \cdot e$, tem como parâmetro $k=1$.

Na figura (4.1) está representado apenas o laço de realimentação, sem o sinal de saída do modelo neuronal. Aqui, temos a ação de um controlador proporcional de ganho unitário, sendo que esta ação de controle fornecerá o sinal de treinamento para o modelo neuronal.

aprendizado, em encontrar no espaço de pesos a melhor configuração para a tarefa pré-fixada, podendo-se prever uma certa demora na excursão dos pesos por todo o espaço de estados possíveis.

Verifica-se que, com o aumento do número de iterações, onde cada iteração representa a apresentação de toda a curva desejada e o respectivo ajuste de pesos, na fase de treinamento, a influência do sinal de controle realimentado ($u_f(t)$) diminui gradativamente, sendo mais influente o sinal de saída do modelo neuronal. A rede após o treinamento guarda na forma dos pesos as características desejadas para o controlador, sendo que estes pesos estão ajustados para o padrão de saída desejado pré-fixado. Com estes valores de pesos, o modelo neuronal pode adaptar-se a uma nova situação, representando um conhecimento adquirido para o qual não tenha sido explicitamente treinado. Esta característica da rede neuronal pode ser encarada como um processo de interpolação [8].

A partir de uma nova trajetória desejada, verificou-se o comportamento de saída da planta com o modelo neuronal estando com os pesos fixados pelo treinamento inicial e o algoritmo de aprendizado fora de operação. Para este novo padrão desejado tem-se como parâmetros do modelo de referência um coeficiente de amortecimento igual a 0.6 e frequência natural de 1.5 rad/seg, acarretando um tempo de subida no valor de 1.8 seg e um percentual de sobresinal máximo igual a 0.09%.

Como pode ser observado na figura (4.4), o modelo neuronal adapta-se a mudanças no padrão de saída desejado, caracterizando o processo de interpolação para pontos não treinados.

Para verificar o comportamento do modelo neuronal face a presença de ruídos, a partir dos pesos treinados inicialmente (pesos responsáveis pelo comportamento apresentado na figura

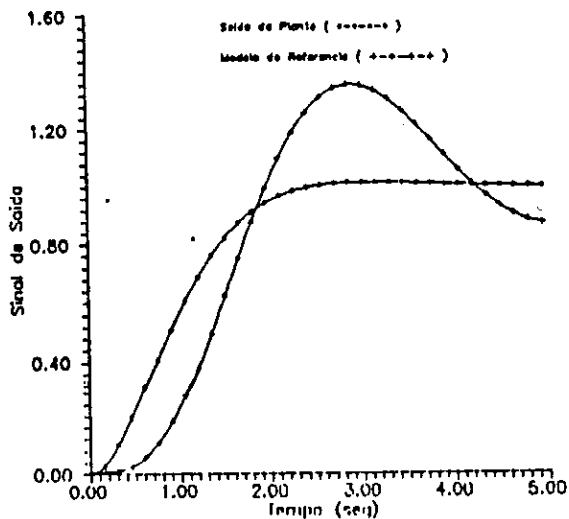
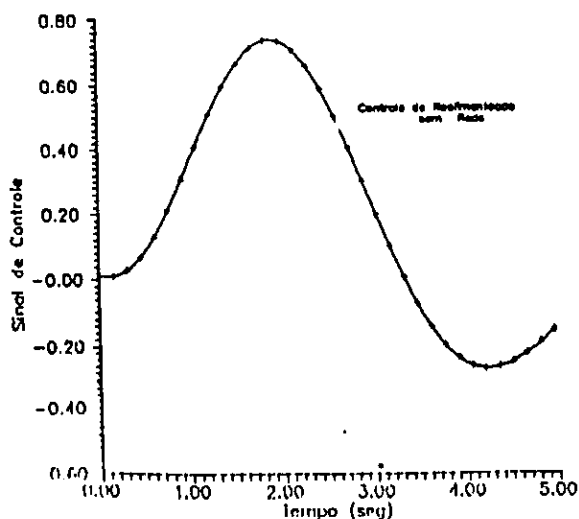


Figura 4.1 - Sinal de Controle e Saída da Planta antes do treinamento

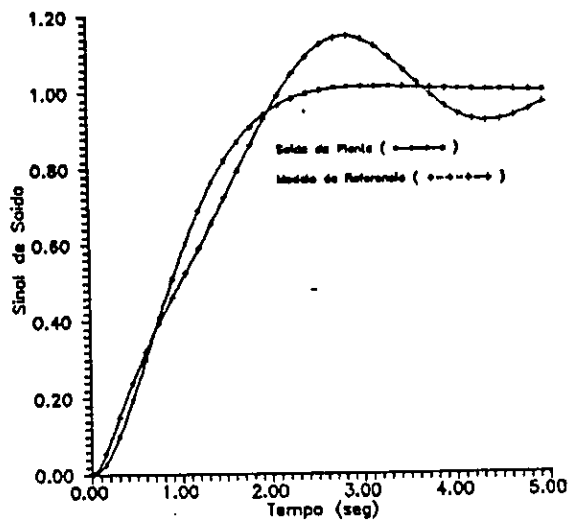
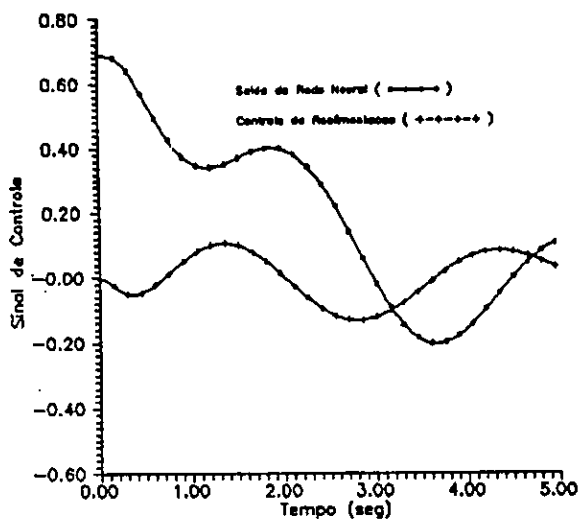


Figura 4.2 - Sinal de Controle e Saída da Planta após 5000 iterações

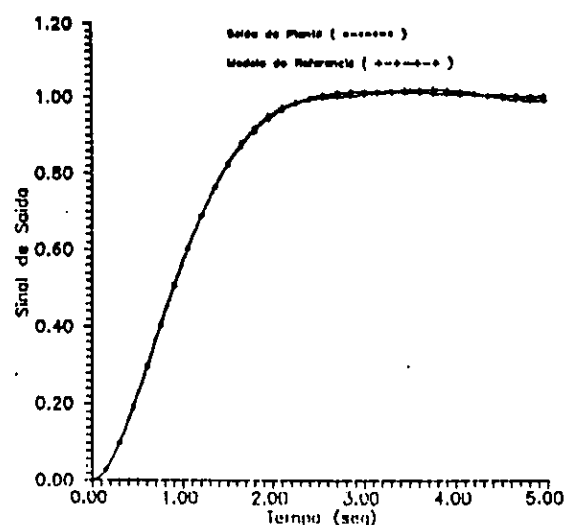
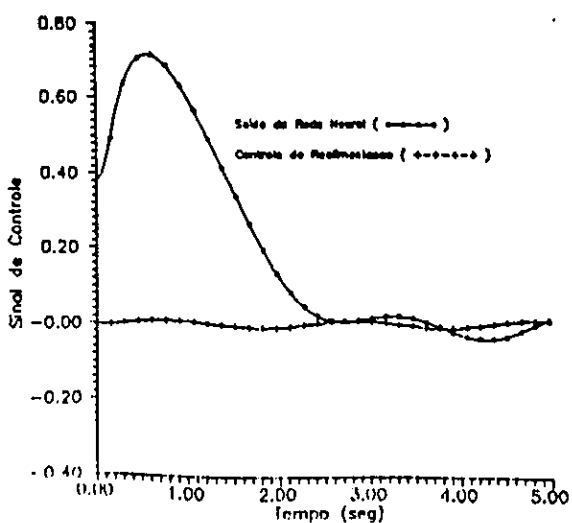


Figura 4.3 - Sinal de Controle e Saída da Planta após treinamento (10000 iterações)

sensor, estando novamente o algoritmo de aprendizado desligado. A performance do

controlador neural é mostrada na figura (4.5), onde observa-se o bom desempenho do mesmo.

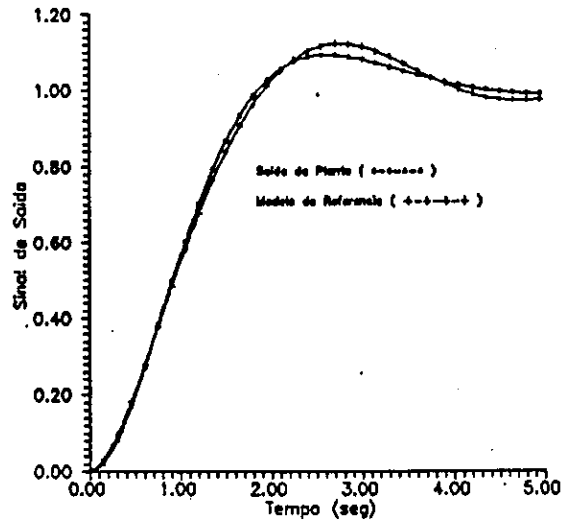
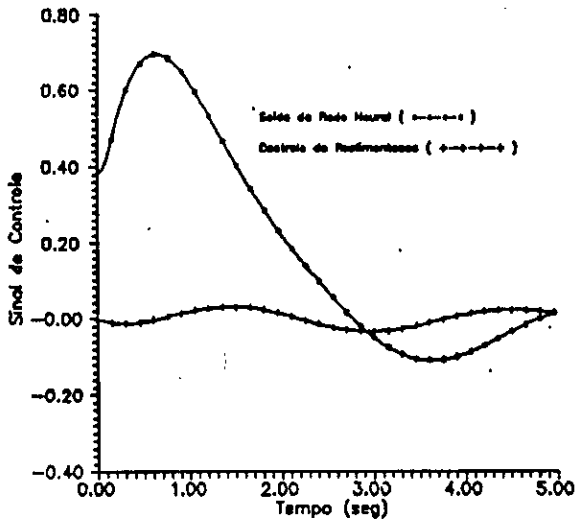


Figura 4.4 - Sinal de Controle e Saída da Planta após mudança no padrão desejado

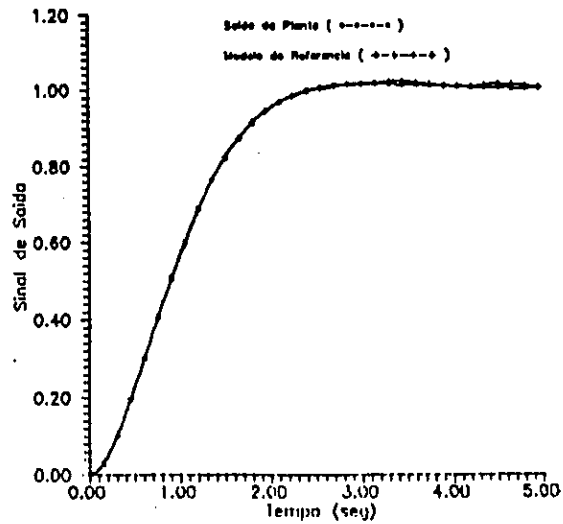
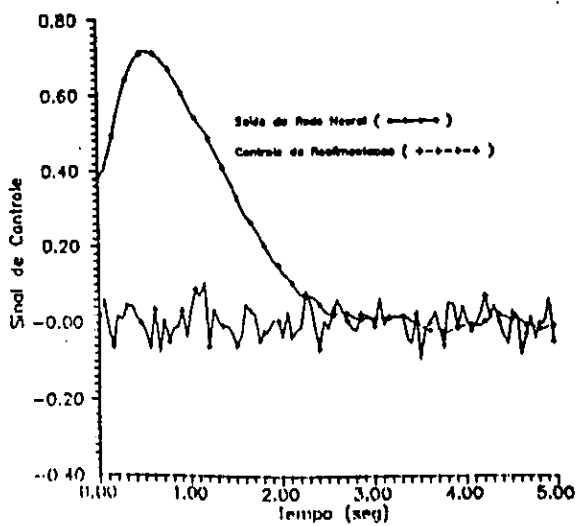


Figura 4.5 - Sinal de Controle e Saída da Planta com ruído no sensor

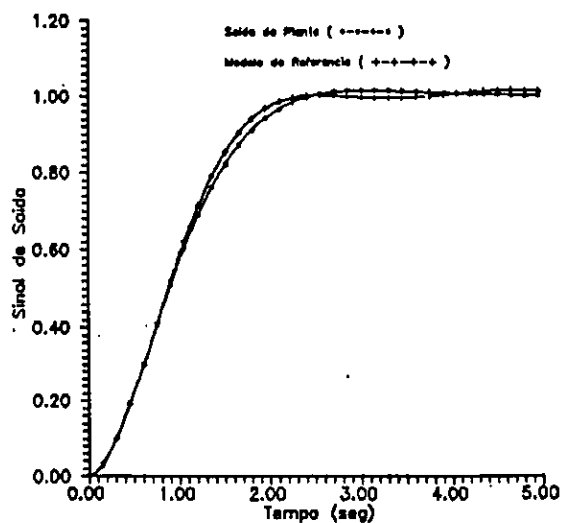
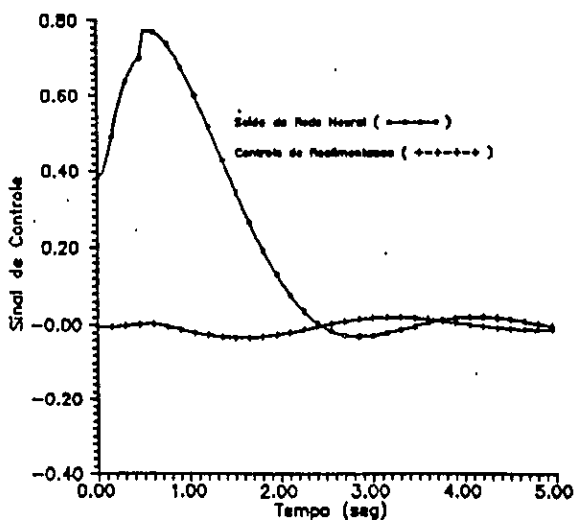


Figura 4.6 - Sinal de Controle e Saída da Planta com variação na carga de 11%

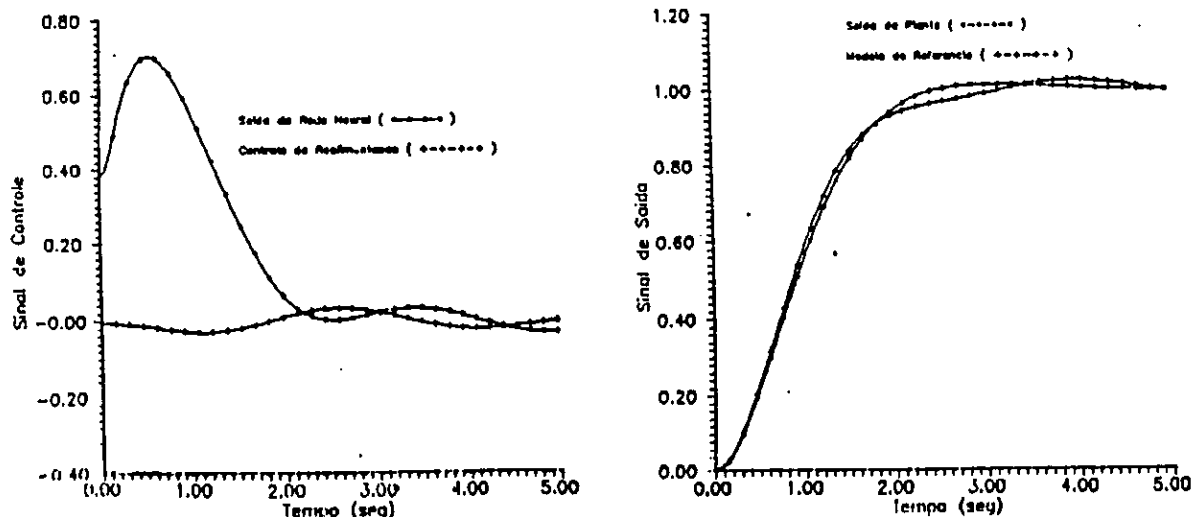


Figura 4.7 - Sinal de Controle e Saída da Planta após modificação no valor do peso

Para avaliar o controlador tendo em vista modificações nos parâmetros da planta, modificou-se a carga no eixo do servo-posicionador em mais 11%, continuando o modelo neuronal a ter os pesos responsáveis pelo controle da figura (4.3) e o aprendizado não ativo. Como pode-se observar na figura (4.6), o controlador neuronal apresenta boa performance, mesmo na presença de mudanças nos parâmetros da planta.

Observando-se agora o comportamento do modelo neuronal quando ocorre alguma falha no mesmo, modificou-se externamente o peso que liga o terceiro neurônio do primeiro nível com o terceiro neurônio do segundo nível, à metade do seu valor original, onde este

degradação de performance, determinada principalmente pela importância do peso no conjunto de pesos disponíveis. Sendo que esta importância diminui à medida que o conjunto de pesos aumenta.

Nos quatro casos de modificações no ambiente para o qual o modelo neuronal foi previamente treinado, podemos considerar como pontos de treinos não distantes do ponto inicial. Para os casos em que as mudanças ocorridas são bem acentuadas, e os pesos inicialmente treinados não são capazes de gerar o apropriado sinal de controle, recomenda-se a utilização de uma realimentação hormonal no controlador neuronal.

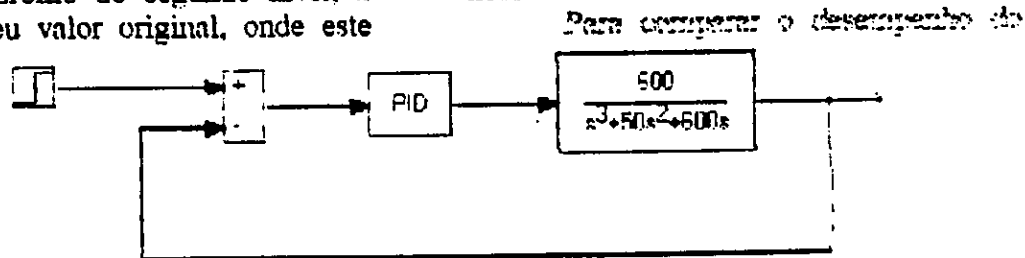


Figura 4.8 - Planta com controlador PID

é dado pelo treinamento da figura (4.3). O resultado desta falha é apresentado na figura (4.7), com o aprendizado inativo, observando-se um desempenho razoável para o controlador neuronal, com alguma

controlador neuronal com um controlador clássico do tipo PID, montou-se o sistema mostrado na figura (4.8), sendo necessário sintonizar os ganho do PID de tal forma que o mesmo tivesse um desempenho que

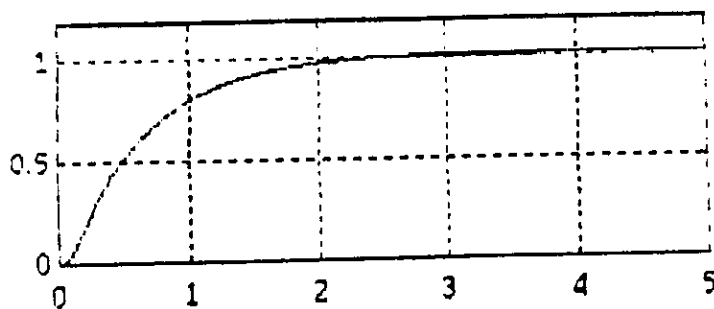


Figura 4.9 - Saída da Planta com o controlador PID

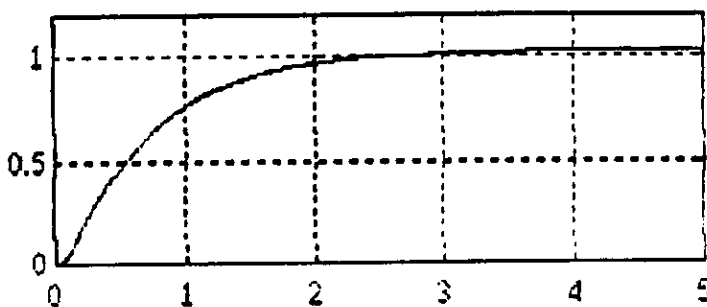


Figura 4.10 - Saída da Planta com variações paramétricas e controlador PID

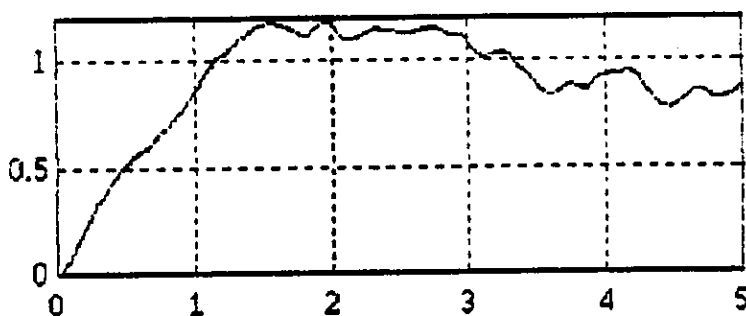


Figura 4.11 - Saída da Planta com ruído no sensor e controlador PID

tornasse a saída da planta próxima ao modelo de referência dado pelo coeficiente de amortecimento igual a 0.8 e frequência natural igual a 1.7 rad/seg. A partir destes ganhos do controlador PID, visto na figura (4.9), verifica-se o comportamento do mesmo frente a problemas de ruído, conforme encontrado na figura (4.5), e a variações de parâmetros da planta, como visto na figura (4.6). Os resultados para o controle através do controlador PID são mostrados nas figuras (4.10) e (4.11), respectivamente para os distúrbios de ruído no sensor e variação paramétrica.

Observando-se estes resultados nota-se o melhor desempenho do controlador neuronal para casos em que

haja presença de ruídos, onde a performance do PID é extremamente degradada

E para a situação de variação paramétrica, o comportamento do controlador neuronal apresenta melhores resultados do que o controlador PID, sendo a diferença de performance menos significativa que o apresentado para o ruído.

Nos casos de mudança no padrão de saída desejado e falha nos ganhos do controlador, não foi testado o PID, porque nestes dois casos o trabalho resume-se apenas em nova sintonia dos ganhos do controlador.

5. CONCLUSÕES

A partir das simulações apresentadas, o modelo neuronal *feedforward*, apresenta um bom desempenho. Necessitando para o seu aprendizado apenas a ordem do sistema a ser controlado, utilizado na forma de $y_d(t)$, $y_d(t-1)$, ..., $y_d(t-n)$, onde n é a ordem do sistema a ser controlado.

Na arquitetura apresentada utiliza-se o sinal de realimentação para resolver o problema da necessidade de se conhecer a priori o sinal de controle da planta sem o qual não consegue-se aplicar o algoritmo de aprendizado.

Nota-se que, apesar do número de iterações necessárias para concretizar-se o treinamento ser grande, com o pequeno número de elementos processadores utilizados no modelo neuronal, o tempo gasto na sessão de treinamento foi de três horas, com o algoritmo sendo executado em uma máquina 386 da HP.

Com base nas figuras (4.4), (4.5) e (4.6), pode-se afirmar que o modelo neuronal utilizado como controlador *feedforward* na arquitetura da figura (3.1), tem bom desempenho quando na presença de ruídos, variações paramétricas e mudanças no padrão desejado. O conhecimento adquirido no treinamento fica armazenado nos valores dos pesos de uma forma distribuída, tornando o controlador menos sensível a falhas, com esta sensibilidade diminuindo a medida que aumenta-se o número de elementos processadores no modelo.

Relacionando o desempenho do controlador neuronal *feedforward* com um controlador PID, verifica-se que o mesmo apresenta um melhor desempenho, sendo que o PID tem sua performance degradada quando ocorre mudanças no ambiente, tipo variações paramétricas e ruídos.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] - Astrom, K. J. & Wittermark, B. "Computer-Controller Systems - Theory and Design", Prentice-Hall Inc., 1990.
- [2] - Hoskins, D. A. & Vagners, J. "A Neural Network Based Explicit Model Reference Adaptive Controller", Proceedings of the 29th Conference on Decision and Control, Honolulu, Hawaii, December, 1990, pp 1725-1729.
- [3] - Lippmann, R. P., "An Introduction to Computing With Neural Nets", IEEE ASSP Magazine, Vol. 3, No 4, April, 1987, pp 4-22.
- [4] - De Oliveira, R. C. L. & Nascimento Jr., C. L. & Yoneyama, T., "A Fault Tolerant Controller Based on Neural Nets", International Conference Control 91, Edinburgh, March, 1991.
- [5] - Psaltis, D. & Sideris, A. & Yamamura, A. A., "A Multi-Layered Neural Network Controller", IEEE Control Systems Magazine, April, 1988, pp 17-21.
- [6] - Rumelhart, D. E. & Hinton, G. E. & Williams, R. J., "Parallel Distributed Processing", MIT Press, Vol. 1 and 2, 1986.
- [7] - Kawato, M. & Uno, Y. & Isobe, M. & Suzuki, R., "Hierarchical Neural Network Model For Voluntary Movement With Application to Robotics", IEEE Control Systems Magazine, April, 1988, pp 8-15.
- [8] - Filho, P. R. B. & De Oliveira, R. P. & Albuquerque, A. B. & De Oliveira, R. C. L. & Garcez, J. N., "Um Estabilizador PID Inteligente para Sistemas de Potência Usando Redes Neurais Artificiais", a ser publicado nos Anais do 10º Congresso Brasileiro de Automática.

